Parallel-in-time Integration of astro- and geo- physical flows; application of Parareal to kinematic dynamos and Rayleigh-Bénard convection

Andrew Thomas Clarke

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds

EPSRC Centre for Doctoral Training in Fluid Dynamics

August 2021

Declaration

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The work presented in Chapter 4 resulted in the following publication (Clarke et al., 2020b)

- **Clarke, A. T.,** Davies, C. J., Ruprecht, D., and Tobias, S. M. 2020b. Parallel-in-time integration of kinematic dynamos. Journal of computational physics: X.

Andrew Clarke incorporated the Parareal algorithm into the Dedalus code, performed the simulations + analysis and wrote the article while Daniel Ruprecht, Chris Davies and Steven Tobias undertook standard advisory roles.

The work presented in Chapter 5 resulted in the following publication (Clarke et al., 2020a)

- **Clarke, A.,** Davies, C., Ruprecht, D., Tobias, S., and Oishi, J. S. 2020a. Performance of parallel-in-time integration for Rayleigh-Bénard convection. Computing and visualization in science.

Andrew Clarke performed the simulations + analysis and wrote the article while Daniel Ruprecht, Chris Davies, Steven Tobias and Jeff Oishi undertook standard advisory roles.

Acknowledgements

Primarily I thank my supervisors Chris Davies, Daniel Ruprecht and Steven Tobias, without whom this work would not have been possible. Daniel's optimism and enthusiasm for Parallel in time methods alongside Chris's enthusiasm for the geodynamo and comprehensive feedback provided ample motivation throughout the project, whilst Steve's vast knowledge and patience helped me to learn more than I imagined possible.

I would like to thank Jeffery Oishi for his hospitality during my research visit to Bates, and for many insights into the Dedalus code base. I would also like to thank Robert Speck for his hospitality during my research trip to Jülich Supercomputing Centre, and for sharing his time during many shared programming sessions. I'd also like to thank Ian Halliday for encouragement and support offered to a recent graduate interested in further study.

I acknowledge the EPSRC Centre for Doctoral Training in Fluid Dynamics (grant number EP/L01615X/1) for the studentship which funded my studies and attendance at a number of very interesting scientific conferences. In particular I would like to thank Peter Jimack and Claire Savy for their support within the CDT. I would like to thank ICIAM, ISC, Jülich Supercomputing centre and Culham Centre for Fusion Energy for further funding towards travel on international conferences.

I would like to thank members of the Astro- geophysical fluid dynamics group and Deep Earth research group for support with my work, and insights into many other areas of research. I would also like to thank my fellow CDT students for their friendship, support, and many interesting discussions, particularly Rob, Tom, Colin, Andy, Eva, and Craig.

Last, and not least, I must thank my family and friends for their constant support. My wife Haley, for supporting me throughout my studies, my children, Thomas and Emily for keeping a smile on my face, David, Andy C, and Andy W et al. for marching on together, and my parents for always believing in me.

Abstract

The precise mechanisms responsible for the natural dynamos in the Earth and Sun are still not fully understood. Numerical simulations of natural dynamos are extremely computationally intensive, and are carried out in parameter regimes many orders of magnitude away from real conditions. Parallelization in space is a common strategy to speed up simulations on high performance computers, but eventually hits a scaling limit. Additional directions of parallelization are desirable to utilise the high number of processor cores now available. Parallel-in-time methods can deliver speed up in addition to that offered by spatial partitioning but have not yet been applied to dynamo simulations. This thesis investigates the feasibility of using Parallel-in-time integration to speed up numerical simulations of dynamos.

We concentrate on applying the non-intrusive Parareal algorithm to two sub-problems of natural dynamos: kinematic dynamos and Rayleigh-Bénard convection (RBC). We perform real-world scaling tests on high performance computing (HPC) facilities using the open source Dedalus spectral solver.

The kinematic dynamo problem prescribes a fluid flow and observes how the magnetic field changes over time. We investigate the time independent Roberts and time dependent Galloway-Proctor 2.5D dynamos over a range of magnetic Reynolds numbers. Speed ups beyond those possible from spatial parallelisation are found in both cases. Results for the Galloway-Proctor flow are promising, with Parareal efficiency found to be close to 0.3, while Roberts flow results are less efficient, with efficiencies < 0.1. Parallel in space and time speed ups of 300 were found for 1600 cores for the Galloway-Proctor flow, with total parallel efficiency of 0.16.

Convective motions are thought to be the source of dynamo action in the Earth and Sun. RBC is the archetypal problem for convection studies, and is also a fundamental problem of fluid dynamics, with many applications to geophysical, astrophysical, and industrial flows. We investigate Parareal for Rayleigh numbers $\mathrm{Ra} = 10^5$, 10^6 and 10^7, finding limited speed up in all cases for up to $\sim$20 processors, whilst performance and convergence of Parareal degrades as Ra increases.

We summarise our results for the kinematic dynamos + RBC, and discuss their relevance and implications on Parallel-in-time simulations for the full dynamo problem.

Contents

List of Figures

List of Tables

Nomenclature

Dimensionless numbers

Nu	Nusselt Number
Pr	Prandtl Number
Ra	Rayleigh Number
Re	Reynolds number
Rm	Magnetic Reynolds Number

Constants and Parameters

N_{BL}	Number of grid points in Boundary Layer
ρ_c	Charge Density
L	Characteristic length scale
U	Characteristic speed
ρ	Density
Ω	Mathematical Domain
μ	Dynamic viscosity
σ	Electrical Conductivity of fluid
g	Gravity
h	Height of convective domain
ν	Kinematic viscosity
c	Speed of Light
η	Magnetic Diffusivity

μ_0 Magnetic Permeability

N_x, N_y, N_z Spatial resolution of x, y, z dimensions

ϵ_0 Permittivity of free space

λ Thermal Conductivity

κ Thermal Diffusivity

α Coefficient of thermal expansion

τ Characteristic time scale

Mathematical Variables

T_n n^{th} Chebyshev polynomial of first kind

t Time

x, y, z Physical Space dimensions

$\boldsymbol{F}$ Body Forces

P Buoyancy Production

q_c Electric Charge

$\boldsymbol{j}$ Current Density

$\boldsymbol{E}$ Electric Field

ε Error/ defect

γ Dynamo Growth Rate

q_T Heat Flux

k_x, k_y, k_z Wave numbers in x, y, z direction

$\boldsymbol{F}_L$ Lorentz Force

L_x, L_y, L_z Length of domain in x, y, z dimensions

$\boldsymbol{B}$ Magnetic Field

p Pressure

T Temperature Field

$\boldsymbol{u}$	Velocity Field
ϵ_U	Viscous Dissipation
u, v, w	Components of velocity in x, y, z direction

Operators

Δ	Difference
FFT (u)	Fourier transform of u
$\widehat{u}$	Fourier transform of u
iFFT $(\widehat{u})$	Inverse Fourier transform of $\widehat{u}$
log	Natural logrithm (base e)
$\langle u \rangle_H$	Horizontal plane average of u
$\overline{u}$	Time average of u
$\langle u \rangle$	Volume average of u

Parareal Notation

$\mathcal{G}$	Parareal Coarse solver
Δt	Coarse time step size
$\mathcal{T}$	Simulation duration
E_P	Parallel Efficiency
$\mathcal{F}$	Parareal Fine solver
δt	Fine time step size
k	Parareal iteration number
$\mathrm{NP}_{\mathrm{Parareal}}$	Parareal time slices/ available processors
$R_{\mathcal{G}}$	Runtime coarse solver
$R_{\mathcal{F}}$	Runtime fine solver
S_P	Parallel Speed up
$\Delta\mathcal{T}$	Length of Parareal time slice
ϕ_n^k	Parareal solution at timeslice n, iteration k

Chapter 1

Introduction

1.1 Overview

Naturally occurring magnetic fields are ubiquitous in our universe. They are present in the Earth, other planets of the Solar System, our Sun, other stars, and in distant galaxies. These magnetic fields are generated by the dynamo effect (e.g. Roberts and Soward, 1992; Weiss, 2002; Rincon, 2019), which changes kinetic energy of a conducting fluid into magnetic energy through complex dynamical motions. Our understanding of the processes governing this effect are incomplete, in large part due to the huge disparities of time and length scales present in these systems which makes mathematical studies very difficult. For example, convection in Earth's core has length scales ranging from $\sim$0.1m to $\sim 7 \times 10^6$m (Matsui et al., 2016), and time scales ranging from a day to millions of years (Biggin et al., 2012; Davies et al., 2015). Despite the difficulty, much effort has been expended to gain insight into these processes, as the applications are of great interest for both practical and intellectual reasons.

Much work on understanding the behaviour of dynamo systems is done using numerical models, see for example the works by Glatzmaier and Roberts (1995b); Gómez et al. (2005); Tobias and Cattaneo (2013); Matsui et al. (2016); Schaeffer et al. (2017). Analytical study is incredibly difficult, due to the nonlinear nature and highly turbulent flows, and experimental studies require large volumes of conducting liquids to be given vast amounts of kinetic energy, a significant practical, engineering, and financial challenge; to date there have been only three successful dynamo experiments (Gailitis et al., 2000; Stieglitz and Müller, 2001; Aumaître et al., 2008). Because of the huge range of time and length scales, the spatial and temporal res-

olutions required to obtain simulations that operate at the physical conditions of geophysical and astrophysical bodies/systems lead to tremendous computing costs, for example a recent study by Schaeffer et al. (2017), used up to 10 million cpu hours per simulation, only ran for a fraction of a magnetic diffusion time, and did not get close to reaching Earth-like conditions. Numerical dynamo simulations are thus run on massive parallel computers (Matsui et al., 2016). Challenges are anticipated for current numerical codes running efficiently on exascale systems due to the requirement for global inter-processor communication in spectral methods (Wicht et al., 2010), and more aggressive parallelization strategies than those in current use are likely to be required (Davies et al., 2011) to make best use of the next generation of high performance computing facilities.

Parallel-in-time (PINT) methods have been studied for the last 50 years (Gander, 2015), and have been shown to increase scalability of numerical simulations beyond saturation of spatial parallelisation (Croce et al., 2014). Interest in the area was boosted by the creation of the Parareal algorithm (Lions et al., 2001). Since its creation, the Parareal algorithm has been shown to speed up numerical simulations in the areas of fluid flow (e.g. Fischer et al., 2005; Croce et al., 2014), plasma physics (Samaddar et al., 2010), financial market derivatives (puts) (Bal and Maday, 2002), simulations of the planetary mantle (Samuel, 2012), robotic manipulation (Agboh et al., 2020) and simulation of an induction machine (Gander et al., 2019), among many others. A number of other Parallel-in-time methods have also become popular in recent years, including PFASST (Minion, 2011), ParaExp (Gander and Güttel, 2013), MGRIT (Friedhoff et al., 2013), and PITA (Cortial and Farhat, 2009). As the method which sparked the recent interest in PINT, and as one of the most simple to implement PINT methods, Parareal is a good choice for initial investigations into new application areas, and its application to dynamo studies will be the focus of this thesis.

The dynamo problem consists of finding or producing a fluid flow that has inductive properties capable of sustaining a growing magnetic field (Charbonneau, 2020). The dynamo problem is very complex, both physically and numerically. Because of this complexity, many studies are carried out which concentrate on just one aspect of the problem, ignoring other physical effects. Two very important examples are how fluid flows affect a magnetic field, and how such fluid flows are generated. The first problem is called the kinematic dynamo problem, and prescribes an arbitrary flow to investigate its effect on an initially infinitesimal seed magnetic field (Rincon, 2019). The solar and planetary dynamos are thought to be convection-driven (Busse, 2002; Jones et al., 2011); convection is thought to be present in the Earth's outer core due to the large Rayleigh number and loss of heat. Therefore,

investigation into the flows responsible for dynamo action should concentrate on convective flows. Rayleigh-Bénard convection is the simplest and best understood form of convective flow, and describes the buoyancy driven flow of a fluid heated from below and cooled from above (Ahlers et al., 2009). It's relative simplicity and high level of analytical and numerical understanding makes Rayleigh-Bénard convection a natural choice for investigation of any novel numerical methods in this application area.

The large range of Parallel-in-time (PINT) methods available, coupled with the immense computational task of completing a realistic dynamo simulation, make the study of PINT methods for the dynamo problem a potentially very large area of research. In order to make the problem more tractable, the research question must be narrowed down somewhat. This thesis investigates the performance of one PINT method - Parareal, when applied to two sub-problems closely related to dynamos - the kinematic dynamo and Rayleigh-Bénard convection.

This thesis is laid out as: in the remainder of this chapter, we will introduce Parallel-in-time methods, the dynamo problem and the specific problems of the kinematic dynamo and Rayleigh-Bénard convection. In Chapter 2 we will survey the recent literature in parallel-in-time methods, and numerical dynamo simulations. Chapter 3 discusses the mathematical background of dynamos, convection, spectral methods, timestepping and Parareal. In Chapter 4 we present results of Parareal applied to the kinematic dynamo problem, Chapter 5 presents results of Parareal for Rayleigh-Bénard convection, and Chapter 6 concludes the thesis with a discussion of the results and possible future work.

1.2 Parallel in time

Moore's law (Moore et al., 1965, 1975) and Dennard scaling (Dennard et al., 1974), are two closely related predictions effectively stating that computing performance would double around every two years, which held remarkably well for decades. Whilst Moore's law, that the number of transistors on a chip would double every 18 months, has held so far, single processor performance via higher clock rates has reached a technical limit due to power dissipation (Pankratius et al., 2010). In order to continue improvements in computing speeds despite the plateau in CPU frequencies, the extra transistors have increasingly been utilised in combining multiple CPU cores on the same processor chip, (see for example Figure 1.1, Esmaeilzadeh et al. (2011); Johnsson and Netzer (2016)). In order to maximise performance on current and future computing platforms, applications must increasingly make use of parallel programming paradigms.

As a result of this, supercomputing (large scale computing with fast interprocessor communication ie. what we have described as HPC in this work) has become so fundamental to scientific understanding that, in many fields, research depends on it (Wilkinson et al., 2021). The subject of HPC is of such great importance now and moving into the future, that the UK Government commissioned the report on this subject by Wilkinson et al. (2021). The dominant method for achieving parallel speed-up in numerical simulations has been to parallelize in space using domain decomposition (Schreiber et al., 2018). Nearly all parallel applications reach a point of diminishing performance with increased processor counts, which can be derived as a consequence of Amdahl's law (Amdahl, 1967) that parallel gains are limited by serial portions of code. Fixed cost serial portions of the code take up an ever growing proportion of code runtime as the parallel portion is split among more and more processors, whilst communication time between processors increases as some function of the number of processors (dependant upon the communication model). Actual and expected saturation in parallel performance of numerical simulations has led to much research on improving parallel performance and efficiency (e.g. Frigo and Johnson, 2005; Matsui et al., 2016; Burns et al., 2020).

When solving time dependent partial differential equations (PDEs), the time direction is usually not parallelised (Gander, 2015). However, when parallelisation in space saturates, the time direction offers itself as a further direction for parallelisation (Gander, 2015). Parallel-in-time integration methods have been studied for at least 50 years, (e.g. Nievergelt, 1964), can increase scalability of computer simulations beyond saturation of spatial parallelisation (Croce et al., 2014),

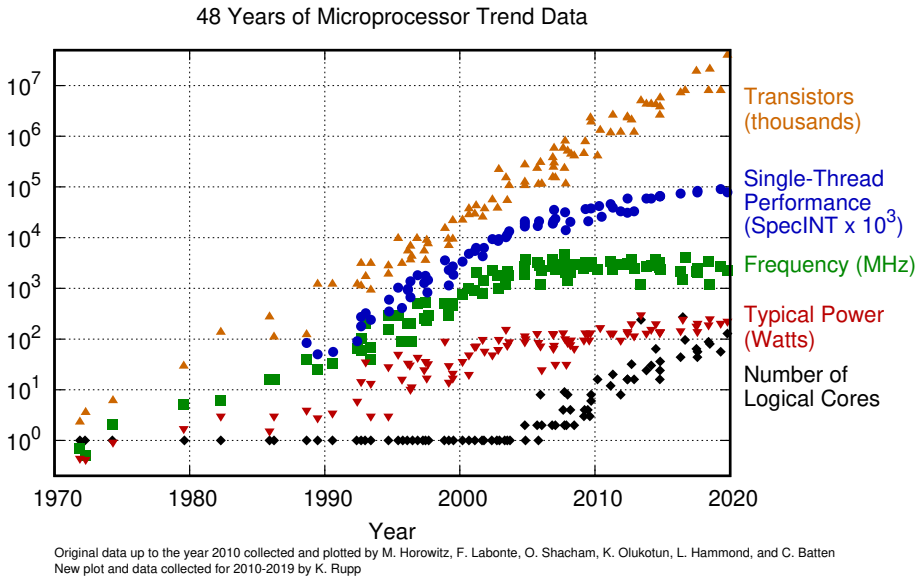


Figure 1.1: Microprocessor Trend Data over 48 years to 2019, by Rupp (2018). Accessible at github.com/karlrupp/microprocessor-trend-data. Original data up to year 2010 by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten.

and there are countless examples of parallel in time methods providing speed ups (e.g. Speck et al., 2012; Schreiber et al., 2018; Samaddar et al., 2017).

In the following section, we describe the method which sparked the recent uptake of interest in parallel in time methods, Parareal. We concentrate on Parareal due to its widespread use in the literature, and because it can be seen as the basis/special case of several other methods. We will give a brief overview of the algorithm in this chapter, whilst further mathematical details can be found in §3.3. A brief overview of some other parallel-in-time methods is shown in table 1.2, with discussion and references in §2.1.2. We provide a review of recent studies in §2.1.3, while a more in depth review of the subject was written by Gander (2015). A comprehensive record of research in the area is maintained at parallel-in-time.org.

1.2.1 Parareal

Parareal is an iterative method for solving initial value problems (IVP's), and consists of a coarse method $\mathcal{G}$ (from the French *grossièr*, for coarse) and a fine method $\mathcal{F}$. $\mathcal{F}$ is the method that would be used in a time serial simulation, whilst $\mathcal{G}$ is a quicker, computationally cheaper method which computes a less accurate approximation to the solution. The simulation time domain is partitioned into $\mathrm{NP}_{\mathrm{Parareal}}$ time slices,

Method	Operation	Notes
Parareal	Iterative use of coarse + fine solvers, linked through amplitude correction	Poor for advection dominated problems. Efficiency limited by $1/k$. Non-intrusive. 'Works' on all problem classes.
MGRIT	Adapts ideas of multigrid to time domain, using F- and V- cycles, coarsening in time. Full approximation scheme allows use in nonlinear problems.	Similar properties to Parareal- 2 level MGRIT with FAS is equivalent to Parareal. Non-intrusive. 'Works' on all problem classes.
PFASST	Replaces timestepper with SDC. Iterates sweeps of coarse and fine solvers, with FAS allowing fine data to improve the coarse correction.	Intrusive - have to replace time stepper. Efficiency not limited by $1/k$. 'Works' on all problem classes.
ParaEXP	Overlapping time intervals, each computes short amount of inhomogeneous time stepping, then solves the rest of the time domain by exponential integration of the homogeneous simplified problem. Sum of all results gives final state.	Only works well for linear problems. Intrusive - time stepper replaced.
REXI	Parallelises computation of the matrix exponential through rational approximation.	Only works for linear problems. Extension to non-linear not straightforward. Intrusive.
RIDC	Parallelises each timestep, by using modified SDC formulation such that all stages can be computed in parallel. Similar idea to parallel high-order Runge-Kutta methods.	Small scale parallelisation - on the same machine. Intrusive - replaces time-stepper.

Table 1.1: Summary of features for various Parallel-in-time methods. For more details, discussion, and references, see §2.1.2.

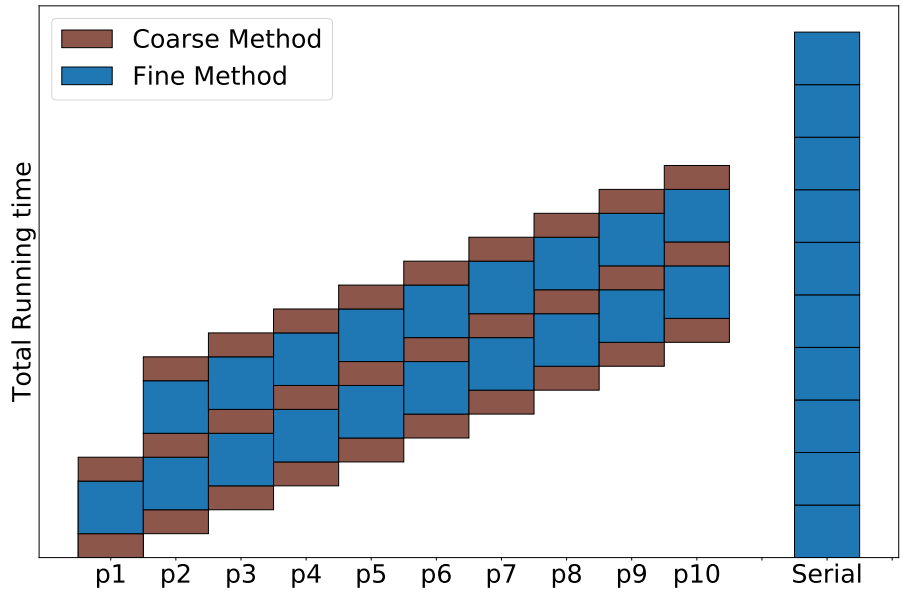


Figure 1.2: Processing time for each processor in Parareal (left) compared with processing time for the same simulation carried out in serial (right). Note that here we have two iterations, and the coarse stepper takes around half the time of the fine stepper.

with each slice being computed by a single processor (or set of spatially parallelised processors). The coarse method runs in serial, and is initially used to generate initial conditions for each process. The computation then consists of alternating between parallel runs of the fine solver, and serial runs of the coarse solver, which communicates information through the time domain using the Parareal correction

$$\phi_{n+1}^{k+1} = \mathcal{G}(t_{n+1}, t_n, \phi_n^{k+1}) + \mathcal{F}(t_{n+1}, t_n, \phi_n^k) - \mathcal{G}(t_{n+1}, t_n, \phi_n^k), \tag{1.1}$$

where ϕ_n^k is the solution at time slice n and iteration k, and t is time. The coarse sweep is the serial bottleneck in Parareal - each processor must wait for the previous time step to complete its correction before carrying out its own coarse sweep. When the change in the solution between two subsequent iterations is deemed small enough (see §3.3 for details), the simulation is deemed to be converged, and stops. Figure 1.2 shows a rough sketch of how the computing time of each processor is spent using Parareal compared to a serial computation, whilst figure 1.3 is a graphical representation of communication between the coarse and fine solvers.

Gaining good performance in Parareal depends on finding a very accurate coarse solver that allows for convergence in very few iterations (performance is bounded by $\mathrm{NP}_{\mathrm{Parareal}}/k$), and a very cheap coarse solver that is much quicker to compute than the standard serial solver (performance is also bounded by $R_{\mathcal{F}}/R_{\mathcal{G}}$, where $R_{\mathcal{F}}$ and $R_{\mathcal{G}}$ are runtimes of the fine and coarse methods over the same du-

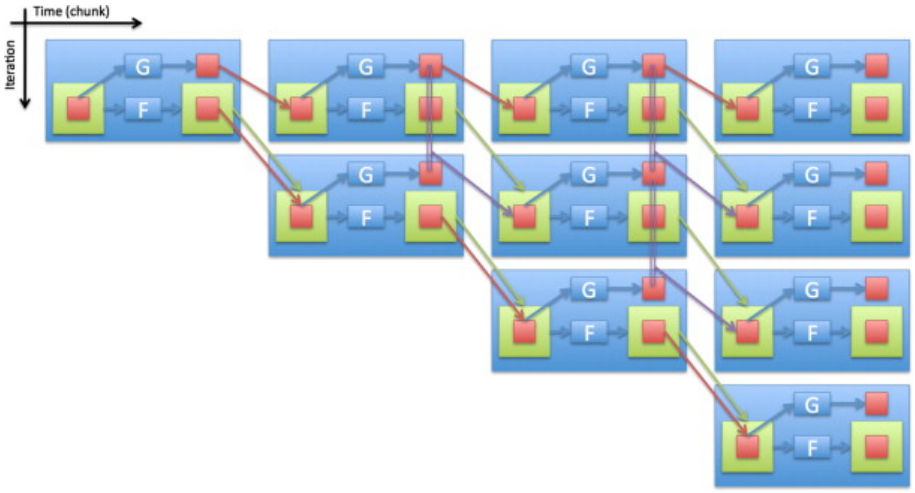


Figure 1.3: Graphical representation of the Parareal algorithm, with parareal correction updates shown in red. Reprinted from Berry et al. (2012), Copyright (2012), with permission from Elsevier.

ration) (Gander and Vandewalle, 2007). These two factors often act in direct opposition to each other, and so a balance must be found between performance and accuracy of the coarse solver. A number of strategies have been employed for finding a coarse solver. The most simple to implement is to simply increase the time step (e.g. Croce et al., 2014; Aubanel, 2011), though this is often not possible due to numerical stability. Changing the timestepping algorithm, such as replacing a high order explicit scheme with a low order implicit scheme can facilitate much larger timesteps (Blouza et al., 2011), though each step is typically more computationally expensive. Using simpler physics in the coarse solver has also been investigated (e.g. Baffico et al., 2002; Maday and Turinici, 2003; Maday et al., 2007). A further strategy is to reduce the spatial resolution along with the time resolution, which allows for relaxation of the CFL condition for the coarse step (e.g. Ruprecht, 2014; Lunet et al., 2018). When using this latter approach, the method of spatial coarsening is important. Lunet et al. (2018) found that low order interpolation and restriction led to a higher number of iterations being required for Parareal to converge, compared with higher order interpolation. Despite a number of investigations into convergence criteria (e.g. Bal, 2005; Gander and Vandewalle, 2007), the effectiveness of a coarse method is, in practice, determined by testing (Berry et al., 2012). This is complicated by Parareal's stability being more complicated than might be expected; a stable coarse solver paired with a stable fine solver can lead to instability in Parareal (Steiner et al., 2015), whilst Buvoli and Minion (2020) found that stability regions of Parareal when using IMEX (see §3.2.2) time stepping were non-trivial for dispersive problems.

Parareal has not been designed to achieve high parallel efficiency (Gander

and Güttel, 2013), rather, it is aimed at reducing time to solution where spatial parallelisation has already saturated. This is highlighted by the bound on Parareal's efficiency of the reciprocal of the number of iterations required (Gander and Vandewalle, 2007). Parareal performs badly for purely hyperbolic systems (Steiner et al., 2015) due to the Parareal correction step reducing defects in amplitude, rather than defects in frequency or phase of the solution. Despite this, there are many examples of Parareal being applied successfully to advection dominated problems such as high Reynolds fluid flows (e.g. Croce et al., 2014; Eghbal et al., 2017), MHD plasma flows (Samaddar et al., 2010, 2017), and convection (Samuel, 2012). This means that we cannot say in advance how well Parareal will perform for the dynamo problem. The dynamo problem is non-linear, so may suffer from poor performance as the Reynolds number is increased and advection dominates, but as successful applications to MHD and complicated systems have been reported, the method may still be of use.

Parareal can be useful as a tool to give insights into potential performance of other Parallel-in-time methods (e.g. PITA, MGRIT, PFASST) due to it sharing similar features (Kreienbuehl et al., 2015). For example, PFASST can be cast as a combination of Parareal and Spectral Deferred Corrections (SDC), while MGRIT with two levels is identical to Parareal (see table 1.2 and §2.1.2 for more details of other PINT methods). Communication overhead in Parareal is minimal, as information need only be exchanged once every iteration (Aubanel, 2011), making it scalable and highly suitable for the MPI parallel computing paradigm (Eghbal et al., 2017). An advantage of Parareal over other PINT methods is that it is non-intrusive. Parareal and MGRIT are the only time parallel algorithms that can be classed as such (Friedhoff et al., 2019), which means that existing codes can be used without major modification. Other algorithms can require substantial changes, for example PFASST requires that the existing timestepping algorithm is replaced with spectral deferred corrections. Parareal does not require modification of sequential application code, and can be used as a wrapper around existing simulation software (Aubanel, 2011). The non-intrusiveness has been exploited with studies using a wide array of different spatial discretisations in conjunction with Parareal, including finite difference (e.g. Croce et al., 2014; Lunet et al., 2018), finite volume (e.g. Steiner et al., 2015), finite element (e.g. Fischer et al., 2005), spectral methods (e.g. Samaddar et al., 2010), Monte-Carlo method (Samaddar et al., 2017), whilst Parareal has also been used in conjunction with RANS turbulence models and detached eddy simulations (DES) (Eghbal et al., 2017). We aim to make use of this non-intrusiveness in our work, for two reasons. The main reason is that if Parallel-in-time methods are to be useful to groups carrying out numerical dynamo simulations, the new methods should not require current codes to be significantly changed. Secondly, it will allow

us to choose an efficient, established numerical solver for simulations in this thesis.

1.3 Dynamo problem

1.3.1 Introduction/Overview

The dynamo problem seeks to answer the question of where the magnetic fields of the Earth, planets, Sun and stars come from. The idea that solar magnetic fields were generated by the flows of conducting fluids was first put forward by Larmor (1919). After the apparent setbacks to the theory caused by the work of Cowling (1933) on anti-dynamo theorems, it was not until Backus (1958) that the first working examples of fluid dynamos were found (Rincon, 2019). It is now widely accepted that planetary and stellar magnetic fields are caused by dynamo action (e.g. Braginsky and Roberts, 1995; Glatzmaier and Roberts, 1998; Cattaneo and Hughes, 2006; Gubbins, 2008; Tobias, 2021).

The equations governing dynamo action (e.g. Davidson, 2001; Tobias, 2021), and Oberbeck-Boussinesq equations modelling convection (Chandrasekhar, 1961) can be combined into the standard benchmark dynamo model (Jones, 2011; Christensen and Wicht, 2015), which will be used here to enable discussion of important parameter spaces for natural dynamos and their simulations. The models and non-dimensionalisations to be used in this thesis will be discussed in §3.1; we include the following to aid discussion. The main equations are

$$\mathrm{Ek}\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}\right)=-\boldsymbol{\nabla}p+\frac{1}{\mathrm{Pm}}\left(\boldsymbol{\nabla}\times\boldsymbol{B}\right)\times\boldsymbol{B} + \mathrm{Ek}\nabla^2\boldsymbol{u}+\mathrm{Ra_m}T\frac{\boldsymbol{r}}{r_0}-2\left(\hat{\boldsymbol{z}}\times\boldsymbol{u}\right), \tag{1.2}$$

$$\frac{\partial \boldsymbol{B}}{\partial t}-\boldsymbol{\nabla}\times\left(\boldsymbol{u}\times\boldsymbol{B}\right)=\frac{1}{\mathrm{Pm}}\nabla^2\boldsymbol{B}, \tag{1.3}$$

$$\frac{\partial T}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}T=\frac{1}{\mathrm{Pr}}\nabla^2T. \tag{1.4}$$

Here, $\boldsymbol{u}$ is the velocity, t is time, p is the pressure, $\boldsymbol{B}$ is the magnetic field, $\boldsymbol{r}$ is radial position vector, r_0 and r_i are the distances of the outer and inner radii, T is the temperature, and $\hat{\boldsymbol{z}}$ is the unit vector in direction of rotation axis. Ek, Pm, $\mathrm{Ra_m}$ and Pr are the Ekman, magnetic Prandtl, modified Rayleigh, and Prandtl numbers

respectively, defined by

$$\mathrm{Ek} = \frac{\nu}{\Omega D^2} \qquad \mathrm{Pm} = \frac{\nu}{\eta} \qquad \mathrm{Ra_m} = \frac{\alpha g_0 \Delta T D}{\nu \Omega} \qquad \mathrm{Pr} = \frac{\nu}{\kappa}. \tag{1.5}$$

Above, ν is the kinematic viscosity, Ω is the rotation rate, $D = r_o - r_i$ is the thickness between the inner and outer radii, η is the magnetic diffusivity, α is the coefficient of thermal expansion, g_0 is gravity at outer radius, ΔT is the temperature difference between inner and outer radii, and κ is the thermal diffusivity. Some other non-dimensional groups are common in the literature, as different formulations can be found, an important one to note here is the magnetic Reynolds number

$$\mathrm{Rm} = \frac{UL}{\eta}, \tag{1.6}$$

where $L = D$ is a characteristic length scale, and U is a characteristic velocity. Rm measures the relative importance of magnetic advection and magnetic diffusion.

Equation (1.2) is the momentum equation, with the Lorentz Force ($(\boldsymbol{\nabla} \times \boldsymbol{B}) \times \boldsymbol{B}$), thermally driven buoyancy ($\mathrm{Ra_m} T \boldsymbol{r}/r_0$), and Coriolis ($2(\hat{\boldsymbol{z}} \times \boldsymbol{u})$) terms included, (1.3) is the induction equation, and (1.4) is the temperature equation. When relevant boundary conditions are enforced (e.g. see §3.1.3), this system sees a temperature difference drive convective fluid motions through the buoyancy term. These fluid motions, if strong enough, can (though a number of conditions need to be met, see §2.2.3) generate a magnetic field through the induction equation. This magnetic field, once it has grown sufficiently, then affects the flow through the Lorentz Force, which is an important factor in virtually all astrophysical dynamos (Brandenburg and Subramanian, 2005). The three mechanisms described by the buoyancy term, induction equation, and Lorentz force can be seen as the basis for three major branches of study: fluid flow generation through convection, exponential growth of magnetic fields due to fluid motions (the kinematic dynamo problem), and fully nonlinear dynamo action, where effects such as saturation (where growth of the magnetic field is inhibited by nonlinear effects) become important factors. Rotation is an extremely important factor in all three of these mechanisms, but will be neglected in this work.

When tackling the problem of nonlinear magnetohydrodynamic dynamos, Glatzmaier and Roberts (1998) state 'numerical integration is generally the only way forward', and we will focus on numerical methods in this work. At the high Reynolds numbers of most astro- and geo- physical bodies there are no analytical solutions. The numerical problem itself is also incredibly difficult, however. As an example of how difficult, Tobias (2021) reports calculations by P. Käpylä that show an accurate

Parameter	Earth	Sun	Simulations
Length Scale	$\sim$0.1m - 7×10^6m	$\sim$1cm $- 7 \times 10^8$m	-
Time Scales	1 day - millions of years	Hours - $\sim 10^9$ years	-
Ek	$\sim 10^{-15}$	$\sim 10^{-15} - 10^{-13}$	10^{-7} - 10^{-8}
Re	$\sim 10^9$	$\sim 10^{12} - 10^{14}$	$\sim 10^3 - 10^4$
Pr	$\sim$1	$\sim 10^{-7}$	$\sim$1
Rm	$\sim 10^3$	$\sim 10^6 - 10^{10}$	$\sim 10^3$
Ra	$\sim 10^{23}$	$\sim 10^{16} - 10^{20}$	$\sim 10^{12} - 10^{15}$

Table 1.2: Typical time/length scales of the dynamo problem, along with typical values of nondimensional numbers in the Earth, Sun and simulations. References can be found in the text. Ek, Re, Pr, Rm, and Ra are the Ekman, Reynolds, Prandtl, magnetic Reynolds, and Rayleigh numbers, respectively.

simulation of our solar dynamo would require 10^{22} W, or the power output of a small star. The problem is intrinsically difficult to simulate, due to the vast range of length and timescales which are present in geo- and astro- physical dynamos. The lengthscales relevant in the earth range from the thickness of boundary layers (~ 0.1m) to the diameter of the core ($\sim 7 \times 10^6$m) (Matsui et al., 2016), while timescales range from the rotation period of one day, through the magnetic diffusion time (around 25,000 years), the reversal timescale of a few hundred thousand years (Davies et al., 2011), to the changes in reversal frequency which happen over millions of years (Biggin et al., 2012). The sun also exhibits a vast range of timescales, with the most well documented being the 11 (or 22) year cycle (Ossendrijver, 2003; Hathaway, 2015; Tobias, 2021). The life time of solar granules and super granules are of the order of hours (Muller et al., 1992; DeRosa and Toomre, 2004), whilst the equatorial rotation period is 26 days (Priest, 1982), and the diffusive timescale is $\sim 10^9$ years (Tobias, 2021). The length scales of the sun range from it's radius of 6.96×10^8m through to the typical width of a sunspot (3×10^7m (Priest, 1982)), the size of granules and super granules ($\sim 10^6$ m, (DeRosa and Toomre, 2004)), down to turbulent length scales, as the internal motions are highly turbulent (Brun and Browning, 2017). The Kolmogorov scale in the bulk of the solar convection zone is estimated to be $\approx$ 1cm (Kupka and Muthsam, 2017).

Because of numerical difficulties already mentioned, and because direct numerical simulations remain probably the most important tool to further understanding of the process, they are run with non-dimensional parameters which are far from those found in the systems we wish to study. The Ekman number, which determines the importance of rotation, is thought to be $\sim 10^{-15} - 10^{-13}$ in the Sun

(Jones et al., 2010), and $\sim 10^{-15}$ in the Earth, far away from values of $\sim 10^{-7}$ found in the most recent simulations (Schaeffer et al., 2017). Pm, which determines the relative importance of viscous/magnetic diffusion, is around six orders of magnitude too large in simulations of the Earth. The Reynolds number (which measures the level of turbulence) is $\sim 10^{9} - 10^{14}$ in stars (Dubrulle, 2008, 2011), $\sim 10^{12} - 10^{14}$ in the Sun (Jones et al., 2010), $\sim 10^{9}$ in Earth's core, and $\sim 10^{3}$ in DNS simulations (King and Buffett, 2013), though has reached $\sim 10^{4}$ in quasi-DNS simulations (Aubert, 2019). Pr, which compares the relative kinematic and thermal diffusivities, is thought to be $\sim 10^{-7}$ in the Sun (Jones et al., 2010), ~ 1 in the Earth, and ~ 1 in simulations (e.g. Matsui et al., 2016; Schaeffer et al., 2017). Rm is $\sim 10^{6} - 10^{10}$ in the Sun (Jones and Roberts, 2000), $\sim 10^{3}$ in the Earth's core (Davies et al., 2015) and 10^{3} in global simulations of the Earth/Sun, whilst more simplified studies have reached far higher Rm in (for example) 2.5D simulations. The Rayleigh number determines how much thermal driving is present, and is $\sim 10^{16} - 10^{20}$ in the Sun (Jones et al., 2010), $\sim 10^{23}$ in the Earth (Christensen and Aubert, 2006; Jones, 2007), and up to $\sim 10^{12}$ in state of the art dynamo simulations, whilst pure Rayleigh-Bénard convection studies have reached 10^{15} (Iyer et al., 2020). The above parameters are summarised in table 1.2, a summary of some important numerical dynamo studies is included at table 1.3, whilst a review and further details can be found in §2.2. The progression of dynamo studies over time seen in table 1.3 show an increase in resolutions due to extra available computing power. These increases in resolution allow for more Earth-like parameters to be simulated. Keeping long in-simulation durations for these studies (in order to study long term behaviour) becomes a challenge as the timestep size has to reduce with increased resolution, and very long runtimes are required, even when thousands of CPU's are employed.

In an effort to reduce the inherent difficulty in numerically simulating the dynamo problem, methods have been used which aim to reduce the required numerical resolutions. Hyperdiffusion replaces the diffusive terms of the momentum and induction equations with terms of the form $(\nabla^2 - \phi^2 \nabla^4)$, which, for small constant ϕ acts as a filter, damping the smallest scales (Jones, 2008). This raises a number of issues; it is usually in practice applied only in specific directions, introducing artificial anisotropy (Christensen and Wicht, 2015), whilst it can also wipe out small-scale behaviour, on which dynamo action can depend (Jones, 2008). It is also argued that any misrepresentation of magnetic diffusion, such as hyperdiffusion, numerical fixes or using the kind of artificial numerical diffusion inherent in (for example) finite difference schemes, should be avoided in dynamo calculations

Ref	Spatial method	Approx Resolution	Selected Parameters
Glatzmaier and Roberts (1995b)	Spherical harmonics and Chebyshev. No inertia.	49 x 32 x 64	40,000 years, Pr = 5000, 2000 CPU-hours
Christensen et al. (2001)	Various.	250 x 92 x 44	15 viscous diffusion times, Ek = 10^{-3}, Pr = 1, Ra = $2\mathrm{Ra}_C$ (Ra modified for rotation).
Sreenivasan and Jones (2006)	Spherical harmonics and finite difference.	60 x 48 x 60	Ek = 10^{-4}, Ra = 750 (modified Ra, using magnetic diffusivity)
Christensen and Aubert (2006)	Spherical Harmonics and Chebyshev.	97 x 224 x 224	Ek = 3×10^{-6}, 50 turnover times,
Kageyama et al. (2008)	Yin-Yang finite difference.	511, 514, 1538	Ek = 2.3×10^{-7}, 4096 processors, ran for 430 sound wave crossing times,
Sakuraba and Roberts (2009)	Spherical harmonics and Chebyshev.	255, 255, 160	Ek = 5×10^{-7}, Ra = 3.2×10^{10}.
Miyagoshi et al. (2010)	Finite Difference, yin-yang	511, 514, 1538	Ek = 2.3×10^{-7}, 0.17 magnetic diffusion times.
Matsui et al. (2016)	Various.	up to 512, 385, 769	Ek = 10^{-3}, (modified) Ra = 100, Pr = 1, Pm = 5.
Aubert et al. (2017)	Spherical harmonics and Chebyshev, hyperdiffusion.	2496, 640, 640	Ek = 10^{-8}
Schaeffer et al. (2017)	Spherical harmonics and finite difference, DNS.	1280, 1504, 2688	Ek = 10^{-7}. Used up to 10 million CPU-hours.
Earth - Davies et al. (2011)	-	-	Ek = 10^{-9}, Ra = $10^3\mathrm{Ra}_C$, 13,392 days on 54,000 CPU's for one magnetic diffusion time.

Table 1.3: Summary of selected dynamo studies. Resolutions are for number of points in radius, colatitude, and azimuth, respectively. For more details see §2.2.

as there is a strong chance it will lead to incorrect results (Tobias, 2021). Using standard diffusion, rather than hyperdiffusion can ensure unbiased dynamics at all scales (Schaeffer et al., 2017). The use of hyperdiffusion remains an open question however, with recent studies suggesting restriction of hyperdiffusion to the highest modes using a cut off give comparable results to pure DNS, as the smallest length scale affecting dynamo action scales as $\mathrm{Rm}^{-1/2}$ (Aubert et al., 2017; Aubert, 2019). Direct numerical simulations of dynamo systems with high spatial resolutions nevertheless remain a highly active area of research and are ultimately the benchmark for large eddy simulations.

The high spatial resolution required also affects the required time resolution through (for example) the CFL condition (Courant et al., 1928) (see §3.2.2 for further discussion). Higher spatial resolution requires smaller timestep sizes, the specific form of this restriction changes with the numerical method, but as a general rule the timestep must be small enough that variable properties do not cross more than one grid point in a single timestep. In addition to the CFL constraint on timesteps, the inherent large range of timescales present in geo- and astro- dynamo systems require lots of small timesteps to accurately resolve the time dimension. A quick calculation shows that one magnetic diffusion time of the earth with timesteps of a single day would require around 9 million timesteps!

For dynamo simulations, time stepping does not seem to have gained the same amount of attention as spatial discretisation in the literature. The work by Livermore (2007) seems to be the exception to this rule, while most other studies rarely comment on timestepping methods. For an example of this, the large scaling study of Matsui et al. (2016) did not investigate the performance effects of time stepping at all, using a fixed timestep even for codes where variable timestepping was available. Rather, most numerical improvements have been sought by optimising spatial discretisations or decompositions. This is understandable given the huge parameter space in numerical dynamo studies, but it does leave a gap in knowledge that it would be interesting and useful to study.

1.4 The Kinematic Dynamo

Due to computational difficulty of full dynamo studies, until recently, kinematic dynamo modelling was the workhorse of solar dynamo modelling (Charbonneau, 2020). As an example of the motivation behind the simplifications used in many of these simulations, pseudo-2D kinematic dynamos allow investigation into higher

Rm regimes by concentrating all numerical resolution and computing power into just two spatial dimensions (Rincon, 2019) (this is further discussed in §2.2.3). We begin by outlining the kinematic dynamo problem, before moving on to review some important theoretical and kinematic dynamo studies. For a comprehensive review of these subjects, see for example Rincon (2019) and Tobias (2021). The kinematic dynamo problem has been defined in different ways in the literature. The term most commonly refers to the problem of seeking pre-determined velocity fields ($\boldsymbol{u}$) which lead to a growing magnetic field ($\boldsymbol{B}$) in the (non-dimensional) induction equation

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) + \frac{1}{\mathrm{Rm}} \nabla^2 \boldsymbol{B} \tag{1.7}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0, \tag{1.8}$$

where the magnetic Reynolds number Rm determines the relative importance of magnetic advection/diffusion (Roberts, 1967). Kinematic dynamos have also been used as the description of studies which include the fluid flow equations, but sets the Lorentz force $\boldsymbol{j} \times \boldsymbol{B}$ to zero. The early stage of fully non-linear dynamo studies is also often described as kinematic (e.g. Jones and Roberts, 2000). During this kinematic stage, magnetic energy is small compared to the kinetic energy of the fluid, so that the Lorentz force has little influence on the flow and so the magnetic field grows or decays exponentially, as in kinematic studies.

The induction equation is linear in $\boldsymbol{B}$, so a given velocity field will cause an initial small magnetic field to either grow or decay exponentially. The most commonly studied kinematic case is for a steady $\boldsymbol{u}$, so that, for a self-sustained dynamo in a bounded domain, solutions of the form

$$\boldsymbol{B} = \boldsymbol{B}_0 (x, y, z) \, \mathrm{e}^{\gamma t}, \quad \boldsymbol{B}_0 \to 0 \text{ as } \boldsymbol{x} \to \infty \tag{1.9}$$

can be sought (Jones, 2008). If a mode with positive growth rate γ exists, then the magnetic field will grow exponentially.

Further mathematical details of kinematic dynamos, such as a derivation of the induction equation can be found in §3.1, non-dimensionalisation is shown in §3.1.2. A review of recent kinematic dynamo literature, with details of anti-dynamo theorems and simplifying assumptions, can be found in §2.2.3.

1.4.1 Numerical Considerations

There are two main strategies for numerically solving the kinematic dynamo, timestepping and solving the eigenvalue problem directly. Both of these strategies typically employ spectral methods (Jones, 2008). Eigenvalue solution is most useful in the case of prescribed flows, whilst timestepping is more common in driven flow dynamos. Simulations are most often carried out in either spherical or Cartesian geometry. Fourier series and polynomials such as Chebyshev are employed in Cartesian geometry, whilst spherical harmonics are used for angular discretisation in spheres. In radius, research has been carried out to investigate the optimal choice of spectral basis (Livermore and Jackson, 2005; Livermore et al., 2007b), while Li et al. (2010) investigated the use of the Galerkin method using jacobi polynomials for the eigenvalue problem. Marti and Jackson (2016) investigated the use of Jones-Worland polynomials for the radial expansion of full sphere magnetohydrodynamic calculations, removing the artificial singularity found at the origin of similar discretisations. Gómez et al. (2005) introduced a parallel solver for MHD simulations using FFTW parallel transposes to allow high parallel efficiency, which has been used in a number of subsequent studies (e.g. Alexakis, 2011; Dallas and Tobias, 2018). Livermore (2007) performed benchmark testing of a number of timestepping schemes, concentrating on the implementation of the Exponential Time Differencing (ETD) which solves the linear (and in MHD stiff) part of the problem exactly. He found that the relatively small timestep size demanded by the nonlinear part of the problem meant that large accuracy gains were not realised, though ETD was thought likely to be more competitive at higher than 2nd order. Recent developments include Väisälä et al. (2021) who investigated the use of GPU's to speed up simulations of MHD flows, though this was used in conjunction with the finite difference PENCIL code.

1.5 Rayleigh-Bénard Convection

The dynamos of the Earth and Sun are believed to be driven by convective fluid flows (Busse, 2002; Jones et al., 2011), so a full understanding of the dynamo problem requires investigation into convection. The simplest and most well understood problem in this field is Rayleigh-Bénard convection (RBC), which describes the buoyancy driven flow of a Boussinesq fluid heated from below and cooled from above (Ahlers et al., 2009). Convection is used as the driving force in geo- and astro- physical dynamo models (e.g. Glatzmaier and Roberts, 1995a; Christensen et al., 2001; Matsui

et al., 2016; Glatzmaier, 1985; Käpylä et al., 2012; Dietrich and Jones, 2018). In order to better understand the underlying physical mechanisms behind convective dynamo action, dynamos driven by Rayleigh-Bénard convection have been extensively studied, for example see Childress and Soward (1972); Jones and Roberts (2000); Stellmach and Hansen (2004); Cattaneo and Hughes (2017). The historical and ongoing interest in dynamos driven by Rayleigh-Bénard convection further justifies starting investigations of novel numerical methods in dynamos with studies of RBC.

1.5.1 Introduction and definition of Problem

In this section, we will first define and outline the problem of Rayleigh-Bénard convection. We will introduce some of the important quantities applicable in the area, before looking at some of the open questions and reviewing some recent studies in the area. Mathematical details, such as non-dimensionalisation and details on calculating relevant quantities will be shown in §3.1.3.

RBC describes the fluid flow of a liquid driven by the temperature difference of two horizontal plates separated by a distance h. The bottom plate is ΔT hotter than the top plate, and local density changes caused by the temperature gradient cause convection to occur above a critical level, defined as the critical Rayleigh number (Ra_c). Figure 1.4 shows some representative snapshots of the temperature field in a typical 2D RBC flow, for a selection of Ra. The non-dimensional Oberbeck-Boussinesq equations modelling non-rotating Rayleigh - Bénard convection can be written as

$$\frac{1}{\mathrm{Pr}}\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}\right)=-\boldsymbol{\nabla}p+\mathrm{Ra}T\cdot\hat{\boldsymbol{z}}+\nabla^2\boldsymbol{u}, \tag{1.10}$$

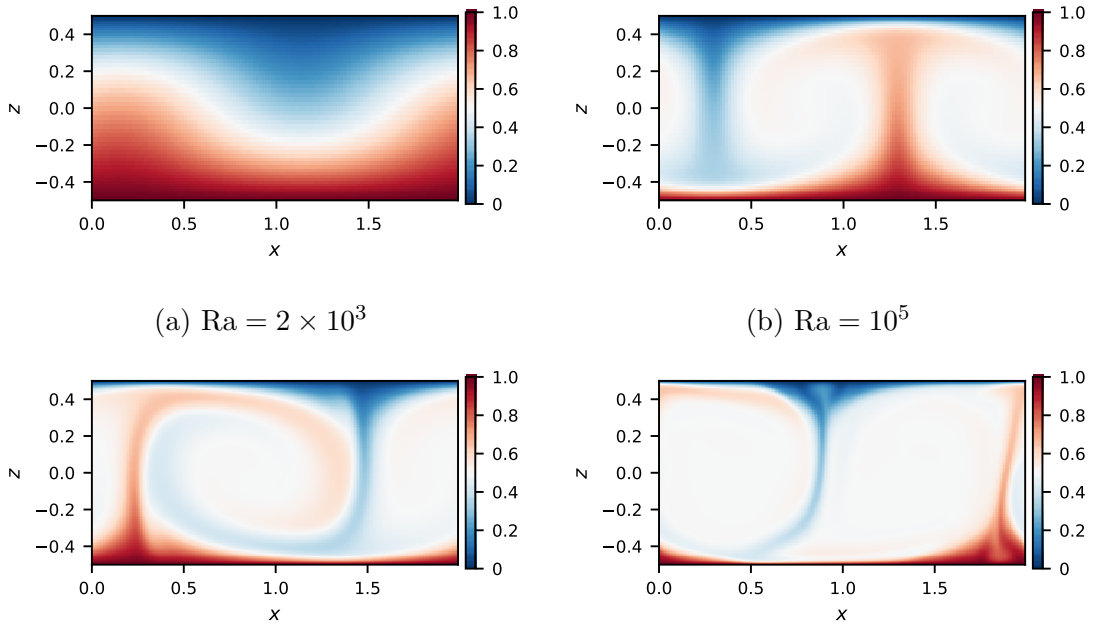
$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0, \tag{1.11}$$

$$\frac{\partial T}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}T=\nabla^2 T, \tag{1.12}$$

with suitable boundary conditions required for velocity and temperature. Here, $\hat{\boldsymbol{z}}$ is the unit vector in the vertical and Ra is the standard Rayleigh number

$$\mathrm{Ra}=\frac{\alpha g h^3 \Delta T}{\nu\kappa}. \tag{1.13}$$

The Boussinesq approximation is made, which ignores the effects of variations in



(a) Ra $= 2 \times 10^3$

(b) Ra $= 10^5$

(c) Ra $= 2 \times 10^6$

(d) Ra $= 10^7$

Figure 1.4: Representative temperature field snapshots of Rayleigh-Bénard Convection at different Rayleigh numbers (Ra).

density except for in the buoyancy term (Bodenschatz et al., 2000). Chandrasekhar (1961) used linear theory to calculate the critical Rayleigh number (Ra_c) required for convection to occur, for a variety of boundary conditions, for example, with rigid boundaries, $Ra_c = 1707.762$. The Nusselt number (Nu) measures convective heat transport, which has a minimum of 1, representing purely conductive heat transfer. Figure 1.5 shows the horizontally averaged temperature profile for the conductive state (Ra $< Ra_c$), and for increasing values of Ra, showing that temperature gradients become sharper at boundaries, with temperatures in the bulk of the fluid becoming more mixed, as Ra is increased. The relationship between the heat transfer (represented by Nu) and the amount of thermal driving (represented by Ra) is a key question of convective studies. We give a brief insight into the issues here, for more in depth discussion see for example Grossmann and Lohse (2000); Ahlers et al. (2009); Plumley and Julien (2019). Relationships of the form

$$\mathrm{Nu} \sim \mathrm{Ra}^{\gamma_{\mathrm{Nu}}} \tag{1.14}$$

are sought, with values of the exponent γ_{Nu} most commonly given as $\approx 2/7$ (e.g. Castaing et al., 1989; Shraiman and Siggia, 1990; Cioni et al., 1997) for moderate Ra ($\leqslant 10^{10}$, (Long, 2020)), with a transition to the classical scaling law $\mathrm{Nu} \sim \mathrm{Ra}^{1/3}$ for Ra $> 10^{10}$, evidenced by theory (Malkus, 1954), experiment (e.g. Ahlers et al., 2009), and numerical studies (e.g. Iyer et al., 2020).

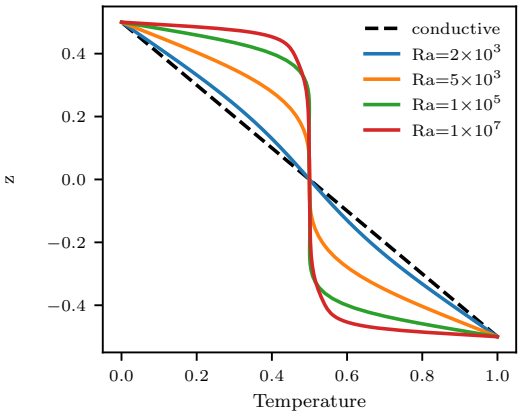


Figure 1.5: Horizontal- and time- averaged temperature profiles plotted against vertical coordinate for different Ra, compared with the conductive background state.

Although we have approached the problem of RBC as a simplified subset of the dynamo problem, RBC is a very active area of research with many open questions in its own right. One major debate is the point at which the ultimate regime (Grossmann and Lohse, 2000) of convection will be reached, whereby the boundary layers become fully turbulent and heat transfer is enhanced with $\mathrm{Nu} \sim \mathrm{Ra}^{1/2}$. Simulations using upto 2.2×10^{10} spatial mesh points were employed by Iyer et al. (2020) in an attempt to shed light on this issue, highlighting the extreme computational requirements of state of the art RBC simulations. In order to definitively find the point at which the ultimate regime is reached, simulations at higher and higher Ra will be required. Increased Ra requires increased spatial resolution and reduced timesteps, and so parallelisation in space and time would be useful tools to help reach these more extreme parameter ranges. Finding increased performance through Parareal for RBC simulations would thus be a useful result in its own right, in addition to the requirements for dynamo studies.

Much work has been done to attempt to speed up simulations of RBC flows, primarily to enable simulations at higher and higher Ra. A standard method of speeding up simulations of RBC is to 'bootstrap' using the statistically steady state of a simulation at a lower Ra as the initial condition for higher Ra studies (e.g. Verzicco and Camussi, 1999; Johnston and Doering, 2009; Long et al., 2020). This has the effect of reducing the 'transients' whereby the flow initially overshoots after growing from small scale instabilities, and reduces the simulation duration required to obtain a statistically stable state. This can reduce computational time considerably, but has limitations. A configuration with more than one statistically stable state may be forced into a particular state by the 'imprinting' of previous lower energy solutions, whilst it may have reached a different state if started from

different initial conditions (Anders et al., 2018). Other methods to speed up studies of convection include attempting to create initial conditions likely to be close to the stable state of the system, by (for example) solving a simpler problem such as an axisymmetric field in the case of a 3D cylinder (Verzicco and Camussi, 1999), or linear eigenvalue solution in the case of a plane layer (Hurlburt et al., 1984). Anders et al. (2018), extending ideas of Hurlburt et al. (e.g 1986), investigated techniques to accelerate the evolution of simulations by adjusting the mean thermodynamic profile within the domain. Kooij et al. (2015) investigated the speeds of different numerical codes for studies of RBC. In Kooij (2017) work on making the parallel in time algorithm Paraexp work in a nonlinear setting was carried out with hope of eventually applying this method to RBC and similar flows. The Paraexp-EBK (exponential block Krylov) method combined Paraexp with a waveform relaxation method to allow non-linear PDE's to be solved, and found that limited speedup is possible.

Understanding of many aspects of RBC has been sought through studies in the literature. The relevance of 2D RBC studies to more realistic 3D studies was investigated by Schmalzl et al. (2004) and later by van der Poel et al. (2013), who found that moderate to high Pr flows showed good agreement between 2D- and 3D- simulations, but that at lower Pr (< 1), significant differences emerged. The effects of rotation have also been investigated thoroughly. King et al. (2013) looked at three regimes, non-rotating, weakly rotating and rotationally constrained. They found that even in weakly rotating fluids the viscous boundary layer thickness (gets smaller/bigger) and horizontal length scales (get smaller) are affected by rotation. In the rotationally constrained regime, the thermal boundary layer thickness, mean temperature gradients, and flow speeds are also affected by rotation. Johnston and Doering (2009) investigated the differences between fixed temperature and fixed flux boundary conditions, finding that as Ra increases, the heat transfer, represented by Nu became equivalent in both cases. Pandey et al. (2018) studied long-term numerical investigations at very low prandtl number to resolve the highly inertial turbulence properly using Nek5000. Stevens et al. (2018) investigated the generation of thermal superstructures, by simulating a domain with very high aspect ratio, reducing the chances of horizontal structures being limited by the domain size.

1.6 Discussion

We have identified that there is a need for more parallel performance for dynamo simulations. We have further identified that the time domain would be a good place

to look, due to the combination of very short and very long timescales, along with the CFL restriction becoming prohibitive as spatial resolution increases. We seek to choose which Parallel-in-time method to use first as a prototype for the dynamo problem. Further, we need to find a problem with enough characteristics of the dynamo that results will give meaningful information for the full dynamo problem, whilst being computationally cheap enough to run many simulations, and simple enough that results can be analysed usefully.

One of most important things in choosing a PINT method is non-intrusiveness (Friedhoff et al., 2019). In this regard, MGRIT and Parareal are preferred options (over PFASST, which replaces the time stepper with SDC, and RIDC, which replaces it with a parallel alternative). The applicability of the method is also very important. Parareal, MGRIT, PFASST and RIDC are all quite general methods, which can be applied to linear and non-linear problems, and have been coupled with many types of spatial solver. REXI, ParaEXP and others, whilst perhaps offering better performance in advection dominated problems, are only applicable to linear problems; until well tested non-linear versions of these algorithms are available, their use for the dynamo problem will likely be limited. The potential performance of a method should also be taken into account. However, no comprehensive study, comparing performance of parallel-in-time methods has been carried out, due to it's inherent difficulty (see §2.1.2 for discussion). Since Parareal is equivalent to MGRIT for nonlinear 2-level setups, the use of Parareal in the first instance is justified, and if good performance is found, further testing with MGRIT or PFASST would be a logical next step.

The full nonlinear dynamo problem would be a poor choice for a first investigation into parallel-in-time methods. The high degree of non-linearity, epitomised by the problem of saturation (see §2.2.3 for details) means that it would be difficult to pick a benchmark problem. There are also a large number of control parameters, and observing how performance changed with each parameter would be very time consuming. Simulations would also be very expensive, both in terms of CPU-hours and in real time durations. Given a proper study will require multiple runs and scaling tests, the simulations should be as cheap as reasonably possible. The kinematic dynamo is thus a good choice. Examples of 2.5D dynamos (introduced in §2.2.3), which can be simulated in just two dimensions, reduces the computational complexity, whilst complex behaviour of the magnetic field is still present.

RBC has been shown to be an interesting and difficult problem in its own right. Much research has been carried out to find the best possible numerical methods, with a benchmark by Kooij et al. (2015), and work by (for example) Anders

et al. (2018) on finding quicker ways of obtaining thermally relaxed states. Assessing the potential of parallel-in-time methods for Rayleigh-Bénard convection is thus useful, and timely, given the current debate around the ultimate regime and recent work on improved numerical methods. This is in addition to its strong link to the dynamo problem. Convection is an important part of the dynamo problem, but removes some of the complexity, such as the problem of saturation, and it can be solved in fully 2D simulations, allowing for many scaling tests to be carried out in a reasonable amount of CPU-hours and real time.

1.7 Aims and Objectives

The overall aim of this thesis is to see how much parallel speed up and efficiency can be obtained when using Parareal to speed up simulations of the kinematic dynamo and Rayleigh-Bénard convection.

The questions we seek to answer in this work are as follows:

1. Can Parareal speed up simulations of kinematic dynamos?
   (a) Confirm that Parareal can accurately reproduce numerical results, e.g. growth rates, magnetic field morphology.
   (b) Investigate the amount of speed up and parallel efficiency that can be obtained using Parareal for the kinematic dynamo.
   (c) Investigate how Parareal performance changes with Rm - over what range of values is Parareal useful?
2. Can Parareal speed up simulations of Rayleigh-Bénard convection?
   (a) Confirm that Parareal can accurately reproduce numerical results, e.g. nusselt number, growth rates, critical Rayleigh number.
   (b) See how much speed up and efficiency can be obtained using Parareal for Rayleigh-Bénard Convection.
   (c) Investigate how Parareal performance changes with Ra - over what range is it useful - will it help at very high Ra?

In Chapter 2 we provide a survey of recent literature in the areas of dynamo simulations and Parallel-in-time methods. Chapter 3 explores the mathematical background of dynamos, convection, spectral methods, timestepping and Parareal. In Chapter 4 we report results of our work on the kinematic dynamo, and Chapter 5 contains results of our work on Rayleigh-Bénard convection. Chapter 6 provides a discussion of our results and possible future work.

Chapter 2

Literature Survey

In this chapter, we give a comprehensive background of Parallel-in-time methods and the dynamo problem. We do this through a review of recent literature, beginning with Parareal and other parallel-in-time methods. We then review numerical studies of the dynamo problem, in geo- and astro- physical settings, followed by theoretical studies of dynamo action.

2.1 Parallel-in-time

We will begin by looking at modifications that have been made to Parareal, to attempt to alleviate some of the shortcomings discussed in §1.2. We will then introduce and describe a number of other parallel-in-time methods. Finally, we will review a number of recent studies carried out using parallel-in-time methods, concentrating on reported speed-ups. To highlight the number of diverse applications of Parallel-in-time methods, we organise the review in terms of application area.

2.1.1 Modifications to parareal

A number of modifications have been proposed to the Parareal algorithm. They can be broadly split into two types: changes to the algorithm to make it more effective for different types of problems, and optimisations to improve its performance. We will first discuss the algorithm changes. Farhat et al. (2006) introduces Parallel Implicit Time-integrator (PITA). Gander and Petcu (2008) uses a Krylov subspace built from previous coarse iterations to speed up convergence, and shows that PITA

and Krylov-subspace-enhanced Parareal (KSE-Parareal) are equivalent for linear problems. Ruprecht and Krause (2012) introduced a partially split method whilst investigating the feasibility of using KSE-Parareal for numerical weather prediction. Dai and Maday (2013) investigated a version of Parareal using projection on to a manifold which allowed the method to remain stable for first and second order hyperbolic problems. Gander et al. (2019) introduced Parareal with the ability to converge for problems with discontinuities, while Maday and Mula (2020) investigated adaptive Parareal accuracy, using low accuracy for early fine solvers, making them cheaper.

For optimisation changes to Parareal, we begin with Aubanel (2011), who introduced pipelined Parareal, allowing processors to start computing the fine solution as soon as they had finished the coarse solution for their own time slice, rather than waiting for the whole coarse run to be completed. Berry et al. (2012) introduced IPS event-based Parareal. This method allows the number of time slices to be different to the number of processors available for parallelisation in space. Two benefits of this approach are for fine and coarse solvers of the same time slice to compute concurrently, and reducing the amount of idle processor time. Nielsen and Hesthaven (2016) examined fault tolerance in Parareal, extending the work of Aubanel (2011). In Nielsen et al. (2018) the communication aware adaptive Parareal method was introduced, to balance processor utilization and convergence. Eghbal et al. (2017) used Parareal as an accelerator for CFD simulations, using a coarse run to generated initial conditions, several Parareal iterations to reduce transients, and a final fine run to find the statistically steady state.

PITA and KSE-Parareal offer some advantages over Parareal, however, the results obtained in the literature rely on the use of an implicit coarse propagator allowing for large coarse time-steps (Ruprecht and Krause, 2012), and the nonlinear version of PITA relies entirely on using an implicit coarse solver. The algorithm would thus require significant modification of existing codes that did not already have the option of fully implicit time stepping, like dynamo codes, which generally use IMEX timestepping methods, and would be very computationally expensive in (for example) spectral codes. A significant amount of time and effort has been spent optimising the numerical codes used in dynamo simulations, and initial studies of parallel-in-time methods should seek to be compatible with these codes, if widespread usage is to be obtained. The use of pipelined Parareal is relatively straightforward, and is now used in the majority of Parareal studies, and we use this version in our work. IPS offers extra computational efficiency in Parareal, and if initial studies of Parareal show promise, then switching to IPS for production runs would be the optimal choice. Fault tolerance can be safely ignored for small

to moderate numbers of processors, and should only become relevant in production runs at extremely high numbers of processors. The work of Eghbal et al. (2017) is very interesting, and if Parareal shows good convergence properties for the problems investigated, should definitely be investigated, as should the adaptive accuracy of the fine solver described by Maday and Mula (2020).

2.1.2 Other Pint methods

There are a number of other Parallel in time methods in the literature, and we will introduce a few below. For a more thorough history of Parallel-in-time methods, see for example Gander (2015). Multigrid reduction in time (MGRIT) (Friedhoff et al., 2012) uses the ideas of traditional multigrid solvers, using V- and F- cycles which coarsen and relax the solution in time rather than in space. MGRIT can be applied to non-linear problems by use of a full approximation scheme (FAS) to move data between different levels. Using MGRIT with a FAS and only two levels is equivalent to using the Parareal algorithm; Parareal can be seen as a special case of MGRIT using only two levels. PFASST (e.g. Minion and Williams, 2008; Emmett and Minion, 2012; Speck et al., 2012; Ruprecht et al., 2013) replaces the standard time stepper with spectral deferred corrections (SDC). SDC treats a time step as a Picard integral

$$\phi(t) = \phi_0 + \int_0^t f\left(\tau, \phi(\tau)\right) \mathrm{d}\tau, \tag{2.1}$$

using spectrally determined quadrature nodes (e.g. Gauss-Lobatto nodes) to compute accurate approximations to the time integral in an iterative process using a calculation for the residual. PFASST can be seen as using Parareal with SDC timestepping for both coarse and fine steppers, with the FAS used to couple the different levels. PFASST also allows information from fine levels to improve the coarse approximation. Intertwining the SDC sweeps on the different levels with the outer iteration allows PFASST to achieve better parallel efficiency than Parareal.

ParaEXP (Gander and Güttel, 2013) can allow parallel time integration for linear time stepping problems. Overlapping time intervals are used, and inhomogeneous problems are stepped through time in parallel over short time slices. The result from these intervals are used as initial conditions for homogeneous simplifications, which are computed to the end time very quickly using exponential propagation. Due to linearity, the solutions of each process can be summed at the end to obtain the full solution. A non-linear extension was proposed in Gander et al. (2017), which performed well for small non-linear terms, but became less efficient as the non-linear term became larger and started to dominate, which would

make their use in the dynamo problem limited. Kooij et al. (2018) used a pressure free projection of the Navier-Stokes equation to enable ParaEXP to be used, and estimated that moderate parallel speed ups could be possible, if communication time was neglected. REXI (rational approximation of exponential integrators) (Haut et al., 2015; Schreiber et al., 2018) seeks to solve time integration problems by parallelising computation of the matrix exponential through a rational approximation. The method is at present only applicable to linear hyperbolic systems (e.g. rotating shallow water equations) and extension of the method to non-linear equations is not necessarily straightforward. One proposed solution is to incorporate asymptotic Parareal (Haut and Wingate, 2014), which replaces $\mathcal{G}$ with a locally asymptotic approximation of the original PDE. RIDC (Christlieb et al., 2010) and parallel high-order Runge-Kutta methods (e.g. Van Der Houwen and Sommeijer, 1990; van der Houwen and Sommeijer, 1991; Sommeijer, 1993; Van der Houwen and Sommeijer, 1993) allow for parallelisation on multicore architectures for single timesteps. The usually sequential stages of (for example) Runge-Kutta integrations are reformulated such that they can be computed in parallel. They allow for smaller scale parallelisation on up to 12 processes in time (Ong and Schroder, 2020). A similar approach was combined with PFASST in Schöbel and Speck (2020) to create PFASST-ER (PFASST with Enhanced ConcuRency), which is parallelised over a single time step and across the entire time domain.

It is incredibly difficult to directly compare performance of parallel in time solvers - to the authors knowledge only one direct comparison of different PINT methods has been published - Benedusi et al. (2020). In fact, it is difficult to accurately determine the useful performance of even a single parallel-in-time method! A recent paper, "Twelve Ways to fool the masses when giving parallel-in-time results" (Goetschel et al., 2021) highlights a number of common problems. PINT speed ups are not always benchmarked against the best serial solver, e.g, for many PFASST studies, speed up is measured against serial SDC, when better serial timesteppers may be available. Different PINT methods have different strengths and weaknesses, so no suite of benchmark tests has been published in the area of parallel in time. For this reason, we concentrated mainly on usability rather than performance when choosing a parallel-in-time method in the first instance, making performance comparisons of different methods a secondary objective. A robust comparison of different algorithms on a complex problem would be a useful addition to the field, however.

2.1.3 Applications of Parallel-in-time

Parallel-in-time algorithms have been used in many application areas, we will discuss some of the highlights below, for recent review see for example Gander (2015); Ong and Schroder (2020). Early studies concentrated on convergence properties rather than speedups on high performance computers (HPC's), such as Baffico et al. (2002), who tested Parareal for a 32 particle molecular dynamics system. Satisfactory convergence was found within 4 iterations. Blouza et al. (2011) studied chemical kinetics in relation to ozone production, and found a speedup (how much quicker the parallel simulation is than the serial simulation) of 31x with 170 processors. Bal and Maday (2002) investigated the use of Parareal to speedup simulations of financial markets, concentrating on the Black-Scholes equations for an American put. They found a speedup of 6.25x using 50 processors. An investigation into European pricing options was carried out by Magoulès et al. (2018), finding a speedup of 3.5x with 64 processors. Parareal was applied to time-dependent Schrödinger equation in Maday and Turinici (2003), with an application to quantum control. A computational model estimated speedups of 10x would be possible.

Parallel-in-time methods have been applied in a diverse number of applications. Kreienbuehl et al. (2015) applied Parareal to the brick and mortar problem as an analogue for skin transport, finding speed-ups of 6x with 32 processors. The collapse of a black hole was successfully sped up by Parareal in Kreienbuehl et al. (2017), finding speed-ups of 30x using 128 cores. The study found Parareal reproduces Choptuik's black hole mass scaling law in an Einstein-Klein-Gordon system. Static mechanics were studied in Hessenthaler et al. (2018), where the oscillations of a linear beam attached to a wall were sped up by 5.25x with 192 processors using the MGRIT algorithm. Optimal control problems have been studied using Parareal (Maday et al., 2013), PFASST (Götschel and Minion, 2018), and MGRIT (Günther et al., 2018), while Gander et al. (2020) created the ParaOpt algorithm, based on Parareal, to deal with the same problem area. Simulations of a nuclear light water reactor were attempted by Baudron et al. (2014), a speedup of 2x with 10 processors was obtained using Parareal. Simulations used for robotic manipulations were sped up by Parareal in Agboh et al. (2019) using a simplified physics model for the coarse solver, whilst in Agboh et al. (2020) a deep neural network was used as the coarse solver in the same problem, reducing its computational complexity.

MGRIT has recently been applied to speedup the training of machine learning algorithms. Schroder (2017) parallelized over training epochs, with a model predicting potential speed-ups of over 6x, whilst Gunther et al. (2020) parallelised over

the layers. They achieved a training performance similar to that of traditional methods, but with layer-parallelism, generating a speed-up of 16x using 512 cores on the MNIST training data. Much work has been carried out on electrical systems. Gurrala et al. (2015) applied Parareal to simulations of the New England 10-generator 39-bus system, finding 7x speed-ups with 60 processors. Friedhoff et al. (2019) used MGRIT with the eddy current model in a copper wire, finding 3x speed-up with 128 processors, while Bolten et al. (2020) used the same method for simulations of a 2D induction machine, finding speed-ups of 21x with 256 processors. An N-body particle simulation of a moving vortex sheet was simulated using PFASST in Speck et al. (2012) and Speck et al. (2014), where the former found speedup of 6x over 32 time slices, allowing the spatially saturated PEPC Barnes-Hut tree code to scale out to 262,144 cores of the Blue Gene computer system.

A number of studies have investigated the ability of Parallel-in-time methods to speedup simulations of fluid flow. Fischer et al. (2005) was the first application of Parareal to the Navier-Stokes equations, looking at 2D flow past a cylinder at Reynolds numbers of $\mathrm{Re} = 200$ using finite elements and $\mathrm{Re} = 7500$ using spectral elements. In this work, convergence rather than speedup was measured, and convergence in only 2 or 3 iterations was found. Croce et al. (2014) performed scaling tests of Parareal for 3D cavity driven fluid flow at Reynolds numbers up to 1000, with speedup of over 35 obtained using 2048 cores, with 32 time slices. This simulation showed the potential of Parareal to provide additional speedup to problems after spatial parallelisation gains had saturated. Steiner et al. (2015) found that performance was reduced for lower viscosities in 2D fluid flow problems. The MGRIT algorithm was used in Falgout et al. (2015) for a compressible fluid flow problem, that of vortex shedding over a cylinder. The spatial discretisation was a finite volume CFD code using Spalart-Allmaras (a closure model designed to work well for aerospace applications) Reynolds Averaged Navier Stokes (RANS) turbulence modelling. Speedups of 7.53x were found using 4096 processors, reducing simulation run time from 655 minutes to 87 minutes. Eghbal et al. (2017) coupled a RANS Shear Stress Transport (SST) coarse solver with a detached eddy simulation (DES) fine solver, in a novel non-traditional Parareal simulation. Instead of directly using Parareal until convergence, Parareal was employed to reduce the time to steady-state by generating initial conditions using the coarse state, and attempting to reduce the transient computation time using Parareal corrections. After a number of Parareal iterations, the fine solver was used in serial to obtain the final statistically steady state. Speedups of up to 2.5x were found using 10 processors with Reynolds numbers up to 50,000. Lunet et al. (2018) studied the Parareal algorithm for homogeneous isotropic turbulence decay in direct numerical simulations (DNS). Performance was

modelled rather than tested on HPC, and attention was focussed on the factors that affected convergence of Parareal. A key factor was found to be the method of interpolation/restriction between coarse and fine spatial grids. High order interpolation methods were found to greatly increase convergence of Parareal over simple linear interpolation.

The shallow water equations are often used as analogues for weather modelling, and have gained much attention in the parallel in time community. Schreiber et al. (2018) investigated the linear oscillatory partial differential equations using the REXI algorithm, and was able to scale to 3586 cores for a problem which saturated for a very low number of cores using only spatial decomposition. Speedups of up to 1503x were found for a finite-difference discretisation, though much reduced performance (118x speedup) was found when spectral discretisation was used. The authors explained that this was due to the much higher accuracy of the spectral discretisation, which required that REXI use more terms in order to fully represent more time frequencies. The method was extended to the rotating sphere in Schreiber and Loft (2019), where a potential wall clock speedup of 13.4x using 512 processors was reported. Using REXI as the solver for the linear part of the full nonlinear shallow water equations was attempted in Schreiber et al. (2019), using Cauchy contour integrals to reduce the number of terms required in the REXI approximations. This was required as in the linear system, REXI can allow for extremely long time steps to be taken with no loss of accuracy; in a nonlinear system this is no longer the case. Hamon et al. (2020) examined the shallow water equations on a rotating sphere using the PFASST algorithm, and found speedups of up to 5.2x using 16 processors in time.

The most relevant applications to our work are mentioned in the following paragraph, though the fluid flow applications mentioned above are also of great interest. Parareal has been studied in simulations of MHD plasma in fusion rectors. In Samaddar et al. (2010) fully developed plasma turbulence simulations were sped up by 8.8x with 88 processors. Reynolds-Barredo et al. (2012) investigated the results and found that high frequency modes, which are not transmitted by the coarse solver, were quickly converged in the fine solver due to their non-linear interactions with the low frequency modes. In Samaddar et al. (2017), edge plasma physics simulations were sped up by Parareal. The Monte-Carlo fine solver was replaced by a fluids neutrals model in the coarse solver, allowing speedups of 22 with 96 time slices. Samuel (2012) used Parareal for convection of the Earth's mantle, using an infinite Prandtl convection model to find parallel in time and space speed-ups of 25 using 64 processors.

2.2 Dynamo

In the following section we will give a brief review of recent numerical dynamo simulations. More in depth reviews can be found at for example Jones (2011), Charbonneau (2020), Rincon (2019) and Tobias (2021). We begin with studies of the geodynamo, then continue by reviewing solar/stellar dynamo studies, before concluding with more fundamental studies of the underlying dynamo processes.

2.2.1 Geodynamo studies

The first example of geomagnetic field reversals was found in Glatzmaier and Roberts (1995b,a). The computing resources of the time were such that this model did not include the inertial terms ($\partial \boldsymbol{u}/\partial t + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}$) of the momentum equation, so that velocity was solved implicitly. The model used spherical harmonics to represent variables in azimuth (longitudinal) and polar (colatitude) directions, whilst Chebyshev polynomials were used for radial expansion. The parameters were chosen such that Pr = 5000, which is more relevant to the mantle than the outer core, whilst a form of hyper-diffusivity was also employed. The number of grid points in radial (N_r), polar/colatitude (N_θ) and azimuthal (N_ϕ) directions were 49, 32, and 64 respectively. Despite the simplifications, low resolution and unrealistic parameter space, this study required over 2 million time steps and 2000 CPU-hours for an in-simulation duration of 40,000 years.

A benchmark numerical geodynamo (Christensen et al., 2001) followed, which includes inertia and does not use hyper-diffusivity. In this study, Pr = 1 which is much more relevant to the Earth's core, but the Ekman number Ek = 10^{-3}, around 6 orders of magnitude from the best case (i.e. highest) estimated value in the Earth of 10^{-9}. The Rayleigh number was Ra = 10^5 (Jones, 2011), while the magnetic Prandtl number Pm = 5. Six codes were used, all using spherical harmonics for polar and azimuth, while half used finite difference for radius, while half used Chebyshev polynomials. Resolutions of up to $N_r, N_\theta, N_\phi = 250, 92, 44$ were employed. Simulations were run for up to 15 units of viscous diffusion time, partly in an effort to reduce required computational runtime. The solution accuracy was found to be very sensitive to radial resolution in the finite difference codes, which highlights the advantages of spectral accuracy. They also found that there was little sensitivity for time step size. Sreenivasan and Jones (2006) studied the importance of the inertial term, and found that as Pr and Pm were lowered below 1, the role of inertia became important, and MAC balance between magnetic, buoyancy and

Coriolis forces was broken, leading to less Earth-like magnetic fields. Simulations here had resolutions of (N_r, $\ell_{\max}$, $m_{\max}$ = 60, 60, 48) for N_r radial points and spherical harmonics of degree ℓ and order m.

Christensen and Aubert (2006) ran simulations down to Ek = 3×10^{-6} using resolutions of ($N_r, \ell_{\max}, m_{\max} = 97, 224, 224$) whilst attempting to derive scaling laws of the non-dimensional parameters. Resolutions of ($N_r, N_\theta, N_\phi = 511, 514, 1538$) and scaling up to 4096 processors were reached using a finite difference code with overlapping Yin-Yang grid by Kageyama et al. (2008); Miyagoshi et al. (2010), who reached Ek = 2.3×10^{-7} on the Earth Simulator. An alternate benchmark problem was devised by Jackson et al. (2014), with boundary conditions designed to be more amenable to local spatial methods such as finite difference/volume/elements, motivated in part by the large error ($\sim 6\%$) these methods displayed when tackling the Christensen et al. (2001) benchmark. Whilst accuracy for local methods was improved, far higher resolutions were required than for spectral methods, and errors of $\sim 1\%$ were still present, while spectral codes converged to within a fraction of 1%. An important study by Matsui et al. (2016) examined the parallel performance of 15 geodynamo codes when tackling the benchmark problems of Christensen et al. (2001) and Jackson et al. (2014). They found most of the codes scaled effectively up to 16,384 cores. However, greater scalability was closely linked to number of directions of parallelization - 2D and 3D parallel codes scaled better than codes that only parallelized in one direction. It was also reported that local methods were less efficient than spectral codes; for the same accuracy and given the same number of cores, they would need longer run times (due to the requirement for higher resolutions). In spherical harmonic codes, those that use finite difference discretisations for the radial direction required 3 times the radial resolution of codes using Chebyshev polynomials. Aubert et al. (2017); Aubert (2019) carried out simulations down to Ek = 10^{-8} using a refined version of hyperdiffusivity, where the hyperdiffusion only kicks in at a cut-off mode number, aiming to allow the smallest *relevant* length scales to be fully resolved. Resolutions of ($N_r, \ell_{\max}, m_{\max} = 2496, 640, 640$) were used. Similar, but non-magnetic, work was carried out by Guervilly et al. (2019), with resolution of ($N_r, \ell_{\max}, m_{\max} = 2016, 351, 319$). Schaeffer et al. (2017) carried out DNS simulations with no hyperdiffusivity at Ek = 10^{-7}, with resolution up to ($N_r, N_\theta, N_\phi = 1280, 1504, 2688$) using up to 8,192 cores and 10 million cpu hours per simulation. Even with this extremely high use of computational resources, the simulation duration was limited to 0.052 magnetic diffusion times for this most extreme case, further highlighting the bottleneck that serial timestepping represents for DNS geodynamo studies.

2.2.2 Astrophysical dynamo studies

Here we present a brief overview of astrophysical dynamo studies, more comprehensive reviews can be found in Brun and Browning (2017) and Nordlund et al. (2009). The first stellar dynamo studies were carried out using similar Boussinesq assumptions used in geodynamo studies (e.g. Gilman, 1975; Gilman and Miller, 1981). However, the large density gradients of the gas giants and stars, whereby density at the centre is far greater than density at the outer radius, means that use of Boussinesq models becomes rather dubious (Jones et al., 2011; Gastine and Wicht, 2012). Fully compressible models, however, are extremely computationally intensive, due to the very small time scales applicable to sound waves (Jones et al., 2011). As convection in planets and stars is usually subsonic (Jones et al., 2011), the anelastic approximation is used, which essentially filters out sound waves but includes the overall density stratification (Brun and Browning, 2017).

Different types of numerical methods (spatial discretisations) are more frequent in the literature for simulations of gas giants and stars than for geo-dynamos. Many simulations have been carried out using high order finite difference discretisation using for example the PENCIL code (e.g. Käpylä et al., 2011, 2012, 2013; Warnecke et al., 2012, 2013), or the Yin-Yang grid (e.g. Kageyama and Sato, 2004; Masada et al., 2013). These studies hope to overcome the difficulties in parallel scaling encountered by spherical harmonic expansions (Masada et al., 2013) by using local grid methods. Studies have been carried out with the EULAG implicit Large-eddy Simulation (ILES) code (e.g. Racine et al., 2011; Beaudoin et al., 2013; Passos and Charbonneau, 2014; Strugarek et al., 2017), which allows for solar-like dynamo behaviour, for example cyclic activity similar to the 11 year cycle, to be observed at relatively low resolutions. A major drawback of this particular approach is that calculating or even estimating important non-dimensional parameters of these studies, such as Ra and Re is difficult, even *a posteriori* (Beaudoin et al., 2013). This is due to dissipation being introduced at the algorithm level, rather than including dissipation terms in the equations. Diffusivities are unknown, and change to keep the solution stable when necessary.

Spectral codes remain popular for convection in main-sequence stars or planets (Brun and Browning, 2017), with MagIC (Gastine and Wicht, 2012), the Leeds Anelastic Spherical Dynamo Code (ALSD) (Jones et al., 2011), Anelastic Spherical Harmonic code (ASH) (Clune et al., 1999; Miesch et al., 2000), and adaptions of the code from Glatzmaier (1984) all used to create a benchmark study (Jones et al., 2011) in a similar fashion to the Christensen et al. (2001) benchmark. When

applied to solar-dynamos, in contrast with geo-dynamo models, these simulations often employ some modelling for the smallest scales. The ASH code in Nelson et al. (2013); Nelson and Miesch (2014); Brun et al. (2017) uses an LES Subgrid Scale (SGS) model, whilst in Augustson et al. (2015) a slope-limited diffusion model was tested. The Rayleigh MHD code has been used to simulate the solar dynamo by making use of turbulent 'eddy' diffusivities, due to the inability to resolve the full range of motion (Matilsky and Toomre, 2020), whilst in Featherstone and Hindman (2016) it was simply noted that diffusivities were much higher than those found in the sun, as in most geodynamo models.

The assorted simulations have been able to produce many features present in the solar and other stellar dynamos, such as large-scale magnetic fields (e.g. Gilman, 1983; Glatzmaier, 1985) cyclic magnetic activity similar to the 11 year cycle (e.g. Racine et al., 2011; Käpylä et al., 2012), including equatorward migration of sunspot activity (e.g. Käpylä et al., 2013; Augustson et al., 2015), and coronal mass ejections (Warnecke et al., 2012). Significant deficiencies remain however, the parameter ranges in these simulations remain far away from values found in the Sun and other stellar dynamo's, $\mathrm{Ra} \sim 10^7 - 10^8$ remain at the cutting edge, whilst is thought to be $\sim 10^{14}$ in the Sun (Käpylä et al., 2013), and $\mathrm{Re} \sim 100$ found in simulations is far away from the highly turbulent flows found in stars (Brun and Browning, 2017). These limitations remain in spite of the computational resources utilised by the simulations, which can take weeks to run (Dietrich and Jones, 2018) and utilise tens of thousands of cores (Featherstone and Hindman, 2016).

The simulations reviewed above hightlight the trade off that has to be made between long durations in simulation time, and high resolution in space and time. The problem is twofold: the CFL condition (or similar constraint), reduces maximum time step size for higher spatial resolution, whilst the higher resolution increases the computational workload per time step. Simulations with very high spatial resolution, such as 'S3' of Nelson et al. (2013); Nelson and Miesch (2014) with $\mathrm{Re} \sim 10^4, \mathrm{Ra} \sim 10^9$, have durations of only a few years, while simulations investigating long term behaviour of solar dynamo such as Augustson et al. (2015), have durations of $\sim$ 100 years, but are limited to $\mathrm{Re} \sim 300, \mathrm{Ra} \sim 10^5$, using around a quarter the number of grid points in each dimension. ILES simulations employ relatively small spatial meshes to permit long time integration (Passos and Charbonneau, 2014), for example Guerrero et al. (2019) simulates for $\sim$ 300 years with a grid of $N_r, N_\theta, N_\phi = 64, 64, 128$, whilst Passos and Charbonneau (2014) simulated 1650 years using a grid of $N_r, N_\theta, N_\phi = 128, 64, 47$.

Dynamo action is also thought to be responsible for the magnetic fields of

the gas giant planets of the solar system (and beyond). In Jupiter (Jones, 2014) and Saturn (Yadav and Bloxham, 2020), a convective layer of high pressure metallic (and thus conductive) hydrogen is thought to surround the cores, with the outer layers of the planets being made up of molecular hydrogen and helium. The ice giants are thought to convect in their ice-rich mantle layers, made up of water, ammonia and methane (Soderlund and Stanley, 2020). The Juno and Cassini missions have sparked interest in the dynamos of Jupiter and Saturn, with studies by, for example Gastine et al. (2014); Heimpel et al. (2016); Duarte et al. (2018); Glatzmaier (2018); Wicht et al. (2019) using a variety of spherical codes. Saturn has also seen some interest, with studies by e.g. Dietrich and Jones (2018); Yadav and Bloxham (2020). Studies on the Ice Giants are less common, though some simulations have been carried out e.g. Stanley and Bloxham (2004, 2006); Soderlund et al. (2013), with a review of the subject written by Soderlund and Stanley (2020). A number of these were high resolution DNS studies, such as Dietrich and Jones (2018) and Yadav and Bloxham (2020), with the latter using over 3 million cpu hours over the course of their simulations.

Our discussion has shown that current numerical methods cannot reach the parameter ranges present in astrophysical objects. Obtaining simulations with parameters closer to those found in planets and stars is an ongoing problem, with many strategies employed in the literature. We believe that Parallel-in-time methods should be explored as an option to help this search, especially where high spatial resolution causes small time step sizes, which limit the in-simulation duration.

2.2.3 Theoretical and Kinematic dynamo studies

In this section, we will review numerical studies that have been primarily focused on investigating the underlying mechanisms of dynamo action. We begin with the first dynamo studies, which aimed simply to find evidence of dynamo action in numerical experiments. It is here that we introduce the Roberts (1972) dynamo which we study in Chapter 4. We then discuss the subject of fast/slow dynamos, which is of particular relevance for astrophysical objects, and is where we introduce the Galloway and Proctor (1992) dynamo, also studied in Chapter 4. Finally we discuss the issues of saturation and large-scale dynamos, two areas of active research.

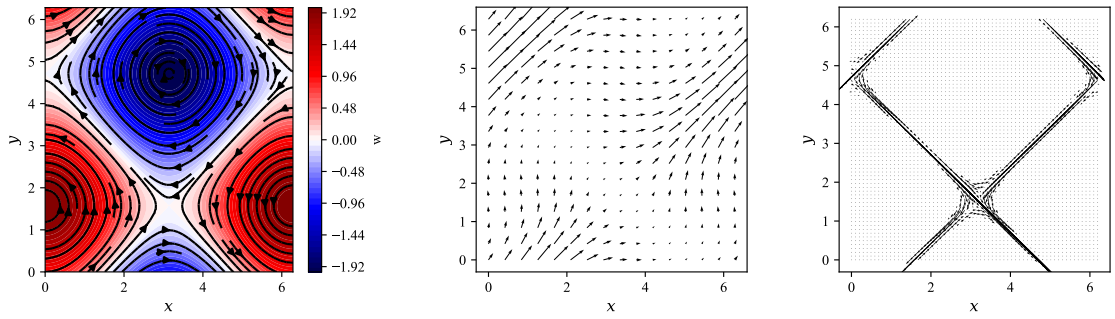
Anti-dynamo theorems, early successful dynamos

A number of anti-dynamo theorems rule out dynamo action for many simplified scenarios. Cowling (1933) states that no axisymmetric magnetic field vanishing at infinity can be maintained by dynamo action. Zeldovich (1957) states that a dynamo in an infinite domain cannot be maintained by a planar flow

$$\boldsymbol{u}(x, y, z, t) = (u(x, y, z, t), v(x, y, z, t), 0)\,. \tag{2.2}$$

It is further impossible to maintain a 2D magnetic field through dynamo action. There are however various caveats to these theorems, and for example, a planar flow in a sphere was found to act as a dynamo in Bachtiar et al. (2006), where η was variable due to boundary constraints. In addition to these constraints on dynamo action, Backus (1958) and Childress (1969) determined that dynamo action cannot happen below a critical value of Rm, known as the critical magnetic Reynolds number Rm_C, due to diffusion dominating over advection in the induction equation (Jones, 2008). The Cowling (1933) theorem in particular is thought to have slowed progress in dynamo studies (Rincon, 2019); it was not until the works of Parker (1955); Herzenberg (1958); Backus (1958) that working dynamos were published.

An important early dynamo was the helical flow by Ponomarenko (1973), which has a discontinuity at the outer boundary. A 2.5D flow which can be solved on a 2D plane while evading the theorem of Zeldovich (1957) was studied by Roberts (1972). This flow $\boldsymbol{u} = (\cos y, \sin x, \sin x + \cos y)$ (see Figure 2.1 for visual representation of the flow and resulting magnetic fields) has 3 components of velocity, so is not planar, but is chosen such that it does not vary in the z direction. The solution can then be calculated on a plane, with solutions of the form $\boldsymbol{B} = \boldsymbol{B}_0(x, y)\mathrm{e}^{\gamma t + ik_z}$, and the z wavenumber k_z treated as a parameter. These early successful numerical dynamos have strongly influenced the design of the first successful physical dynamo experiments at both Riga (Gailitis et al., 2000) based on the dynamo of Ponomarenko (1973) and Karlsruhe (Stieglitz and Müller, 2001) based on the dynamo of Roberts (1972). Tilgner and Busse (1995) studied the Roberts flow dynamos in cases where the spatial period of the magnetic field is different to that of the specified flow field, work that was adapted by Plunian and Rädler (2002) to study the Karlsruhe experiment, and found that Rm calculated purely for the axial component of the flow had to be above a critical value in order for the dynamo to operate.



(a) $\boldsymbol{u}$ (b) Rm = 4 (c) Rm = 4096

Figure 2.1: (a)Roberts flow: streamlines of u, v with contours of w. Positive w (red) is directed out of the page, while negative w is directed into the page. (b,c) vector map of Roberts magnetic field for different Rm. We can see that the length scale of the magnetic field features are much smaller at higher Rm, while magnetic flux is concentrated in the stagnation points.

Fast/slow dynamos - ABC and time dependent 2D

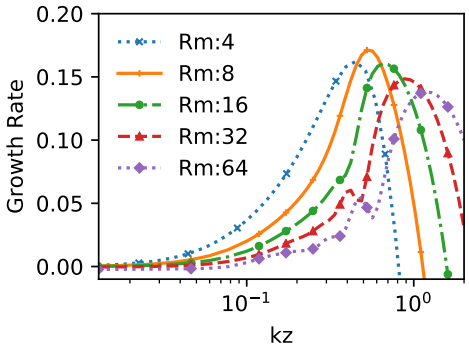
The dynamos previously mentioned are all 'slow' dynamos, operating on diffusive timescales, where the maximum growth rate decays as Rm tends to infinity. Figure 2.2a shows how the maximum growth rate w.r.t k_z reduces as Rm increases in the Roberts flow, whilst Figure 2.2b shows the asymptotic behaviour of the Galloway-Proctor growth rates for high Rm. Whilst slow dynamos may still allow for dynamo action in (for example) the Earth, which has Rm $\sim 10^3$ (Davies et al., 2015), for astrophysical dynamos, such as in the Sun, Rm is expected to be much higher at $\sim 10^6 - 10^9$ (Weiss, 2001). This means that 'fast' dynamos, operating on the timescale of the fluid flow turnover (Galloway and Proctor, 1992), with growth rates becoming independent of Rm as Rm $\to \infty$ are required. Roberts and Soward (1992) and Childress et al. (1990) reviewed the main candidates for fast dynamos, and reported a number of 3D flows that were expected to act as fast dynamos. An important example is the ABC (Arnold-Beltrami-Childress) flow

$$\boldsymbol{u} = (C\sin z + B\cos y, A\sin x + C\cos z, B\sin y + A\cos x)\,, \tag{2.3}$$

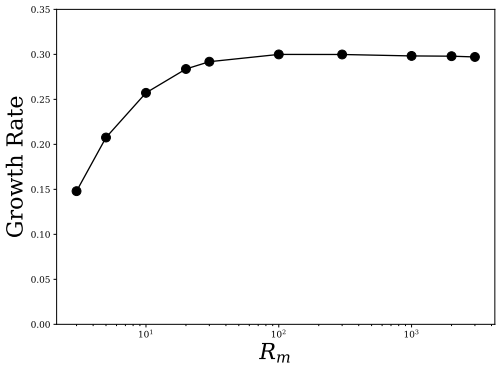
where A, B and C are arbitrary constants, usually small integers. (The Roberts dynamo is a specific case of ABC flow with C set to 0, to give the required 2.5D spatial dependence.) In the early 1990's the computational expense of 3D dynamo calculations restricted investigations to Rm values too low to give compelling evidence for the existence of fast dynamos (Galloway, 2012). Therefore the first convincing numerical fast dynamos were time dependent variations of existing 2.5D dynamos. The time dependence introduces chaos into the fluid flow, which is a necessary con-

dition of fast dynamo action (Hollerbach et al., 1995), and which is already present in fully 3D ABC flows (Jones, 2008). The first two fast dynamos with smooth flows (i.e. did not contain singularities or discontinuities) were published by Galloway and Proctor (1992) and Otani (1993). Galloway and Proctor (1992) added a time dependent 'wobble' to the Roberts (1972) flow, whilst Otani (1993) extended the work of Bayly and Childress (1988) into a time continuous setting by modulating the amplitude of the flow. These were the first fast dynamos to have physically realizable analogues (Otani, 1993). Both of these dynamos showed magnetic growth rates which became independent of Rm at the highest Rm simulated, though it remains impossible to prove numerically that these time dependent flows are fast (Tobias, 2021). Hollerbach et al. (1995) found evidence of fast dynamo action in a spherical shell, with simulations carried out up to $\mathrm{Rm} = 10^5$. The flow field here was time dependent and axisymetric with circulation cells shaken back and forth in radius (r). They used 512 angular modes in spherical harmonics and 1000 finite difference grid points in radius. Reyl et al. (1996) found evidence for fast dynamo action with growth rates collapsing onto a single positive growth rate at Rm upto 10^5 in a quasi-two-dimensional random fluid flow. A fully 3D model was studied by Archontis et al. (2003), with ABC dynamos with different ratios of $A : B : C$. The magnetic field in $A : B : C = 1 : 1 : 1$ dynamos appears as cigar like concentrations of magnetic energy concentrated around the stagnation points in the flow. If A, B and C are chosen such that there are no stagnation points, the cigars are replaced by flux sheets, and the amplification mechanism is the constructive folding of flux 'ribbons'.

Rotation and helicity are thought to be important to dynamo action due to their symmetry breaking properties (Rincon, 2019). Llewellyn Smith and Tobias (2004) investigated a dynamo generated by a helical forcing. The forcing, in a 2.5D flow, causes an inverse cascade creating large coherent vorticity patches. This flow is a good candidate for fast dynamo action, with Rm considered up to ~ 5000. Alexakis (2011) conducted a detailed survey of the dynamo properties of ABC flows with different ratios of A,B and C, finding that Roberts type flows work at the lowest Rm, and that $A = B \approx 2C/5$ has the highest growth rate as Rm is increased. Jones and Gilbert (2014) compared timestepping with directly solving for the eigenvalue in a number of ABC flows, and found that it is sometimes preferable to use timestepping methods even when an eigenvalue solution is possible. This can be considered a somewhat surprising result, but comes from the ability for timestepping codes to be parallelised more easily than eigenvalue solvers. This is relevant to our work, as we will be timestepping the Roberts flow, which can be solved as an eigenvalue problem. Seshasayanan and Alexakis (2016) studied turbulent 2.5D dynamos of



(a) Growth Rates of Roberts dynamo for different values of k_z for increasing Rm



(b) Growth rates of Galloway-Proctor dynamo with increasing Rm. See equation (4.6) for details of flow

Figure 2.2: Comparison of how growth rates change with increasing Rm for slow (Roberts) and fast (Galloway-Proctor) kinematic dynamos.

flows without large scale fluid flow features, and found that helicity led to lower Rm_C, and showed evidence of fast dynamo action. Despite the inability to prove the existence of fast dynamos, that large number of computed dynamos which are apparently fast, means their existence can be regarded as established (Galloway, 2012). Seshasayanan et al. (2017) found that Rm_C can be lower in rotating flows than for comparable non-rotating flows. A 3D kinematic dynamo with random turbulent flow with and without rotation was studied in Dallas and Tobias (2018), and it was found that the growth rate is increased for flows with rotation.

Saturation and Large scale fields

Much attention in the recent past has been concentrated on the saturation mechanism and the generation of large scale fields. These two problems, while they may seem distinct and easy to study on their own, turn out to be quite closely linked. In a talk at Kavli Institute for Theoretical Physics, F. Cattaneo (http://online.kitp.ucsb.edu/online/dynamo_c08/cattaneo) discussed the 'general wisdom' of large scale dynamo theory. This wisdom is that reflectionally symmetric turbulence generates magnetic energy well, but on a small scale, and that reflectionally non-symmetric (helical) turbulence generates magnetic flux well (leading to large scale magnetic field). The talk went on to discuss that there were flaws in this general wisdom, as a number of recent results had failed to adhere to the theory. For example, Livermore et al. (2007a) investigated a number of different flows in a full sphere, and found that the flow that generated the largest-scale field had zero net helicity, while flow with more helicity had smaller large-scale fields. In

Livermore et al. (2010), this work was extended into the nonlinear regime, and the previous small scale fields gave way to large scale fields with a large mean component, suggesting a strong link between saturation/nonlinear dynamo effects and the problem of generating large-scale magnetic fields.

The mechanism by which dynamos saturate is not yet understood. It is known that at a certain level of magnetic energy, the magnetic field affects the fluid flow such that the dynamo effect is supressed, and the magnetic field either stabilises or decays. However, exactly how the magnetic field changes the fluid flow is not known, nor can we predict at what level of magnetic energy this will occur in advance. For example Brummell et al. (1998) found a kinematic dynamo flow which, when saturated, did not maintain the magnetic field at a stable level, but allowed it to decay. The saturated fluid flow was stable to the forcing that generated the initial flow, but was not the original flow, and this new saturated flow did not act as a kinematic dynamo. A further surprising result was shown by Cattaneo and Tobias (2009) who investigated the saturation of magnetic fields by running a fully nonlinear dynamo calculation to saturation, and then using the Lorentz saturated fluid flow to drive the growth of a new seed magnetic field. If dynamo saturation is due to simple statistical properties or magnitude of the flow being changed by the Lorentz force, the new magnetic field should be saturated (not grow) just as the original magnetic field is. However, it is found that for any other 'passive' field (that does not act back on the flow through the Lorentz force), the saturated flow acts as a kinematic dynamo. This implies that magnetic field saturation is a *very* nonlinear process, which relies on the coherence between the driving flow and the magnetic field. Similar work was carried out by Tilgner and Brandenburg (2008), but using a simpler case - that of Roberts flow with infinite magnetic prandtl number leading to no inertial term in the momentum equation. They found similar results- a passive field is grown exponentially by a saturated flow field. The work links into large/small scale dynamo theory, as it is argued that in the nonlinear regime, the ability to observe a small scale field depends on the relative saturations of the large and small scale fields (Cattaneo and Tobias, 2014). Conversely, in the kinematic regime it is proposed that if small scale dynamo action (characterised by strong fluctuations unrelated to the mean field) could be avoided or suppressed, finding a large scale field would be easier (Cattaneo and Hughes, 2006). Studies by Tobias and Cattaneo (2013); Cattaneo and Tobias (2014), found a large scale magnetic field was generated in the kinematic case if a large scale shearing flow were imposed on underlying small scale dynamo flows. Large scale magnetic fields were generated (at Rm up to 2500) along the regions of high shear and high helicity, and it was determined that the large scale magnetic fields came about from the suppression of

the small scale dynamo action. The highly nonlinear nature of dynamo saturation is further illustrated by the existence of sub-critical dynamos in convection (see §3.1.3 for explanation/details on convective flows), where a magnetic field generated by super-critical Rayleigh number (Ra) allows convection and dynamo action to continue when Ra is reduced below its critical value (Stellmach and Hansen, 2004; Cooper et al., 2020). Conversely, strong magnetic fields have been shown to suppress convective motions (e.g. Cattaneo et al., 2003; Stein, 2012).

2.3 Discussion

In this chapter, we have carried out a survey of recent work in the fields of Parallel-in-time and numerical dynamos. This has been done to help inform our choices of specific algorithms and problems for initial studies. We also gave an overview of Parallel-in-time methods, and discussed the challenges faced in the numerical dynamo simulations. We have shown that successful application of Parallel-in-time methods to dynamos would be a useful tool in carrying out numerical simulations in parameter regimes closer to those of the Earth, Sun, stars and planets. On parallel-in-time methods, the non-intrusiveness of Parareal, coupled with the similarities it shares with other parallel-in-time methods, make it a valid choice. For choice of problem, the large resolutions, nonlinear dynamics, and high computational cost of full dynamo simulations would be prohibitive to carrying out large numbers of scaling simulations. The ability to run simulations in 2D is important in this regard, and the Roberts and Galloway-Proctor dynamos are canonical 2D dynamo problems. They also have applications in recent work such as the Karlsruhe dynamo experiment (Stieglitz and Müller, 2001) and in investigating the problems of large scale magnetic field generation (Cattaneo and Tobias, 2014). In Chapter 3, we discuss the mathematical specifics of Parareal, dynamos, Rayleigh-Bénard convection, and the pseudospectral method we use in our simulations.

Chapter 3

Theory and Methods

In this chapter, we will discuss the mathematical details and pre-requisites applicable to this work. We will begin with a brief overview of the dynamo problem. We will derive the equations of magnetohydrodynamics (MHD), starting from the equations of electro-magnetism and fluid dynamics. We will then investigate the mathematical properties of the kinematic dynamo problem and then Rayleigh-Bénard convection. We will then investigate the numerical methods used to obtain simulation results of these systems. We will describe spectral methods based on Fourier series and Chebyshev polynomials, and outline implicit-explicit (IMEX) timestepping. A description of the Dedalus code, which implements these algorithms will follow. This chapter then ends with an explanation and discussion on the properties of the Parallel-in-time algorithm Parareal.

3.1 Fluids and Magnetohydrodynamics

The dynamo problem uses the equations of MHD, which combine the pre-Maxwell equations with the Navier-Stokes equations. We will here show how the two sets of equations can be combined for MHD.

3.1.1 MHD

The differential form of Maxwell's equations in a vacuum are (from e.g. Davidson (2001)):

$$\boldsymbol{\nabla} \cdot \boldsymbol{E} = \frac{\rho_c}{\epsilon_0}, \tag{3.1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0, \tag{3.2}$$

$$\boldsymbol{\nabla} \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}, \tag{3.3}$$

$$\boldsymbol{\nabla} \times \boldsymbol{B} = \mu_0 \boldsymbol{j} + \epsilon_0 \mu_0 \frac{\partial \boldsymbol{E}}{\partial t}, \tag{3.4}$$

and Ohm's law for a moving conductor is

$$\boldsymbol{j} = \sigma \left(\boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B} \right), \tag{3.5}$$

where $\boldsymbol{E}$ is the electric field, ρ_c is the charge density, ϵ_0 the permittivity of free space, μ_0 the magnetic permeability, $\boldsymbol{B}$ the magnetic flux density, $\boldsymbol{j}$ the current density, $\boldsymbol{u}$ the velocity, σ the conductivity of the fluid, and t is time. Equation (3.1) is Gauss's law for an electric field, that divergence of the electric field is proportional to the charge density at a point, (3.2) is Gauss's law for a magnetic field, that divergence of the magnetic flux is always zero, which indicates there are no magnetic monopoles, and all magnetic field lines are closed loops. Equation (3.3) is Faraday's law of induction, that a changing magnetic field produces an electric field, whilst equation (3.4) is the Ampére-Maxwell law that a circulating magnetic field is generated by electric current and by a changing electric field.

To derive the equations of MHD, we first note that for fluid flows much slower than the speed of light $c = 1/(\epsilon_0 \mu_0)^{1/2}$, the last term of equation (3.4) can be neglected by dimensional arguments, and because ρ_c is insignificant everywhere except equation (3.1), Gauss's law is also neglected (Davidson, 2001). We now substitute (3.5) into (3.4), giving

$$\frac{1}{\sigma \mu_0} \left(\boldsymbol{\nabla} \times \boldsymbol{B} \right) = \boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B}, \tag{3.6}$$

which we then take the curl of:

$$\boldsymbol{\nabla} \times \eta \left(\boldsymbol{\nabla} \times \boldsymbol{B} \right) = \boldsymbol{\nabla} \times \boldsymbol{E} + \boldsymbol{\nabla} \times \left(\boldsymbol{u} \times \boldsymbol{B} \right), \tag{3.7}$$

where $\eta = 1/(\mu_0 \sigma)$ is the magnetic diffusivity. Substituting (3.3) gives

$$\boldsymbol{\nabla} \times \eta \left(\boldsymbol{\nabla} \times \boldsymbol{B} \right) = -\frac{\partial \boldsymbol{B}}{\partial t} + \boldsymbol{\nabla} \times \left(\boldsymbol{u} \times \boldsymbol{B} \right), \tag{3.8}$$

which can be rearranged for constant η using the vector identity (for an arbitrary field $\boldsymbol{a}$) $\boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{a}) = \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{a}) - \nabla^2 \boldsymbol{a}$, as

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B}. \tag{3.9}$$

This equation governs how the magnetic field evolves in time, and is dependent upon the velocity of the fluid. We can make this equation more physically meaningful through a little manipulation. If we use the vector identity $\boldsymbol{\nabla} \times (\boldsymbol{a} \times \boldsymbol{b}) = \boldsymbol{b} \cdot \boldsymbol{\nabla} \boldsymbol{a} + (\boldsymbol{\nabla} \cdot \boldsymbol{b})\, \boldsymbol{a} - \boldsymbol{a} \cdot \boldsymbol{\nabla} \boldsymbol{b} - (\boldsymbol{\nabla} \cdot \boldsymbol{a})\, \boldsymbol{b}$ (for an arbitrary vector pair $\boldsymbol{a}$, $\boldsymbol{b}$), and the solenoidal nature of $\boldsymbol{B}$, we can write the induction equation as

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{B} \cdot \boldsymbol{\nabla} \boldsymbol{u} - \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{B} - (\boldsymbol{\nabla} \cdot \boldsymbol{u})\, \boldsymbol{B} + \eta \nabla^2 \boldsymbol{B}. \tag{3.10}$$

This can now be seen as an advection diffusion equation for the magnetic field, with a couple of important additions. $\boldsymbol{B} \cdot \boldsymbol{\nabla} \boldsymbol{u}$ is the key extra term; it represents stretching of the magnetic field lines by the velocity field, it is this effect that allows growth of the magnetic field. $(\boldsymbol{\nabla} \cdot \boldsymbol{u})\, \boldsymbol{B}$ represents the changes of $\boldsymbol{B}$ due to compressibility of the flow, for incompressible flows this disappears.

Having derived the equations governing the evolution of the magnetic field, we now move on to the fluid velocity. Conservation of mass and momentum in fluid flow are expressed by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{u} = 0, \tag{3.11}$$

and the momentum equation:

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} \right) = -\boldsymbol{\nabla} p + \rho \boldsymbol{g} + \boldsymbol{F} + \mu \nabla^2 \boldsymbol{u} + \frac{\mu}{3} \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}), \tag{3.12}$$

where ρ is the density of the fluid, $\boldsymbol{u}$ is the velocity, p is pressure, $\boldsymbol{g}$ is gravity, $\boldsymbol{F}$ represents any other forces (here we will be concentrating on the Lorentz force) and μ is the dynamic viscosity (Pedlosky et al., 1987). The left hand side of the momentum equation is the mass per unit volume multiplied by the fluid acceleration, while the right hand side collects all the forces acting on a unit volume; the momentum equation is Newton's law of motion for a fluid. Where ρ is not constant and the fluid is compressible, we must invoke thermodynamics and find relationships between pressure, density, and temperature. In astrophysical dynamo settings, such as the Sun and stars, effects of compressibility are important and should not be neglected (Jones et al., 2011) as the vertical scale of the flow is very large which causes significant changes in density (Kundu et al., 2016). In the Earth's core, however, it is reasonable to invoke the Boussinesq approximation (Gubbins and Roberts, 1987). In this approximation, density variations are deemed to be small, and are neglected except for their effect on gravitational force. In this work, we are examining the more simple cases first, as our aim is to find information on the performance of numerical methods that have not been applied in this area before. Therefore, we

restrict ourselves to the Boussinesq case here. In the Boussinesq approximation, density variations in the gravity term are assumed to be linear and of the form

$$\rho = \rho_0 \left(1 - \alpha(T - T_0)\right), \tag{3.13}$$

(e.g. Chandrasekhar, 1961; Kono and Roberts, 2002) where T is the temperature, ρ_0 is the reference or background density found at the reference temperature T_0, and α is the coefficient of thermal expansion. Except in the gravity term, ρ is replaced by ρ_0. Ignoring internal sources of heat, and assuming a constant thermal diffusivity κ, temperature is governed by the heat equation

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T \tag{3.14}$$

(Kundu et al., 2016). The Boussinesq approximation reduces equation (3.11) to the form

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{3.15}$$

Using equations (3.13) and (3.15), dividing through by ρ_0, and absorbing the static gravity potential into the pressure term, the momentum equation can be re-written as

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) = -\frac{1}{\rho_0} \boldsymbol{\nabla} p - \alpha(T - T_0)\boldsymbol{g} + \frac{1}{\rho_0} \boldsymbol{F} + \nu \nabla^2 \boldsymbol{u}, \tag{3.16}$$

where $\boldsymbol{g}$ is the force due to gravity and $\nu = \mu/\rho_0$ is the kinematic viscosity.

The remaining body force is due to the magnetic field, and we will now derive its form. The force on a particle due to a magnetic field is given by

$$\boldsymbol{F}_L = q_c \left(\boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B}\right), \tag{3.17}$$

where q_c is the charge of the particle. For a continuous charge distribution, this becomes

$$\boldsymbol{F}_L = \rho_c \boldsymbol{E} + \boldsymbol{j} \times \boldsymbol{B}. \tag{3.18}$$

For fluid flow speeds much less than the speed of light, the first term is negligible, and can be ignored (Davidson, 2001), so that the Lorentz force can be written as

$$\boldsymbol{F}_L = \boldsymbol{j} \times \boldsymbol{B} = \frac{1}{\mu_0} (\boldsymbol{\nabla} \times \boldsymbol{B}) \times \boldsymbol{B}. \tag{3.19}$$

For clarity, we now write down the full set of equations for incompressible,

non-rotating MHD which have been derived in agreement with (Davidson, 2001):

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) = -\frac{1}{\rho_0} \boldsymbol{\nabla} p - \alpha (T - T_0) \boldsymbol{g} + \frac{1}{\mu_0 \rho_0} (\boldsymbol{\nabla} \times \boldsymbol{B}) \times \boldsymbol{B} + \nu \nabla^2 \boldsymbol{u}, \tag{3.20}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T, \tag{3.21}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B}, \tag{3.22}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{3.23}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0. \tag{3.24}$$

The above equations define the incompressible MHD problem once suitable boundary conditions are defined for T, $\boldsymbol{u}$, and $\boldsymbol{B}$. We now move on to two simplifications of this problem, the kinematic dynamo followed by Rayleigh-Bénard convection.

3.1.2 Kinematic Dynamo

The kinematic dynamo problem seeks pre-determined velocity fields ($\boldsymbol{u}$) which lead to a growing magnetic field ($\boldsymbol{B}$) in the induction equation

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B}, \tag{3.25}$$

where the magnetic field is solenoidal:

$$\boldsymbol{\nabla} \cdot \boldsymbol{B} = 0. \tag{3.26}$$

This is a subset of the non-linear dynamo problem described by equations (3.20) - (3.24), where generation of the fluid flow is ignored. The kinematic dynamo problem is studied as a more tractable analogue of the full dynamo problem, with less computational requirements, and less physical effects to take account of. The draw backs of this method include neglecting the effect of the magnetic field on the fluid flow, and ignoring saturation of the magnetic field growth. Solutions are restricted to either exponential growth or decline of the magnetic field strength. (see §1.4 for a full discussion on the applicability of the kinematic dynamo approach). In this section, we will first nondimensionalize the induction equation. We will then discuss a number of important restrictions on dynamo action, known as anti-dynamo theorems. We will then look at strategies to overcome those theorems, and discuss the type of results that can be found.

We first show the non-dimensionalisation of the induction equation, which will be used throughout this work. We normalise the magnetic field $\boldsymbol{B}$ with a characteristic field strength B, distance $\boldsymbol{X}$ with a characteristic length L, speed with the characteristic flow speed U, and time with L/U. Non-dimensional quantities are identified by a superscript *, so that for example dimensionless magnetic field $\boldsymbol{B}^* = \boldsymbol{B}/B$. Substituting into (3.25) gives

$$\frac{BU}{L}\frac{\partial \boldsymbol{B}^*}{\partial t^*} = \frac{UB}{L}\nabla^* \times (\boldsymbol{u}^* \times \boldsymbol{B}^*) + \frac{\eta B}{L^2}\nabla^{*2}\boldsymbol{B}^*. \tag{3.27}$$

This can be simplified to

$$\frac{\partial \boldsymbol{B}^*}{\partial t^*} = \nabla^* \times (\boldsymbol{u}^* \times \boldsymbol{B}^*) + \frac{\eta}{LU}\left(\nabla^*\right)^2 \boldsymbol{B}^*. \tag{3.28}$$

The magnetic Reynolds number, which determines whether advection or diffusion of the magnetic field dominates, is defined as

$$\mathrm{Rm} = \frac{UL}{\eta}, \tag{3.29}$$

which gives the non-dimensional induction equation

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) + \frac{1}{\mathrm{Rm}}\nabla^2 \boldsymbol{B}, \tag{3.30}$$

once we drop the superscript *.

There are a number of anti-dynamo theorems, which rule out dynamo action for common simplified models. We will briefly outline their consequences here, for a more in depth discussion, see for example Gilbert (2003). The most famous anti-dynamo theorem is that of Cowling (1933). This theorem states that an axisymmetric magnetic field vanishing at infinity cannot be maintained by dynamo action. The Zeldovich (1957) theorem is the analogue of the Cowling theorem applied to cartesian coordinates. It states that a dynamo cannot be maintained by a planar flow $\boldsymbol{u}(x, y, z, t) = (u(x, y, z, t), v(x, y, z, t), 0)$. A further anti-dynamo theorem states that in Cartesian coordinates, it is impossible to maintain a 2D magnetic field through dynamo action. We finally note that for dynamo action to occur, the magnetic Reynolds number Rm must be greater than some critical value, dependent upon the geometry of the problem, for dynamo action to occur (Backus, 1958; Childress, 1969).

The anti-dynamo theorems of Zeldovich (1957) etc. rule out dynamo action in certain simplified domains, such as purely 2D or purely axisymmetric fluid flow.

It is advantageous in numerical and computational studies to study problems on a 2D domain, as this allows greater resolutions and more simulations to be carried out on the same computer hardware. To allow dynamo action to be investigated in a 2D domain, a flow which depends upon only two of three spatial dimensions can be used. This is commonly called the 2.5D approximation. Here, the flow $\boldsymbol{u}(x, y, t)$ leads to magnetic field solutions of the form $\boldsymbol{B}(x, y, z, t) = \boldsymbol{B}(x, y, t)\mathrm{e}^{ik_z z}$, for which the k_z wave number can be treated as a parameter, which allows the problem to be solved in a 2D plane (e.g. Jones, 2008; Rincon, 2019; Tobias, 2021).

As the induction equation is linear in $\boldsymbol{B}$, if $\boldsymbol{u}$ is constant in time, then solutions of the form

$$\boldsymbol{B} = \boldsymbol{B}_0(x, y, z)\mathrm{e}^{\gamma t}, \tag{3.31}$$

can be obtained, where $\boldsymbol{B}_0$ is the initial state and γ is a growth rate. When combined with the 2.5D approximation, we can look for solutions of the form $\boldsymbol{B}(x, y, z, t) = \boldsymbol{B}_0(x, y)\mathrm{e}^{\gamma t + ik_z z}$. The full solution is a sum over all possible eigenmodes, and the largest eigenmode will dominate as $t \to \infty$. If a periodic time dependence is added to the flow, then the problem can be treated in a similar manner, however the growth rate is no longer constant, but a Floquet exponent (Jones, 2008).

3.1.3 Rayleigh-Bénard Convection

As discussed in chapter 1, it is believed that the solar and planetary dynamos are convection-driven (e.g. Busse, 2002; Jones, 2011), so that investigation into the flows responsible for dynamo action should concentrate on convection. Rayleigh-Bénard convection is often used as a simplified analogue for astro- and geo- physical systems such as planet and stars (e.g. Busse, 1978; Getling, 1998), and we take that approach here.

Rayleigh-Bénard convection describes the motion of a fluid held between two fixed horizontal plates separated by a distance h, heated from below and cooled from above (Ahlers et al., 2009). We model RBC in a 2D Cartesian domain with periodic horizontal boundaries and length L_x. Ignoring magnetic effects (thus setting the Lorentz force to zero), equations (3.20) - (3.24) are reduced to the Oberbeck-Boussinesq equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u}) = -\frac{1}{\rho_0}\boldsymbol{\nabla} p - \alpha(T - T_0)\boldsymbol{g} + \nu\nabla^2\boldsymbol{u}, \tag{3.32}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{3.33}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \kappa \nabla^2 T, \tag{3.34}$$

(e.g. Tritton, 1977; Kundu et al., 2016) where $\boldsymbol{u}$ is the velocity, ρ_0 is the reference density at reference temperature T_0, p is pressure, α is the coefficient of thermal expansion, $\boldsymbol{g} = -g\hat{\boldsymbol{z}}$ is gravity directed in the negative vertical direction, T is the temperature, ΔT is the temperature difference between the top and bottom plate, ν is the kinematic viscosity, and κ is the thermal diffusivity. Fixed temperature, no slip boundary conditions on the top and bottom plates are given by

$$T|_{z=0} = T_{\text{hot}} = T_{\text{cold}} + \Delta T, \quad T|_{z=h} = T_{\text{cold}}, \tag{3.35}$$

$$\boldsymbol{u}|_{z=0} = \boldsymbol{u}|_{z=h} = 0. \tag{3.36}$$

We describe below the non-dimensional scaling for RBC, we use a different scaling than for the kinematic dynamo problem, as temperature and velocity are the variables rather than the magnetic flux density. To find the non-dimensionalised RBC equations, we set $T_{\text{cold}} = T_0 = 0$, $T_{\text{hot}} = T_{\text{cold}} + \Delta T$, and normalise the temperature with ΔT. We normalise distance $\boldsymbol{X}$ with h, t with the diffusion time scale $\tau = h^2/\kappa$, and $\boldsymbol{u}$ with $U = h/\tau$. The Rayleigh number determines how much thermal driving is present in the system. We define the Rayleigh number as

$$\text{Ra} = \frac{\alpha g h^3 \Delta T}{\kappa \nu}, \tag{3.37}$$

(e.g. Chandrasekhar, 1961; Grossmann and Lohse, 2000) and the Prandtl number as

$$\text{Pr} = \frac{\nu}{\kappa}, \tag{3.38}$$

allowing us to write the non-dimensional Oberback-Boussinesq equations:

$$\frac{1}{\text{Pr}}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u}\right) = -\boldsymbol{\nabla} p + \text{Ra} T \cdot \hat{\boldsymbol{z}} + \nabla^2 \boldsymbol{u}, \tag{3.39}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{3.40}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} T = \nabla^2 T, \tag{3.41}$$

with vertical boundaries defined as

$$T|_{z=-0.5} = 1, \quad T|_{z=0.5} = 0, \tag{3.42}$$

$$\boldsymbol{u}|_{z=-0.5} = \boldsymbol{u}|_{z=0.5} = 0, \tag{3.43}$$

and periodic horizontal boundaries.

Critical Rayleigh Number

The fluid layer heated from below and cooled from above found in Rayleigh-Bénard convection studies is stationary when the temperature difference is small. Heat is transported purely by conduction through the fluid. When the temperature difference is large enough, convective instability occurs and the fluid begins to move, enhancing heat transport by advection of the fluid. What is meant by 'small' and 'large enough' temperature differences is defined by the critical Rayleigh number (Ra_c). For a given set of boundary conditions, Ra_c is a fixed constant, that (in a number of simple cases) can be derived from linearisation of the governing equations (e.g. Chandrasekhar, 1961). For fixed temperature, and rigid no slip boundary conditions in a Cartesian domain (as will be studied in this work), $Ra_c \approx 1707.762$ (Chandrasekhar, 1961). As Ra is increased above Ra_c, fluid motions increase in magnitude and eventually become turbulent, and heat transfer is mostly achieved by convection.

Nusselt Number

The Nusselt number is the total heat flux through a surface divided by purely conductive heat flux of a system (Chandrasekhar, 1961). The heat flux of a purely conductive system can be written as

$$q_{\text{cond}} = \lambda \Delta T / h, \tag{3.44}$$

where λ is the thermal conductivity. To determine the relative importance of convective and conductive heat transfer, we normalise the total heat flux q by the conductive heat flux $\lambda \Delta T / h$, which gives the Nusselt number

$$\text{Nu} = q / \left(\lambda \Delta T / h\right) = q h / \lambda \Delta T. \tag{3.45}$$

An expression for the Nusselt number can derived from the dimensionless temperature equation (equation (3.41)). For a Boussinesq fluid the term $\boldsymbol{u} \cdot \boldsymbol{\nabla} T$ can be written as $\boldsymbol{\nabla} \cdot (\boldsymbol{u} T)$, since $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$. The convective and diffusive terms can then be combined as $\boldsymbol{\nabla} \cdot (\boldsymbol{u} T - \boldsymbol{\nabla} T)$. If a time average over a suitable time period is taken, then integration over the volume generates the volume-averaged Nusselt number

$$\text{Nu}_V = \frac{1}{V} \int \left(-\frac{\partial T}{\partial z} + wT \right) \mathrm{d}V, \tag{3.46}$$

where w is the vertical component of velocity, the subscript V denotes calculation over the whole domain, $\partial T/\partial z$ represents conduction, and wT represents convection.

In numerical simulations, this can be approximated as

$$\mathrm{Nu}_V = \left\langle wT - \frac{\partial T}{\partial z} \right\rangle \tag{3.47}$$

where $\langle \ldots \rangle$ indicates a volume average (King et al., 2012). The Nusselt number is 1 for any conductive state, as can be seen by setting $w = 0$ and observing that the total non-dimensional height is 1 and the temperature difference is set to -1. As convection begins to occur, the Nusselt number will begin to rise. For more detailed derivations of the Nusselt number, and those using different dimensionless scaling, see, for example, Moore et al. (1975); Siggia (1994); Grossmann and Lohse (2000); Kerr and Herring (2000); Rieutord (2014). The heat flux is independent of z (Siggia, 1994) due to conservation of energy, which requires

$$\mathrm{Nu} = \overline{\mathrm{Nu}_V} = \overline{\mathrm{Nu}_b} = \overline{\mathrm{Nu}_t}, \tag{3.48}$$

where

$$\mathrm{Nu}_b = \left\langle -\frac{\partial T}{\partial z} \right\rangle_H \Bigg|_{z=b}, \quad \mathrm{Nu}_t = \left\langle -\frac{\partial T}{\partial z} \right\rangle_H \Bigg|_{z=t}, \tag{3.49}$$

are the Nusselt numbers at the bottom and top boundaries, respectively, with $\langle \cdots \rangle_H$ denoting a horizontal average and overbar $\overline{\cdots}$ denoting a time average. To test the convergence of numerical convection studies, adherence to eq (3.48) is checked as

$$\varepsilon_{\mathrm{Nu}} = \frac{\max\left(|\overline{\mathrm{Nu}_b} - \overline{\mathrm{Nu}_V}|, |\overline{\mathrm{Nu}_b} - \overline{\mathrm{Nu}_t}|, |\overline{\mathrm{Nu}_V} - \overline{\mathrm{Nu}_t}|\right)}{\overline{\mathrm{Nu}_V}}. \tag{3.50}$$

The standard test in the literature is for $\varepsilon_{\mathrm{Nu}}$ to be lower than 1% (e.g. Stevens et al., 2010; King et al., 2012; Mound and Davies, 2017).

3.2 Numerical Methods

As was discussed in §2.2, for Geodynamo studies there are clear accuracy benefits for spectral discretisations when compared with competing methods such as finite volume/element/difference. For the Christensen et al. (2001) benchmark, local methods showed errors of $\sim 6\%$, while spectral methods were converged to under 1% at comparable resolutions. In the Jackson et al. (2014) benchmark, which was specifically designed to allow better accuracy for local methods, they were still sig-

nificantly less accurate than spectral codes. In gas giants and solar/stellar dynamos, spectral methods, whilst remaining popular (Brun and Browning, 2017), have a less clear cut advantage, and many studies are carried out using different methods such as finite difference (e.g. Käpylä et al., 2011; Kageyama and Sato, 2004) and Implicit Local Eddy Simulations (e.g. Passos and Charbonneau, 2014; Guerrero et al., 2019). Since spectral methods dominate the geodynamo literature, and are given significant interest in solar/stellar dynamos, they will be used in this work.

3.2.1 Spatial Discretisation - Pseudospectral Method

Numerical computations require that a continuous domain x is discretised to a set of points x_n, and data on that domain $\phi(x)$ is approximately represented by the value of the data at x_n by $\phi_n = \phi(x_n)$. For spectral methods using Fourier series as a basis function, the points x_n are equidistant and scaled such that $x_n \in [0, 2\pi]$. The vector of values $(\phi_1, \ldots, \phi_N)^T$ can be transformed into its Fourier series coefficients $(\widehat{\phi}_1, \ldots, \widehat{\phi}_N)^T$ using the discrete Fourier transform (DFT)

$$\widehat{\phi}_k = \frac{2\pi}{N} \sum_{n=1}^{N} \mathrm{e}^{-\mathrm{i}kx_n} \phi_n, \quad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}, \tag{3.51}$$

where k is known as the wave number. In this form, the data is said to be in spectral space. The data can be transformed back into its original form with the inverse discrete Fourier transform

$$\phi_n = \frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} \mathrm{e}^{\mathrm{i}kx_n} \widehat{\phi}_k. \tag{3.52}$$

Certain operations are trivial to carry out when data is represented in spectral space. Differentiation can be carried out with respect to x for each $\widehat{\phi}_k$ as

$$\frac{\partial \widehat{\phi}_k}{\partial x} = \frac{\partial}{\partial x} \frac{2\pi}{N} \sum_{n=1}^{N} \mathrm{e}^{-\mathrm{i}kx_n} \phi_n = -\mathrm{i}k \frac{2\pi}{N} \sum_{n=1}^{N} \mathrm{e}^{-\mathrm{i}kx_n} \phi_n = -\mathrm{i}k \widehat{\phi}_k. \tag{3.53}$$

Higher order derivatives $\frac{\partial^n}{\partial x^n}$ simply require multiplication by $(-\mathrm{i}k)^n$.

The standard discrete Fourier transform for a vector of length N requires $\mathcal{O}(N^2)$ operations (Cooley and Tukey, 1965), which is comparable with standard matrix multiplications. One of the major reasons for the widespread adoption of spectral methods utilising Fourier series (and Chebyshev polynomial) basis functions, is the Fast Fourier Transform by Cooley and Tukey (1965), which reduces the

computation to $\mathcal{O}(N \log N)$ operations. This allows for differentiation operations to be carried out in spectral accuracy at better than the standard $\mathcal{O}(N^2)$ operation count.

To multiply data in spectral space, a convolution is required. This is an expensive operation to compute, and so where possible, multiplications are carried out in grid space. This means that FFT's are required in every time step, to transform data back and forth between grid/spectral space. This limits the ability of spectral methods to be used in fully implicit time stepping schemes, which will be discussed in more detail in §3.2.2.

Chebyshev Polynomials

What follows is a brief description of the use of Chebyshev polynomials and the Fast Fourier Transfrom to numerically solve partial differential equations, for a more in depth discussion, see for example Trefethen (2000, Ch. 8) or Boyd (2001, Ch. 9,12). Chebyshev polynomials of the first kind are found from the trigonometric identities

$$T_n(\cos(\theta)) = \cos(n\theta). \tag{3.54}$$

A change of variable $x = \cos(\theta)$ is used to convert these trigonometric relations into a set of orthogonal polynomials $T_n(x)$. These Chebyshev polynomials can then be found from the recurrence relation

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned} \tag{3.55}$$

When the change of variables is applied, an equidistant discretisation of $\theta \in [0, \pi]$ into N points becomes a boundary clustered grid in x, defined by

$$x_j = \cos\left(\frac{j\pi}{N}\right). \tag{3.56}$$

When data is sampled at these Chebyshev points, rather than at equidistant points in x, spectral differentiation can be performed. This can be achieved by matrix multiplication with a dense differentiation matrix, which offers spectral accuracy but no speed benefit over standard methods, or by using the Fast Fourier Transform, which allows similar performance to that found in Fourier based spectral methods.

Spectral differentiation using Chebyshev polynomials and the FFT can be carried out as below, adapted from (Trefethen, 2000, Ch. 8, pp. 78).

With data $\phi_0, \ldots, \phi_N$ at Chebyshev points $x_0 = 1, \ldots, x_N = -1$, create the extended vector V of length $2N$ by concatenating the vectors $\phi_0, \ldots, \phi_N$ and $\phi_{N-1}, \ldots, \phi_1$. The FFT is used to find

$$\widehat{V}_k = \frac{\pi}{N} \sum_{n=1}^{2N} \mathrm{e}^{-\mathrm{i}k\theta_n} V_n. \tag{3.57}$$

Define $\widehat{W}_k = \mathrm{i}k\widehat{V}_k$, except that $\widehat{W}_N = 0$. The derivative with respect to θ can now be calculated on the physical grid by using the inverse FFT.

$$W_n = \frac{1}{2\pi} \sum_{k=-N+1}^{N} \mathrm{e}^{\mathrm{i}k\theta_n} \widehat{W}_k, \quad n = 1, \ldots, 2N. \tag{3.58}$$

The derivative of the original data with respect to x can now be found from the chain rule by multiplying by $-\frac{1}{\sin(\theta_n)} = -\frac{1}{\sqrt{1-x_n^2}}$ as

$$\phi_n' = -\frac{W_n}{\sqrt{1-x_n^2}}. \tag{3.59}$$

At the end points, this is undefined, due to a divide by zero, so l'Hopital's rule is implemented as

$$\phi_0' = \frac{1}{2\pi} \sum_{n=0}^{N} n^2 \widehat{V}_n, \quad \phi_N' = \frac{1}{2\pi} \sum_{n=0}^{N} (-1)^{n+1} n^2 \widehat{V}_n. \tag{3.60}$$

As with Fourier spectral bases, multiplicative terms are best treated whilst in grid space, with differentiation operations treated in spectral space.

Boundary Conditions

The major advantage of Chebyshev polynomials over other orthogonal polynomial series, is that because of their origin as a cosine series with a change of variable, they can be quickly computed by the FFT. The advantage of using Chebyshev polynomials over Fourier series, is that arbitrary boundary conditions can be set, using basis recombination, global coefficient adjustment, or modifying the value at the boundary using either a low order finite difference approximation or the highest chebyshev polynomial coefficient.

3.2.2 Time-Stepping

The subject of parallelizing a simulation in the time domain requires a thorough understanding of time-stepping methods in general. In this section, we will give a brief overview, and then discuss the problems of stability and stiffness. We will then introduce Implicit-Explicit (IMEX) timestepping methods, concentrating on IMEX Runge-Kutta methods, which are used in this work. Time stepping methods are used to find an approximate solution to a problem of the form

$$\frac{\partial \phi}{\partial t} = f(t, \phi), \tag{3.61}$$

$$\phi(t = 0) = \phi_0 \tag{3.62}$$

for a variable ϕ, an arbitrary function f at any time t with initial condition given by ϕ_0. As we are using spectral spatial discretisations, ϕ contains the spectral coefficients of the state variables. In the kinematic dynamo, this is $\widehat{\boldsymbol{B}}$, (the spectral coefficients of the magnetic field) whilst in Rayleigh-Bénard convection, it will contain $\widehat{\boldsymbol{u}}$ and $\widehat{T}$ (the spectral coefficients of the velocity and temperature). f is the right hand side of each discretised system of PDE's, containing the advection, diffusion and source terms etc.

Timestepping methods can be constructed by truncating Taylor expansions of the time derivative about a set time point. The Euler forward method truncates the Taylor series about the point $t + \delta t$ where δt is a small step in time

$$\phi(t + \delta t) = \phi(t) + \delta t \frac{\partial \phi(t)}{\partial t} + \frac{(\delta t)^2}{2} \frac{\partial^2 \phi(t)}{\partial t^2} + ... + \mathcal{O}((\delta t)^n) \tag{3.63}$$

to give

$$\phi(t + \delta t) = \phi(t) + \delta t \frac{\partial \phi(t)}{\partial t} + \mathcal{O}((\delta t)^2), \tag{3.64}$$

where $\mathcal{O}(a)$ indicates a term of size $\sim a$. Substituting in the original PDE (3.61) gives

$$\phi(t + \delta t) \approx \phi(t) + \delta t f(\phi(t)), \tag{3.65}$$

(Chapra and Canale, 2011). Euler's method is defined by taking the above approximation to be exact (Atkinson et al., 2011), so with timesteps defined by $t^{n+1} = t^n + \delta t$, and simplifying notation such that $\phi^n = \phi(t^n)$, Euler's method is

$$\phi^{n+1} = \phi^n + \delta t f(\phi^n). \tag{3.66}$$

The Euler backward method can be found by similar methods, and is

written as

$$\phi^{n+1} = \phi^n + \delta t f(\phi^{n+1}). \tag{3.67}$$

Euler forward and backwards time stepping methods have local truncation errors of $\mathcal{O}(\delta t^2)$, and overall truncation errors of $\mathcal{O}(\delta t)$ (Atkinson et al., 2011). To reduce the error of the solution, methods can be constructed using more terms from the Taylor expansion for both explicit and implicit techniques. In effect, these methods use slope information from inside the interval $t + \delta t$ (eg Runge-Kutta or multi-stage methods) or from previous time steps (eg. Multi-step methods), to give a better estimation of the solution. Well known timestepping methods, of 2nd order (global error is $\mathcal{O}(\delta t^2)$) include the implicit Crank-Nicolson scheme

$$\phi^{n+1} = \phi^n + \frac{\delta t}{2}\left(f(\phi^{n+1}) + f(\phi^n)\right), \tag{3.68}$$

the explicit multi-step Adams-Bashforth method

$$\phi^{n+1} = \phi^n + \delta t\left(\frac{3}{2}f(\phi^n) - \frac{1}{2}f(\phi^{n-1})\right), \tag{3.69}$$

and explicit 2nd order Runge-Kutta midopint method

$$K_1 = f(\phi^n), \tag{3.70}$$

$$K_2 = f(\phi^n + \delta t K_1), \tag{3.71}$$

$$\phi^{n+1} = \phi^n + \frac{\delta t}{2}(K_1 + K_2), \tag{3.72}$$

where K_1 and K_2 are intermediate stages computed within each timestep. Note in eq (3.69) the ϕ^{n-1} term, meaning the method requires information to be stored from previous timesteps. A consequence of this is that multi-step methods are unable to start from initial conditions on their own, a number of steps of a different method are required to generate the required data. The Runge-Kutta method described above is one of many variations of two stage Runge-Kutta formulas. The general

explicit Runge-Kutta method is

$$\begin{aligned}
K_1 &= f(t^n, \phi^n), \\
K_2 &= f(t^n + c_2\delta t, \phi^n + \delta t(a_{21}K_1)), \\
K_3 &= f(t^n + c_3\delta t, \phi^n + \delta t(a_{31}K_1) + \delta t(a_{32}K_2)), \\
&\vdots \\
K_i &= f\left(t^n + c_i\delta t, \phi^n + \delta t \sum_{j=1}^{i-1} a_{ij}K_j\right), \\
\phi^{n+1} &= \phi^n + \sum_{i=1}^{s} b_i K_i.
\end{aligned} \tag{3.73}$$

The coefficients a, b, and c are compactly written in notation known as a Butcher Tableau

$$\begin{array}{c|cccc}
c_1 & a_{1,1} & & & \\
c_2 & a_{2,1} & a_{2,2} & & \\
\vdots & \vdots & \ddots & \ddots & \\
c_s & a_{s,1} & \dots & a_{s,s-1} & a_{s,s} \\
\hline
 & b_1 & \dots & b_{s-1} & b_s
\end{array} \tag{3.74}$$

(e.g. Butcher, 2008), which can be used to define any Runge-Kutta method with s stages. If diagonal entries, and those on upper right of the diagonal are zero, the method is explicit. If the diagonal entries are non-zero, then the tableau describes a diagonally-implicit Runge-Kutta method (DIRK), while a tableau containing non-zero upper right entries describes a fully implicit Runge-Kutta method. In this form, the RK2 method above is written as

$$\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & 1/2 & 1/2
\end{array} \tag{3.75}$$

as can be seen below

$$\begin{aligned}
K_1 &= f(t^n + 0\delta t, \phi^n + 0), \\
K_2 &= f(t^n + 1 \times \delta t, \phi^n + \delta t(1 \times K_1)), \\
\phi^{n+1} &= \phi^n + \frac{1}{2}K_1 + \frac{1}{2}K_2.
\end{aligned} \tag{3.76}$$

Derivations and further examples of timestepping methods can be found, for example, in Butcher (2008), Chapra and Canale (2011), and Atkinson et al. (2011).

Stability and Stiffness

Time stepping methods can sometimes generate errors that grow exponentially with time, which indicates that the timestepping method is unstable. Stability of a timestepping method can be analysed by use of a scalar test problem

$$\frac{\partial \upsilon}{\partial t} = \lambda \upsilon, \tag{3.77}$$

$$\upsilon(0) = 1, \tag{3.78}$$

where υ is a scalar variable, and λ is a complex constant (Atkinson et al., 2011). The solution to this problem at time t is

$$\upsilon(t) = \mathrm{e}^{\lambda t}. \tag{3.79}$$

For the solution to be stable, $\Re(\lambda) < 0$, so that small changes in initial conditions do not cause exponentially large changes in the solution. This leads to

$$\upsilon(t) \to 0 \quad \text{as} \quad t \to \infty. \tag{3.80}$$

A timestepping algorithm should maintain this property so that the computed solution υ_n at timestep t_n satisfies

$$\upsilon_n(t_n) \to 0 \quad \text{as} \quad t_n \to \infty. \tag{3.81}$$

A timestepping method is absolutely stable, or A-stable if the above condition holds for any δt. Explicit time stepping methods cannot be A-stable, so when the time step exceeds a limit $\delta t_{\max}$, that depends on the time stepping algorithm, the physics underlying the problem, and the spatial resolution, the error grows exponentially with time (Boyd, 2001). This error growth is called the CFL instability, named after the authors of Courant et al. (1928). All explicit methods have finite $\delta t_{\max}$, while many implicit methods are stable (but not necessarily accurate) for any timestep size (Boyd, 2001). The exact value of the maximum time step size must be calculated for each combination of time and space discretisations and each specific problem, but some 'rules of thumb' can generally be used. For advective terms, the CFL condition is

$$\delta t_{\max} \leq \frac{\delta x}{|\boldsymbol{u}|}, \tag{3.82}$$

where $|\boldsymbol{u}|$ is the local magnitude of the fluid velocity and δx is the smallest distance between two adjacent grid points. The physical explanation of this condition is that the time step size should be smaller than the time taken for fluid to flow from one

grid point to another. For diffusion terms the condition is of the form

$$\delta t_{\max} \leq C(\delta x)^2 \tag{3.83}$$

where C is a constant linked to the coefficient of diffusion, ν (as an example, for a forward-Euler + 1^{st} order finite difference scheme, $C = 1/2\nu$). The dependence of the time step size on $(\delta x)^2$ rather than on δx is an example of numerical stiffness. Increases in spatial resolution cause severe reductions in time step sizes, forcing step sizes much smaller than necessary to obtain reasonable accuracy (Grooms and Julien, 2011). This makes high resolution simulations extremely difficult to carry out.

Implicit Explicit Time stepping (IMEX)

The problem of stiff equations can be alleviated by using implicit time stepping methods, as many implicit methods are A-stable (though some higher order mulit-step methods, for example, are not), meaning that large timestep sizes can be taken. However, non-linear problems are generally very expensive to solve implicitly, requiring iterative techniques to invert large systems (Grooms and Julien, 2011). In many problems, the right hand side can be split into different types of terms, and different timestepping procedures can be used on different terms. A key example of this is the use of IMEX schemes to solve equations of the type

$$\frac{\partial \phi}{\partial t} = f(\phi) + \nu g(\phi) \tag{3.84}$$

where $f(\phi)$ is non-linear, but not stiff, $g(\phi)$ is stiff but linear, and ν is a diffusion constant. IMEX timesteppers can generally be split into those which combine explicit and implicit multistep methods, and those which combine explicit and implicit Runge-Kutta methods. The most basic IMEX timestepper combines Euler forwards with Euler backward. The Euler Forward Backward scheme is written as

$$\phi^{n+1} - \delta t\ g(\phi^{n+1}) = \phi^n + \delta t\ f(\phi^n), \tag{3.85}$$

which treats the non-linear, non-stiff term explicitly, and the linear, stiff term implicitly. To solve each timestep, typically the right hand side is first evaluated. Then the resulting linear system $(1 - \delta t\ g)\,\phi^{n+1} = [\phi^n + \delta t f(\phi^n)]$ is inverted to find ϕ^{n+1}. The non linear terms being kept on the right hand side allows them to be solved explicitly using data at t^n, which requires only that function evaluations be carried out. It is at this point that transforms from spectral space to real space,

and back again, are carried out in pseudospectral methods such as those used here. The linear terms on the left hand side can be solved directly by inverting a differentiation matrix, which in spectral space is trivial. One of the most popular IMEX schemes is CNAB (Ascher et al., 1995), which combines the second order implicit Crank-Nicolson (eq (3.68)) and explicit Adams-Bashforth (eq (3.69)) schemes. The scheme is defined as

$$\phi^{n+1} - \frac{\delta t}{2} g(\phi^{n+1}) = \phi^n + \frac{\delta t}{2}\left[3f(\phi^n) - f(\phi^{n-1})\right] + \frac{\delta t}{2} g(\phi^n). \tag{3.86}$$

Overall, this scheme is second order in time. Higher order multistep methods have been created, with a number of examples investigated in Ascher et al. (1995). Multistep methods tend to have smaller stability regions as the order of timestepper increases (Ascher et al., 1995, 1997), and require only one set of computations per time step.

The other major type of IMEX time steppers is IMEX Runge-Kutta. An IMEX Runge-Kutta scheme of stage s is defined a combination of the Butcher Tableaux of the implicit and explicit schemes

$$\begin{array}{c|cccc} c_1^I & a_{1,1}^I & & & \\ c_2^I & a_{2,1} & a_{2,2}^I & & \\ \vdots & \vdots & \ddots & \ddots & \\ c_s^I & a_{s,1}^I & \dots & a_{s,s-1}^I & a_{s,s}^I \\ \hline & b_1^I & \dots & b_{s-1}^I & b_s^I \end{array} \qquad \begin{array}{c|cccc} c_1^E & 0 & & & \\ c_2^E & a_{2,1}^E & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ c_s^E & a_{s,1}^E & \dots & a_{s,s-1}^E & 0 \\ \hline & b_1^E & \dots & b_{s-1}^E & b_s^E \end{array} \tag{3.87}$$

(e.g. Butcher, 2008). Here, E denotes the explicit part of the problem and I denotes the implicit part of the problem. the nodes. The scheme is implemented by first calculating $K_1^E = f(\phi^n)$, and then completing the following loop:

$$\begin{aligned} K_i^I &= g\left(\phi^n + \delta t \sum_{j=1}^{i} a_{ij}^I K_j^I + \delta t \sum_{j=1}^{i} a_{i+1,j}^E K_j^E\right) \\ K_{i+1}^E &= f\left(\phi^n + \delta t \sum_{j=1}^{i} a_{ij}^I K_j^I + \delta t \sum_{j=1}^{i} a_{i+1,j}^E K_j^E\right) \end{aligned} \tag{3.88}$$

for $i = 1$ to s, where K^I is found by solving the implicit system. ϕ^{n+1} is then found by combining the intermediate stages as

$$\phi^{n+1} = \phi^n + \delta t \sum_{i+1}^{s} b_j^I K_j^I + \delta t \sum_{i=1}^{s+1} b_j^E K_j^E. \tag{3.89}$$

In effect, a number of intermediate stages K are calculated, in similar manner to

the explicit Runge-Kutta scheme introduced above (eq (3.70)). However, in this case, the implicit stages depend on the explicit stages, and all bar the first explicit stage depend on the preceding intermediate result. Equation (3.89) then collects the intermediate results to complete the timestep. A number of IMEX Runge Kutta schemes can be found in the literature (Spalart et al., 1991; Ascher et al., 1997; Kennedy and Carpenter, 2003; Kassam and Trefethen, 2005).

The extra work carried out in each timestep for IMEX Runge-Kutta methods, when compared to IMEX multistep methods, is balanced out by the increased time step size that is available, such that similar computation times are required for each method. Therefore, when choosing a timestepper for Parareal, the need to restart the solver repeatedly makes Runge-Kutta the preferred choice. Restarting multistep IMEX solvers requires many small steps of a low order timestepper in order to create the required information from previous timesteps.

3.2.3 Dedalus Code

Dedalus (Burns et al., 2020) is an open source spectral solver written in python. It uses the FFTW (Frigo and Johnson, 2005) library to perform fast parallel transforms between real and spectral space, and parallelizes in space over $n-1$ dimensions of an n dimensional domain using the mpi4py library (Dalcín et al., 2005). A number of different IMEX time stepping methods up to 4^{th} order are implemented, consisting of both Runge-Kutta and Multistep methods. There are a number of different spectral bases implemented, with Fourier, and Sine/Cosine series for periodic domains, and Chebyshev Polynomials, and Legendre/Hermite/Laguerre polynomials used for setting more complicated boundary conditions. Linear terms are treated in spectral space, whilst non-linear, and multiplicative terms are transformed into real space for evaluation.

3.3 Parareal

3.3.1 Algorithm

In the following section we will describe the parallel-in-time algorithm Parareal, describe its operation, and discuss some of its performance parameters. Parareal is

an iterative method for solving initial value problems (IVP's) of the form

$$\frac{\partial \phi}{\partial t} = f\left(\phi(t), t\right), \quad \phi(0) = \phi_0, \quad 0 \leq t \leq \mathcal{T} \tag{3.90}$$

where the right hand side of f typically comes from spatially discretizing a partial differential equation (PDE). In this work, as discussed in §3.2.2, ϕ contains the spectral coefficients of the various state variables: $(\widehat{\boldsymbol{u}}, \widehat{T}, \widehat{\boldsymbol{B}})$. The time domain is split into $\mathrm{NP}_{\mathrm{Parareal}}$ time slices, and each slice is determined by the time-points $0 = t_0 < t_1 < \cdots < t_{\mathrm{NP}_{\mathrm{Parareal}}-1} < t_{\mathrm{NP}_{\mathrm{Parareal}}} = \mathcal{T}$. Each time slice can be of different length, but here we restrict to the case where each time slice is of equal size, $\Delta\mathcal{T} = \mathcal{T}/\mathrm{NP}_{\mathrm{Parareal}}$.

Parareal facilitates the parallelisation of computation in time by utilising two different numerical solvers. The fine solver $\mathcal{F}$ is the equivalent of the standard serial-in-time solver, with spatial resolution and timestep size δt required for desired accuracy. The coarse solver $\mathcal{G}$ is a computationally cheaper approximation with lower than the required accuracy, and timestep size Δt.

A Parareal simulation begins with the coarse solver integrating from the initial conditions at $t = 0$ through to $t = \mathcal{T}$. This gives an approximation $\phi_n^{k=0}$ for each time slice $n = 0, 1, \ldots, \mathrm{NP}_{\mathrm{Parareal}}$, where k denotes the Parareal iteration number and n the time slice. (Note that this is different notation than in §3.2.2, where n was used to denote individual timesteps of a standard timestepping algorithm). Using this notation, the initial conditions are $\phi_0^0 = \phi(0)$. Each processor (or set of processors, if parallelisation is used in space as well as time) is responsible for integrating the solution through one time slice. The initial coarse run can be performed on a single processor, with the results for each time slice distributed afterwards to the other processors, but more often each processor computes the coarse and fine solutions for its own time slice. $\phi_{n+1}^0 = \mathcal{G}(t_{n+1}, t_n, \phi_n^0)$ is the coarse result computed on each time slice, and becomes the initial condition for the subsequent timeslice, for both the fine and coarse solvers. The computation consists of taking a sufficient number steps of a standard timestepping algorithm with the coarse timestep to move from $t = t_n$ to $t = t_{n+1}$ ie.

$$\text{No. coarse steps} = \frac{\Delta\mathcal{T}}{\Delta t}. \tag{3.91}$$

Each processor then computes $\mathcal{F}\left(t_{n+1}, t_n, \phi_n^k\right)$ from its initial condition ϕ_n^k, from $t = t_n$ to $t = t_{n+1}$ using timestep δt. The fine solver can use the same standard algorithm as the coarse solver, but more often uses a higher order method. In either case, $\Delta\mathcal{T}/\delta t$ timesteps are computed. This fine stepping computation is the most

computationally expensive part of the algorithm, and can be run in parallel; each processor only needs ϕ_n^k in order to start computing the fine solution. A second run using the coarse propagator is now started on the first processor, and the Parareal correction iteration

$$\phi_{n+1}^{k+1} = \mathcal{G}(t_{n+1}, t_n, \phi_n^{k+1}) + \mathcal{F}(t_{n+1}, t_n, \phi_n^k) - \mathcal{G}(t_{n+1}, t_n, \phi_n^k) \tag{3.92}$$

is used to communicate information from the fine solver to all subsequent processors. This second coarse propagation has to run in serial, and is effectively the serial overhead in Parareal. Each subsequent parallel fine computation, followed by a serial coarse correction computation, makes up a Parareal iteration. Figure 3.1 shows a rough sketch of how the computing time of each processor is spent using Parareal compared to a serial computation.

3.3.2 Stopping Criteria, Accuracy

As $k \to \mathrm{NP}_{\mathrm{Parareal}}$, the solution $\phi^k_{n=\mathrm{NP}_{\mathrm{Parareal}}}$ converges to the solution that would have been obtained by running the fine solver in serial for the full duration, $\mathcal{F}(t = \mathcal{T}, t = 0, \phi_0^0)$. The algorithm stops when the error between the Parareal solution and the expected fine solution is of similar magnitude to the error inherent in the fine solver (due to discretisation in time and space). However, the error is not trivial to find, as in realistic applications, the fine solution will not be available. As a proxy for this accuracy check, a common method of checking convergence is to compare the solution at k with the solution at $k-1$. When the defect between these solutions is of similar magnitude to error inherent in the fine solver, the solution is deemed to be converged. It was proven in Gander and Vandewalle (2007) that when $k = \mathrm{NP}_{\mathrm{Parareal}}$, the solution from the Parareal algorithm will be identical to the solution obtained from the fine solver in serial. A brief description of how Parareal converges at $k = \mathrm{NP}_{\mathrm{Parareal}}$ follows. The main prerequisite is that both the fine and coarse solvers are stable and always compute the same output for a given input. On the first timeslice, ϕ_0 is the initial condition for both the coarse and fine solver. When the first Parareal correction sweep is computed, the coarse solver again uses ϕ_0 as the initial condition; $\phi_0^1 = \phi_0^0$. Therefore,

$$\mathcal{G}(t_1, t_0, \phi_0^1) = \mathcal{G}(t_1, t_0, \phi_0^0). \tag{3.93}$$

This means that

$$\begin{aligned}\phi_1^1 &= \mathcal{G}(t_1, t_0, \phi_0^1) + \mathcal{F}(t_1, t_0, \phi_0^1) - \mathcal{G}(t_1, t_0, \phi_0^0) \\ &= \mathcal{F}(t_1, t_0, \phi_0^1) + \mathcal{G}(t_1, t_0, \phi_0^1) - \mathcal{G}(t_1, t_0, \phi_0^0) \\ &= \mathcal{F}(t_1, t_0, \phi_0^1). \end{aligned} \tag{3.94}$$

In subsequent Parareal iterations, $\phi_1^{k+1} = \phi_1^k$, so does not change. This will be used in all subsequent iterations as the initial condition for the second time slice, for both the coarse and fine solvers. The coarse solver on the second time slice will then also compute its result twice from the same initial conditions. The coarse results in the Parareal correction (eq (3.92)) will cancel out, leaving only the result from the fine solver. This will lead to the same behaviour in the third time slice on the subsequent iteration, and so on and so on. Thus, with every iteration of Parareal, the result of one additional time slice becomes the exact result of the fine solver. When $k = \mathrm{NP}_{\mathrm{Parareal}}$, this has propagated through every time slice, so that

$$\phi_{\mathrm{NP}_{\mathrm{Parareal}}}^{k=\mathrm{NP}_{\mathrm{Parareal}}} = \mathcal{F}(\mathcal{T}, 0, \phi_0^0). \tag{3.95}$$

Convergence in $k = \mathrm{NP}_{\mathrm{Parareal}}$ iterations, however, would not lead to speed up, and for speed up to be obtained, an acceptable solution must be found within a small number of Parareal iterations, as will be discussed in the next section.

3.3.3 Speedup and performance

Parareal speed up can be estimated as runtime of the time-serial (likely to be parallelised in space) method over the runtime of the time-parallel method. Some simplifying assumptions are made in the following analysis. First, that all $\Delta\mathcal{T}$ are equal. Second, that fixed timestep sizes are used for both Δt and δt. The communication times are also assumed to be zero. Finally, due to these assumptions, it is assumed that the load is balanced perfectly across the Parallel-in-time processors. The assumption of equal $\Delta\mathcal{T}$ size neglects any load imbalance, and this can be significant. Fixed timestep sizes can also ignore the effect of load imbalance: for systems with sudden physical changes, there could be large differences in the number of timesteps taken by each processor. An example of how load imbalance can effect Parareal performance is shown in figure 3.2. Neglecting communication times is not so problematic in Parareal. The serial portion of Parareal is relatively large: the coarse solver. Typically, many coarse, serial time steps are computed on each processor in every Parareal iteration, whilst only one communication is required for each pro-

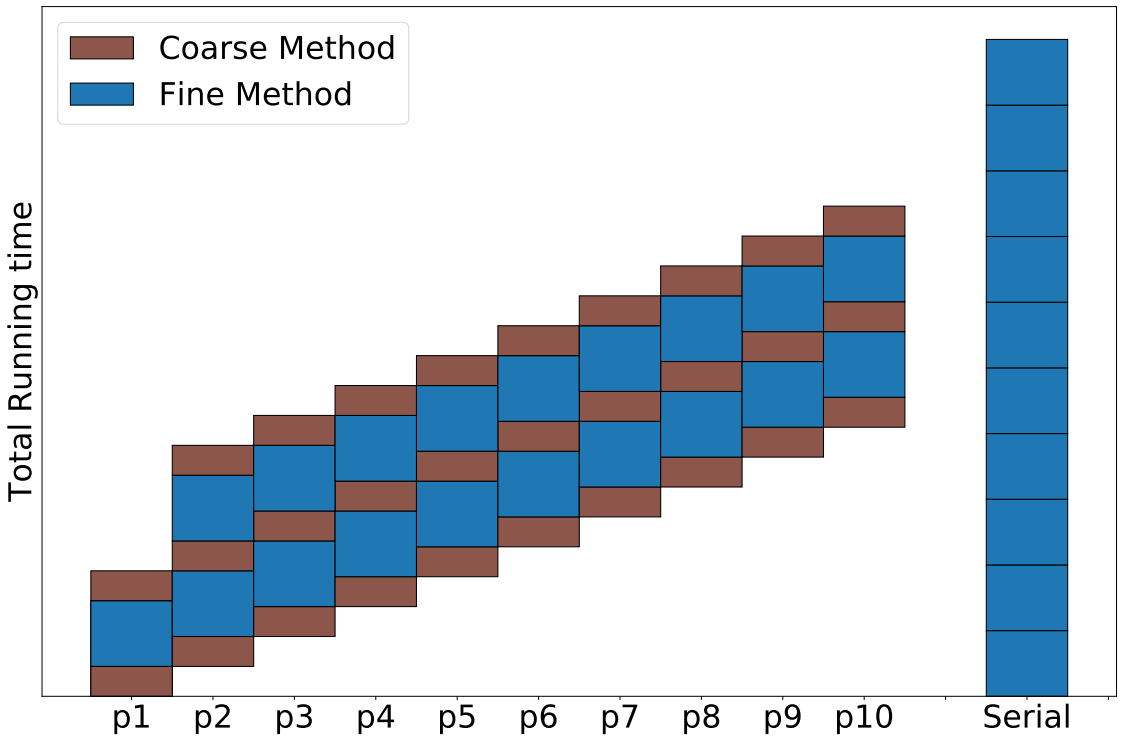
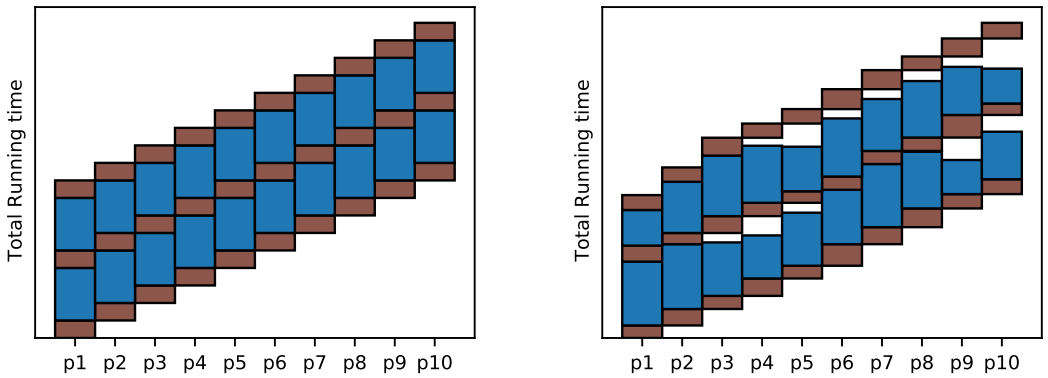


Figure 3.1: Processing time for each processor in Parareal (left) compared with processing time for the same simulation carried out in serial (right). Note that here we have two iterations, and the coarse stepper takes around half the time of the fine stepper.



(a) Load balanced

(b) Load imbalanced

Figure 3.2: Effect of load imbalance on Parareal. Example plot of two simulations both converging in 2 iterations, with a fine/coarse runtime ratio of 3. The white gaps showing on the imbalanced chart indicate processor idle time, due to waiting for previous processors. Blue indicates fine computations, brown indicates coarse computations, and pN represents the processor time slice.

cessor per iteration. Therefore, serial coarse timestepping is usually far more time consuming than communication (Aubanel, 2011). There are however times where this is not the case, and real world performance should always be measured using real simulations on real systems where possible, so that all possible bottle necks are accounted for (Goetschel et al., 2021). Under these assumptions, and with the runtime of the fine method over a timeslice $\Delta\mathcal{T}$ labelled $R_{\mathcal{F}}$, and the runtime of the coarse method over the same interval labelled $R_{\mathcal{G}}$, the total runtime of a serial simulation can be estimated as $\mathrm{NP}_{\mathrm{Parareal}} R_{\mathcal{F}}$. Total runtime of the Parareal solver can be estimated as

$$\mathrm{NP}_{\mathrm{Parareal}} \times R_{\mathcal{G}} + k\left(R_{\mathcal{G}} + R_{\mathcal{F}}\right). \tag{3.96}$$

This gives the speed up as

$$S_P = \frac{\mathrm{NP}_{\mathrm{Parareal}} \times R_{\mathcal{F}}}{\mathrm{NP}_{\mathrm{Parareal}} \times R_{\mathcal{G}} + k\left(R_{\mathcal{G}} + R_{\mathcal{F}}\right)}, \tag{3.97}$$

which can be simplified to

$$S_P = \left[\left(1 + \frac{k}{\mathrm{NP}_{\mathrm{Parareal}}}\right)\frac{R_{\mathcal{G}}}{R_{\mathcal{F}}} + \frac{k}{\mathrm{NP}_{\mathrm{Parareal}}}\right]^{-1}. \tag{3.98}$$

It is evident from this equation that a larger ratio between the coarse and fine runtimes leads to a larger speed up. It can also be seen that larger k leads to smaller speed ups. From this estimate, a bound can be found for the speed up

$$S_P \leq \min\left\{\frac{\mathrm{NP}_{\mathrm{Parareal}}}{k}, \frac{R_{\mathcal{F}}}{R_{\mathcal{G}}}\right\} \tag{3.99}$$

(Gander and Vandewalle, 2007; Minion, 2011). This bound illustrates the trade off which needs to be made to obtain good speed up from Parareal. The number of iterations needs to be kept as low as possible, which means the coarse solver should be close in accuracy to the fine solver. However, at the same time, the ratio of the runtimes of the fine and coarse solver should be as large as possible, so the coarse solver should be as computationally cheap as possible. Finding a good compromise coarse solver, which is cheap 'enough' and accurate 'enough' is the key to speed up in Parareal. The parallel efficiency of Parareal is found as

$$E_P = \frac{S_P}{\mathrm{NP}_{\mathrm{Parareal}}} = \left[\frac{R_{\mathcal{G}}}{R_{\mathcal{F}}}\left(\mathrm{NP}_{\mathrm{Parareal}} + k\right) + k\right]^{-1}, \tag{3.100}$$

from which another bound can be inferred

$$E_P \leq \frac{1}{k}. \tag{3.101}$$

This highlights further the need for as low a number of iterations as possible, in order to keep the parallel efficiency as high as possible.

Chapter 4

Parareal Kinematic Dynamo

In this chapter we present results from our work on speeding up simulations of the kinematic dynamo using Parareal. We simulate two different dynamos, the steady Roberts (1972) dynamo, and the time periodic Galloway and Proctor (1992) dynamo. In Chapter 1 we introduced the kinematic dynamo and the Parareal algorithm, and showed the motivation for this work. In Chapter 2 we performed a review of recent studies in both these areas, whilst in Chapter 3 we examined the mathematical details of the dynamo and Parareal. Here, we begin with a brief description of the models used, before showing how the numerical models were validated. We then discuss the accuracy levels required for numerical results, before reporting our results. We finish with a short discussion. The work in this chapter was previously published in Clarke et al. (2020b).

4.1 Description and Validation

Here, we briefly describe the model that we will be using, and provide more details of the Roberts (1972) and Galloway and Proctor (1992) dynamos in particular. We solve the induction equation

$$\frac{\partial \boldsymbol{B}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{B} = \boldsymbol{B} \cdot \boldsymbol{\nabla} \boldsymbol{u} + \frac{1}{\mathrm{Rm}} \nabla^2 \boldsymbol{B}, \tag{4.1}$$

for magnetic field $\boldsymbol{B}$ with a pre-determined velocity $\boldsymbol{u}$ which is divergence-free ($\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$), with magnetic Reynolds number Rm.

The Roberts (1972) flow is defined as

$$\boldsymbol{u} = (B\cos(y), A\sin(x), B\sin(y) + A\cos(x)), \tag{4.2}$$

where A, B are arbitrary constants (set to 1 here). The domain is a periodic square plane in x and y of side length 2π. The flow is $2.5D$ (as described in §2.2.3), with z components, but no dependence on z. The magnetic field variation in the z direction is then found through the wavenumber k_z, which is treated as a parameter, so that we seek solutions of the form $\boldsymbol{B}(x, y, z, t) = \boldsymbol{b}(x, y, t)\mathrm{e}^{ik_z z}$. Derivatives in z thus reduce to a multiplication by ik_z. Substituting the flow into the induction equation gives

$$\frac{\partial b_x}{\partial t} + \left(\cos(y)\frac{\partial}{\partial x} + \sin(x)\frac{\partial}{\partial y} + (\sin(y) + \cos(x))\mathrm{i}k_z\right) b_x = \\ - b_y\sin(y) + \frac{1}{\mathrm{Rm}}\left(\nabla^2 - k_z^2\right) b_x, \tag{4.3}$$

$$\frac{\partial b_y}{\partial t} + \left(\cos(y)\frac{\partial}{\partial x} + \sin(x)\frac{\partial}{\partial y} + (\sin(y) + \cos(x))\,\mathrm{i}k_x\right) b_y = \\ b_x\cos(x) + \frac{1}{\mathrm{Rm}}\left(\nabla^2 - k_z^2\right) b_y, \tag{4.4}$$

$$\frac{\partial b_x}{\partial x} + \frac{\partial b_y}{\partial y} + \mathrm{i}k_z b_z = 0, \tag{4.5}$$

where b_x, b_y, b_z are the x, y, z components of the magnetic field, respectively.

The Galloway and Proctor (1992) flow is defined as

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} C\sin(z + \sin(\omega t)) + B\cos(y + \cos(\omega t)) \\ C\cos(z + \sin(\omega t)) \\ B\sin(y + \cos(\omega t)) \end{pmatrix} \tag{4.6}$$

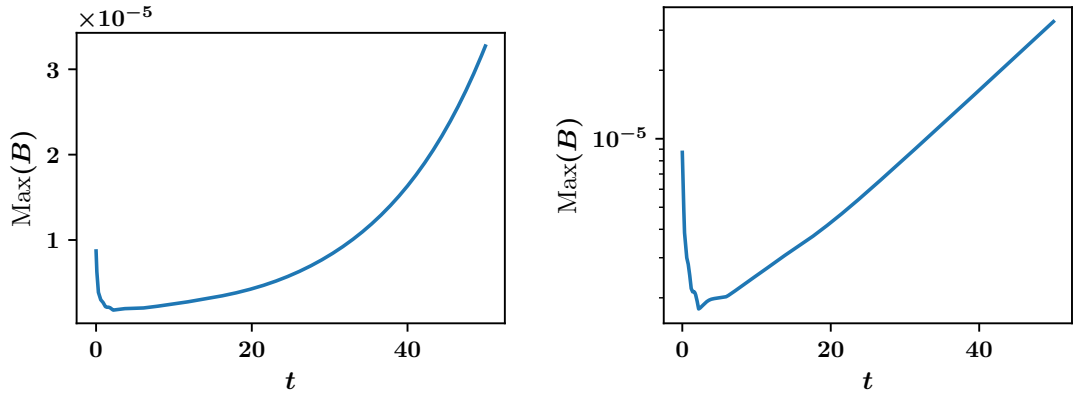
where $B = C = \sqrt{3/2}$ and $\omega = 1$ are arbitrary constants. The domain is a periodic plane in y and z of length 2π, and the flow is also $2.5D$, as there is no dependence on x. The magnetic field is of the form $\boldsymbol{B}(x, y, z, t) = \boldsymbol{b}(y, z, t)\mathrm{e}^{ik_x x}$, so that k_x is the parameter rather than k_z (as in the Roberts flow). Substitution into the induction equation gives

$$\frac{\partial b_y}{\partial t} + \left(uik_x + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z}\right) b_y = b_z\frac{\partial v}{\partial z} + \frac{1}{\mathrm{Rm}}\left(\nabla^2 - k_x^2\right) b_y, \tag{4.7}$$

$$\frac{\partial b_z}{\partial t} + \left(u\mathrm{i}k_x + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} \right) b_z = b_y \frac{\partial w}{\partial y} + \frac{1}{\mathrm{Rm}} \left(\nabla^2 - k_x^2 \right) b_z, \tag{4.8}$$

$$\mathrm{i}k_x b_x + \frac{\partial b_y}{\partial y} + \frac{\partial b_z}{\partial z} = 0. \tag{4.9}$$

For both the Roberts (1972) and Galloway and Proctor (1992) models, initial conditions for $\boldsymbol{B}$ were set to random values $\sim 10^{-5}$, as in Archontis et al. (2003). Time was measured in units of the turnover time for both cases. When the simulation begins, there is an initial drop in magnetic field strength, as the random field is not the optimal configuration for growth in either problem. After a short time period (typically a few time units), the magnetic field begins to grow. For the Roberts flow, the growth is exponential (showing as linear on a log-linear plot, see fig 4.1), whilst for the Galloway Proctor flow, the growth is exponentially growing and periodic, but can be averaged out over a long enough period to give a single growth rate. After the initial drop in magnetic field strength, the average field strength grows until it is back above $\sim 10^{-5}$ of the initial conditions (for values of Rm with positive growth rates). Calculation of the growth rate can be carried out by ignoring the initial transient, and tracking the magnetic field strength after this period. In this work, we ran all simulations to 50 time units to ensure that the field strength was higher than initial conditions, and calculated the growth rate over the period 25-50 time units, to ensure that the initial drop in field strength was ignored. An example plot showing how max($\boldsymbol{B}$) changes over time for the Roberts flow can be found in figure 4.1.



(a) Linear plot of B over time

(b) Log plot of B over time

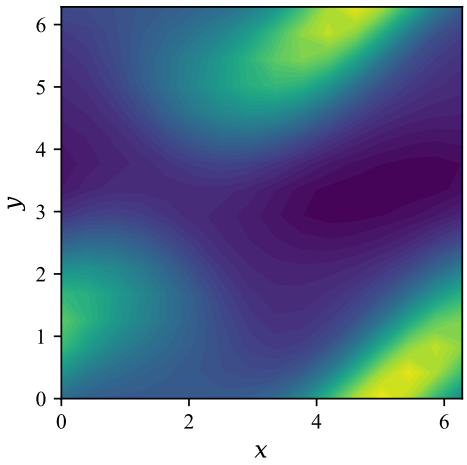
Figure 4.1: Example plot of how **B** changes over time, in linear and log scales. The gradient of the log scale graph can be used to find the growth rate.

Figure 4.2 shows the contours of the y-component of the magnetic field for $R_m = 3$ and 3000 in the Galloway-Proctor flow, and $R_m = 4$ and 4096 in the Roberts flow. Larger scale and more diffuse structures are present in the $R_m = 3$

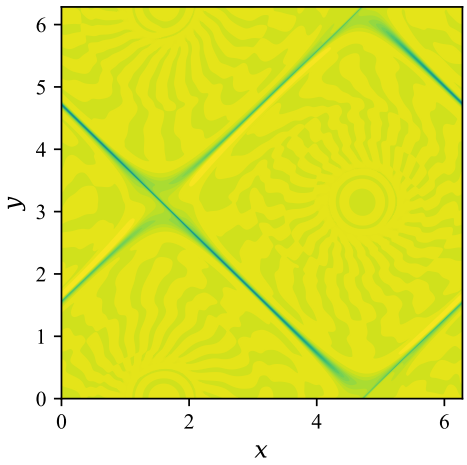
and $R_m = 4$ simulations. Finer structures emerge in the $R_m = 3000$ and $R_m = 4096$ cases, showing the effect of the ${R_m}^{-1/2}$ scaling on the smallest structures. The Galloway-Proctor magnetic field morphology changes over time, with the periodic change in the velocity field, which leads to the more complicated pattern shown in Figure 4.2d. For the Roberts flow, the magnetic field has a constant morphology and simply grows exponentially in magnitude.

The (time serial) numerical model that we used in numerical tests was validated by comparing the growth rates against results in the literature. Roberts growth rates were compared with those reported by Plunian and Rädler (2002), for different Rm and k_z. We also calculated the growth rate for a range of k_z, for each Rm, and were able to reproduce the behaviour reported in Roberts (1972) and Jones (2008), see figure 4.3. We first wrote a code using just Python mathematical libraries, and then moved on to creating the same model in Dedalus (see §3.2.3 for details), and growth rates were equal in the literature, Python code and Dedalus model (see figure 4.4).

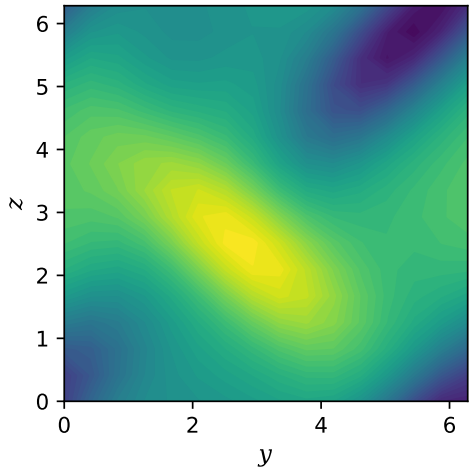
The Galloway Proctor model was validated against results from Galloway and Proctor (1992) and analysis printed in Charbonneau and Steiner (2012), the peak growth rate is found at $k_x = 0.57$ for all Rm, and does not decrease with increasing Rm, showing the expected fast-dynamo behaviour. See figure 4.5 for details.



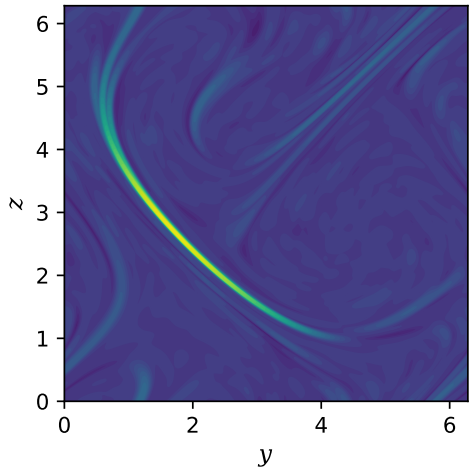
(a) Roberts flow, $R_m = 4$



(b) Roberts flow, $R_m = 4096$



(c) Galloway-Proctor flow, $R_m = 3$



(d) Galloway-Proctor flow, $R_m = 3000$

Figure 4.2: Contour plots of the y-component of the magnetic fields for the labelled flows at time $T = 50$. The left hand side shows the low R_m field whilst the right hand plots show the high R_m fields. Much finer structures are apparent in the high R_m cases, due to the ${R_m}^{-1/2}$ scaling of the spatial structures. The Galloway Proctor field shows more spatial variability, due to the time dependence of the flow. The Roberts field stays effectively fixed in space, only varying in magnitude over the course of the simulation, whilst the morphology and magnitude of the Galloway Proctor field changes over time.

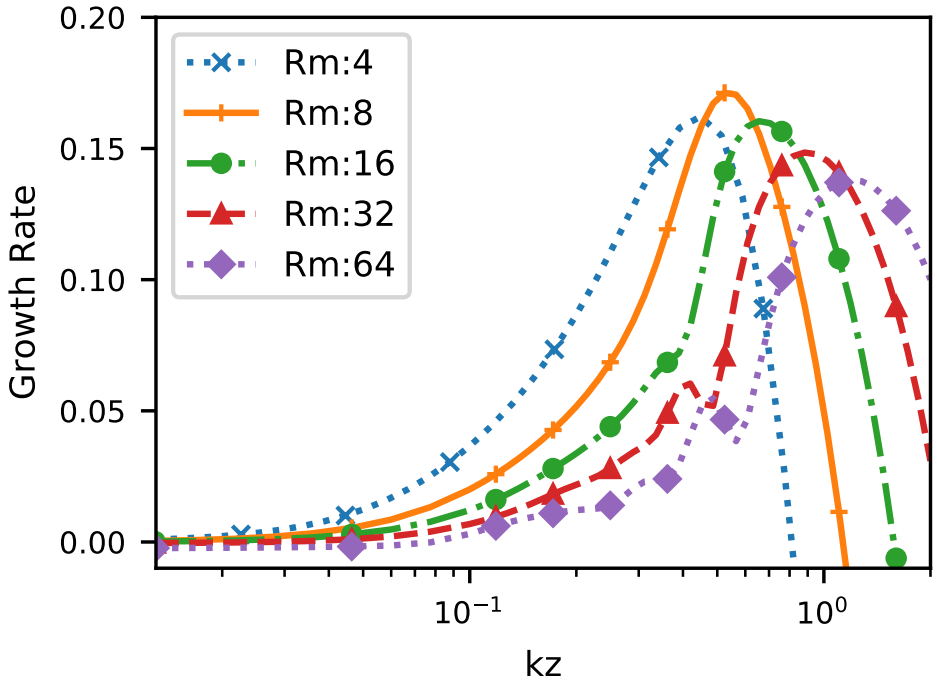


Figure 4.3: Growth rate of the Roberts dynamo for different Rm over a range of k_z. Each Rm has a peak growth rate at a different k_z, and for Rm > 8, these growth rates fall as Rm increases, showing slow dynamo behaviour. Results match those found in Roberts (1972) and Jones (2008).

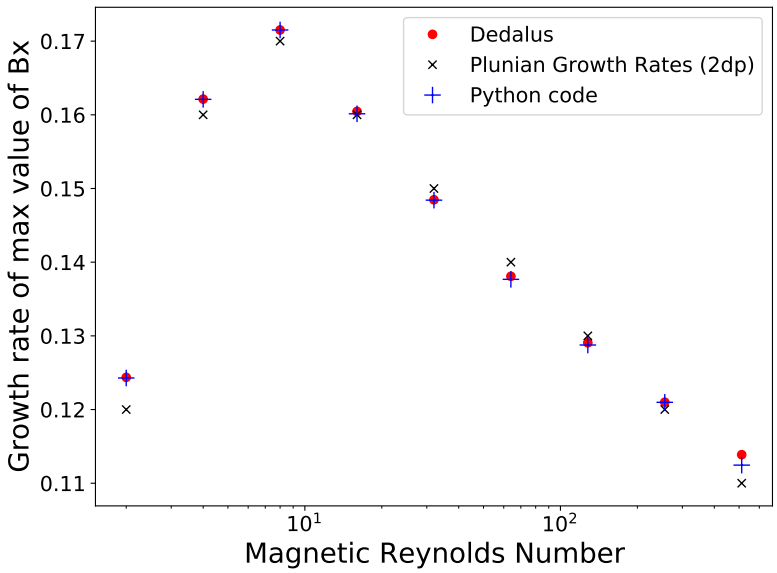


Figure 4.4: Comparison of peak growth rates calculated by Dedalus, simple python code, and reported in the literature, for different Rm. Results from literature (Plunian and Rädler, 2002) were reported to 2 decimal places.

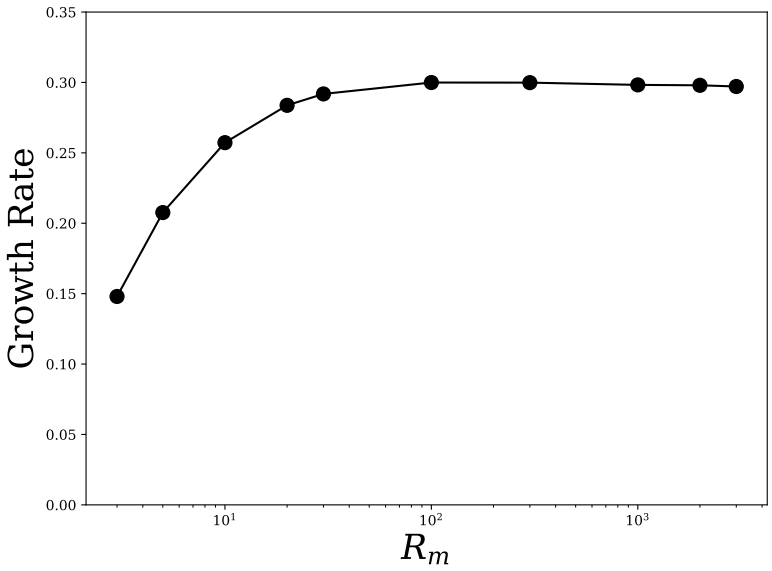


Figure 4.5: Growth rates calculated for Galloway and Proctor (1992) dynamo at $k_x = 0.57$. Growth rate does not fall as Rm $\to \infty$, and matches Charbonneau and Steiner (2012).

4.2 Accuracy measurement

Determination of the level of accuracy in a Parareal simulation is crucial to measuring performance accurately. Parareal is determined to have converged when the error between consecutive iterations is below a given threshold. This threshold however should be linked to the error expected from the solution overall. This error is equivalent to the error expected from the fine solver, if it were run in serial over the whole time period. If the Parareal convergence threshold is set much higher or much lower than the error inherent in the fine solver, then results will not be very meaningful. Performance of the fine solver is also important. To obtain realistic speed up measurements, Parareal must be measured against the optimum serial solver for a given problem and accuracy. Dynamo calculations require high accuracy, as under-resolved simulations can produce erroneous growing magnetic fields (Tobias, 2021), such as the solutions of Bullard and Gellman (1954). We denote ϕ_N as the solution vector containing b_x and b_y (for the Roberts dynamo), b_y and b_z (for the Galloway Proctor dynamo), in real space, with N_x, N_y, N_z collocation points in x, y, z directions, respectively. We simplify notation by defining $N = N_x = N_y = N_z$, the spatial resolution of a simulation. Errors are estimated by comparison with a high resolution reference solution $\phi^{\text{reference}}$ in terms of the L^2 norm

$$||\phi - \phi^{\text{reference}}||_2 = \left[\sum_i^N (\phi_i - \phi_i^{\text{reference}})^2\right]^{1/2}, \tag{4.10}$$

where ϕ_i represents the i^{th} entry of the solution vector with N entries. The error ε we report is the L^2 norm of the error, normalised by the L^2 norm of the solution vector

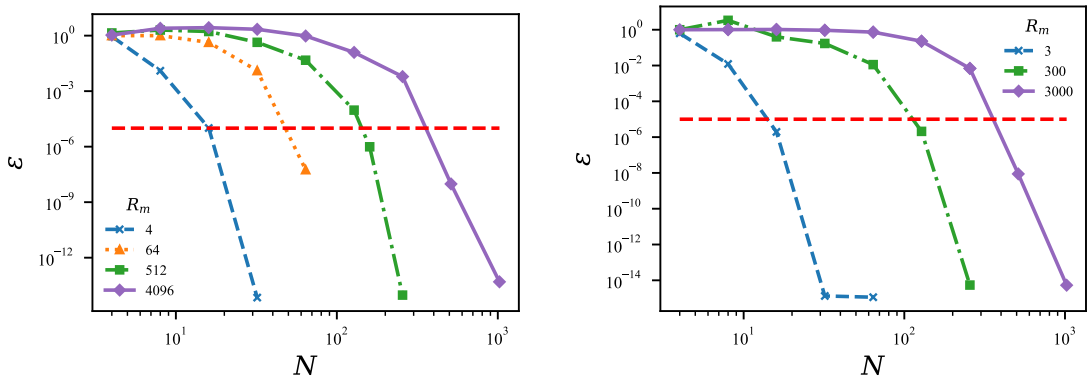
$$\varepsilon = \frac{||\phi - \phi^{\text{reference}}||_2}{||\phi^{\text{reference}}||_2}. \tag{4.11}$$

Spectral interpolation is used to allow the computed solution to be compared to the reference solution. Given the need for highly accurate results in dynamo simulations, and our aim in this work to inform more complicated studies that could be carried out in future, we aim for an accuracy of 10^{-5}.

4.2.1 Fine solution

First, we need to fix the required spatial resolution for $\mathcal{F}$. Convergence in space was tested by running simulations at double the previous spatial resolution until the normalised L^2 difference between two solutions was $\sim 10^{-15}$ (machine precision). At this point, the error from the spatial discretisation is of the order of machine precision. The most highly resolved solution, was then used as the reference solution $\phi^{\text{reference}}$ to compute relative error of the solution computed with the solution computed for each resolution ϕ_N. The results are shown in Figure 5.4, confirming the expected spectral convergence behaviour. For each Rm, we set the fine solver spatial resolution, $N_{\mathcal{F}}$, to the smallest value that gives a solution with error smaller than 10^{-5} (indicated by the dashed red line). Because higher magnetic Reynolds numbers produce smaller scale features, they require better spatial resolution to match our error tolerance.

Next, we fix the time stepping method and time step. IMEX Runge-Kutta methods were chosen, as they offer comparable performance to IMEX multistep methods (see §3.2.2) but are able to start without recourse to another time-stepping method. As Parareal requires each time slice to start from a new initial condition on every iteration, for both fine and coarse solvers, this is an important factor. Creating a reference solution with a temporal error of the order of machine precision proved to be unfeasible, due to computational constraints, especially in the higher Rm cases. Therefore, a result with error lower than 10^{-7} was used as a reference solution for setting the time step of the fine solver, δt. This is two orders of magnitude smaller than the desired result of 10^{-5} and should provide an accurate estimation of the error due to time-stepping. A comparison of the Runge-Kutta time steppers available in Dedalus is shown in Figure 4.7. Because RK443 reaches the required tolerance of 10^{-5} with the smallest number of evaluations of the right hand side function, it is the most efficient choice. Similar results were found for other magnetic Reynolds



(a) Roberts Flow

(b) Galloway Proctor Flow

Figure 4.6: Graphs showing spatial convergence of the solvers for the two flows investigated. Spectral order of convergence is observed, with the decrease in error ε accelerating as the number of spectral modes, N, is increased. The line at 10^{-5} shows the required level of accuracy in the solution. The number of modes required for each Rm follows the predicted $\mathrm{Rm}^{1/2}$ scaling.

numbers. We therefore use RK443 for the fine method $\mathcal{F}$ throughout this work.

Figure 4.8 shows the dependence of the solution error on time step size for a range of Rm for the Roberts and Galloway-Proctor flows. All simulations were carried out with the optimal $N_{\mathcal{F}}$ found from the spatial resolution study (fig 5.4) . We can see that smaller step sizes are required to meet a given level of accuracy for the Galloway-Proctor flow than for the Roberts flow. This is likely due to the Galloway-Proctor flow depending explicitly on time. In the case of the Roberts flow, a δt small enough to satisfy the stability requirements for a given $N_{\mathcal{F}}$ is sufficient to also satisfy the accuracy requirement of 10^{-5}, except for the most simple case of $\mathrm{Rm} = 4$. This affects the performance of Parareal through the ratio of computational run times $R_{\mathcal{F}}/R_{\mathcal{G}}$ because we have to use essentially the same time step for both the coarse and fine method. In contrast, for the Galloway-Proctor simulations, satisfaction of the stability requirement did not guarantee accuracy within the required tolerance, and a smaller δt must be used for the fine solver, leading to a better coarse-to-fine computation time ratio.

4.2.2 Coarse Solver

For each Rm, the $N_{\mathcal{F}}$ and δt determined above are used in the fine solver. We now discuss the different possibilities available for choosing a coarse solver. Using the same spatial resolution with coarse time step $\Delta t > \delta t$ was not suitable for

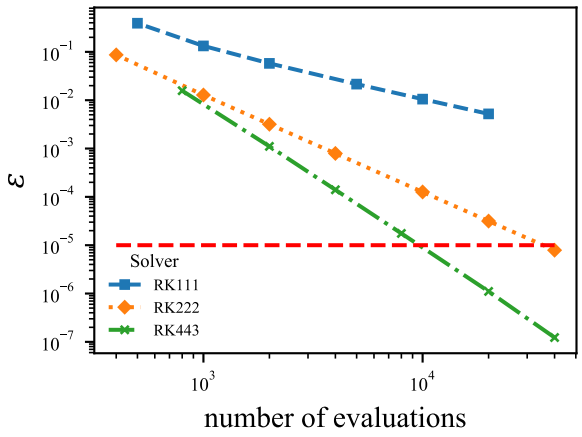
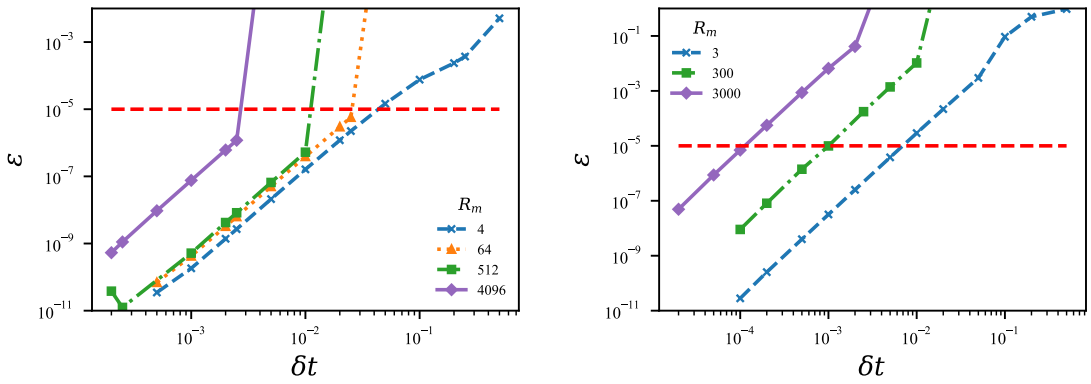


Figure 4.7: Work required for different time stepping methods to obtain solutions of a certain accuracy, measured as the relative two-norm ε of the each solution $\phi_{\delta t}$ and the solution obtained using the smallest timestep with the RK443 stepper ($\phi_{\delta t}^{\min}$). The number of evaluations required to compute from $T = 0$ to $T = 1$ are shown. This number depends on δt and the number of stages in each time-stepping method. For any degree of accuracy better than 10^{-2}, the RK443 time-stepper requires fewer evaluations than either RK111 or RK222. Results shown are for the Galloway-Proctor flow with $R_m = 300$, $N_{\mathcal{F}} = 256$.

the Roberts flow, as δt was the largest stable time step for a given resolution. It was also unsuitable for the Galloway-Proctor simulations, as the ratio of $\Delta t/\delta t$ was not large enough to give meaningful speedup. Use of a fully implicit coarse solver was rejected, as the spectral spatial discretisation meant that a dense matrix solve would be required at each time step. This large increase in computational complexity would reduce the difference in computations required between the fine and coarse solvers, leading to smaller speed ups. There was little scope to attempt to use reduced physics in this study, as we are already considering the simplest form of dynamo problem. However, this strategy may be useful in further work on a non-linear dynamo. Coarsening in both space and time was found to be the most promising strategy. As time step stability is linked to spatial resolution, reducing spatial resolution allows a larger time step to be taken, even where the fine solver is at the largest stable time step. This means that coarse resolution $N_{\mathcal{G}} < N_{\mathcal{F}}$ and $\Delta t > \delta t$, opening up the possibility of a large difference in computational complexity between the coarse and fine solvers. However, too aggressive coarsening will lead to a very inaccurate coarse solver and slow convergence. The most efficient amount of spatial coarsening was studied by carrying out Parareal simulations with a wide range of coarse method spatial resolution.

Simulations were carried out for the Roberts flow with Rm = 512. This is moderately high, whilst allowing relatively modest compute resources to be used.



(a) Roberts Flow

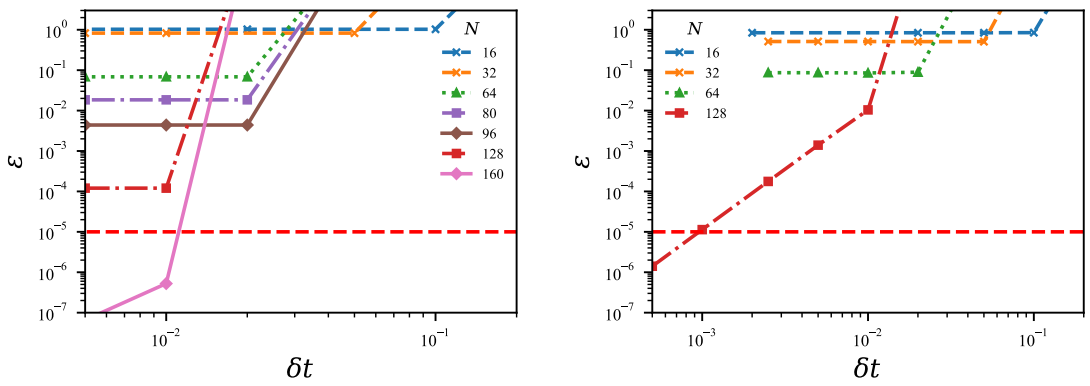
(b) Galloway Proctor Flow

Figure 4.8: Error ε with respect to time step size for the different flows and Rm simulated. The Roberts flow, which is independent of time, shows high accuracy for the highest stable time step for all simulations except Rm = 4. The Galloway Proctor flow has a larger error for the same time step size. This is believed to be because of the incorporation of time on the right hand side of the equations. Galloway-Proctor flows therefore require smaller time step sizes to reach the desired accuracy. Where the error goes past the top of the figure, the solver has diverged and is unstable for this time step size.

The fine solver parameters were fixed, with $N_{\mathcal{F}} = 160$, and $\delta t = 10^{-2}$, while the coarse step Δt was set to the highest stable step for the given $N_{\mathcal{G}}$. This was found by estimating the error at different time steps for each resolution, as shown in Figure 4.9a. A similar study was carried out for the Galloway-Proctor flow (Figure 4.9b). The number of Parareal time slices $\mathrm{NP}_{\mathrm{Parareal}}$ was fixed at 10. Figure 4.10 shows that the peak speed up is acquired when $N_{\mathcal{G}} = 0.5N_{\mathcal{F}}$. When $N_{\mathcal{G}} < 0.5N_{\mathcal{F}}$, the speed up is reduced by the extra number of Parareal iterations required to converge, and when $N_{\mathcal{G}} > 0.5N_{\mathcal{F}}$, the difference in computational complexity between the coarse and fine solvers is insufficient.

4.3 Scaling Results

Scaling tests were carried out for both the Roberts flow and the Galloway-Proctor flow. Simulations of the Roberts flow were carried out on the ARC 3 HPC facility at the University of Leeds, made up of Intel Xeon E5-2650v4 (Broadwell) CPUs, with a total of 6,048 cores. Simulations of the Galloway-Proctor flow were carried out on the ARCHER HPC facility, made up of Intel Xeon E5-2697v2 (Ivy Bridge) CPUs, with a total of 109,056 cores.



(a) Roberts Flow, $R_m = 512$ (b) Galloway-Proctor Flow, $R_m = 300$

Figure 4.9: Error ε vs. time step size δt for a range of spatial resolutions (N) for the Roberts and Galloway-Proctor flows. Accuracy is constrained by spatial resolution, until the finest resolution is reached in each case. As the resolution reduces, the largest stable time step increases as expected. Error increases above 10^1 (off the top of the graph) indicate that the method has become unstable at that time step. Accuracy for a given resolution/ time step size is higher for the Roberts flow than the Galloway-Proctor flow.

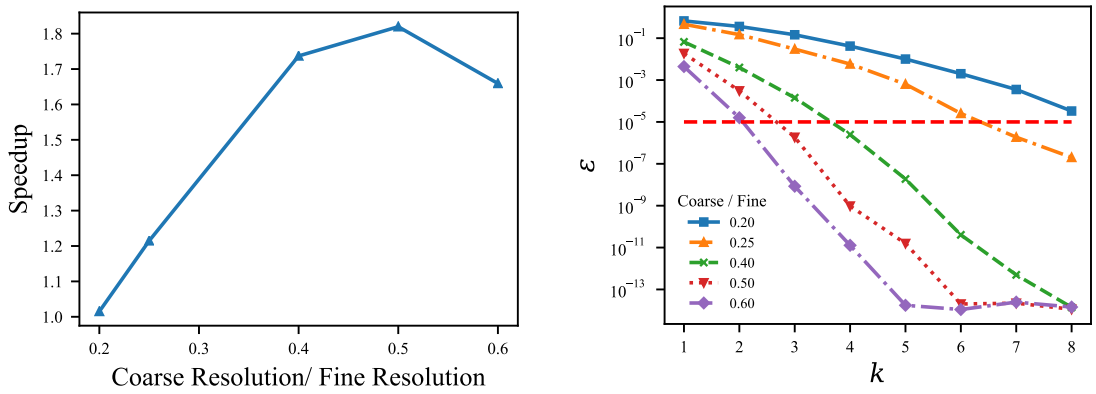
Table 4.1: Parameters for simulations. Rm: magnetic Reynolds number, k_z: wave number in z co-ordinate, k_x: wave number in x co-ordinate, $N_{\mathcal{F}}$: number of modes in fine propagator, $N_{\mathcal{G}}$: number of modes in coarse propagator, δt: time step for fine propagator, Δt: time step for coarse propagator, Growth rate indicates growth rate of the magnetic field.

Flow	Rm	k_z	k_x	$N_{\mathcal{F}}$	$N_{\mathcal{G}}$	δt	Δt	Growth Rate
Roberts	512	2.87		160	80	10^{-2}	2×10^{-2}	0.11
	4096	7.5		512	256	2.5×10^{-3}	5×10^{-3}	0.097
Galloway-Proctor	3		0.57	16	8	5×10^{-3}	10^{-1}	0.15
	300		0.57	128	64	10^{-3}	2×10^{-2}	0.3
	3000		0.57	512	256	10^{-4}	5×10^{-3}	0.3

A range of Rm were simulated to see the effect on Parareal performance (see Table 4.1). Scaling performance was compared with pure spatial scaling of the Dedalus solver. Fully parallel in space and in time simulations were also carried out in order to show how Parareal can increase scalability beyond the saturation of spatial scaling.

4.3.1 Roberts Flow

Scaling results for the Roberts flow are shown in Figure 4.11 for Rm = 512 (upper figures) and Rm = 4096 (lower figures). For both values of Rm, both parallel scaling

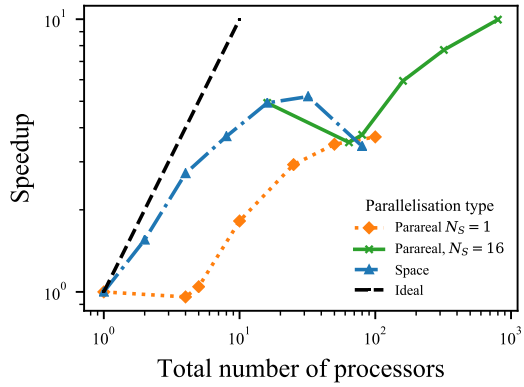


(a) Speedup

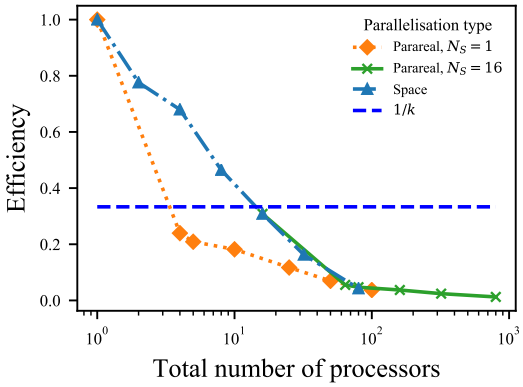
(b) Convergence

Figure 4.10: (a) Speedup vs ratio of coarse, fine spatial resolutions for Roberts flow, with Rm = 512, $N_{\mathcal{F}} = 160$, and $\delta t = 10^{-2}$. Long run times are found for very low coarse resolutions as the estimated solution is not accurate enough to allow quick convergence. As $N_{\mathcal{G}}$ increases, the run time reduces due to the reduced number of Parareal iterations required to converge. The best performing coarse solver has a resolution of $0.5N_{\mathcal{F}}$. Further increasing the resolution of the coarse solver increases the complexity of the coarse solver to a level close to that of the fine solver, reducing any speed up possible. (b) Graph showing how defect to previous solution ε changes with number of Parareal iterations (k) for different resolutions of the coarse solver. Very low resolutions results in Parareal taking many iterations to converge, reducing opportunity for speed up. High resolutions show quicker convergence.

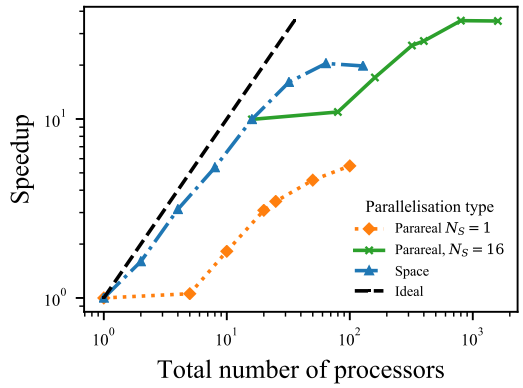
and efficiency in space are superior to Parareal at low processor counts. As expected, spatial scaling is better for the Rm = 4096 case with higher spatial resolution, due to higher workload per processor. While Parareal alone is not competitive, in both cases a combined space-time parallelization generates slightly more speedup than a pure spatial parallelization. The theoretical maximum efficiency for Parareal is 1/3, indicated by a horizontal dashed line, due to the simulation requiring three iterations to converge. However, because of the relatively expensive coarse solver, Parareal's observed efficiency is mostly substantially lower. As the efficiency of the combined space-time parallelisation is the product of the parallel in space efficiency and the parallel in time efficiency, it is low for high numbers of processors because of the low efficiency of Parareal for the Roberts flow. Despite the larger overall speedup, with efficiencies below 0.1, space-time parallelization using Parareal may not be particularly attractive .



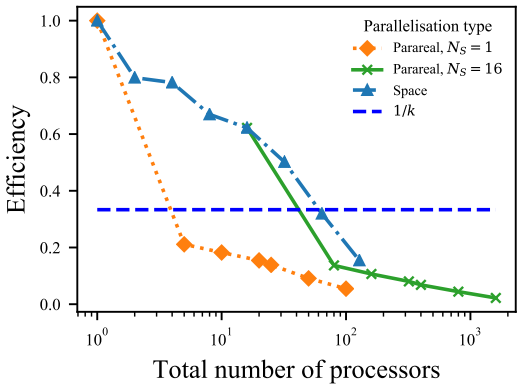
(a) Speed up, Rm = 512



(b) Efficiency, Rm = 512



(c) Speed up, Rm = 4096



(d) Efficiency, Rm = 4096

Figure 4.11: Speed up (a), (c) and parallel efficiency (b), (d) of the Parareal method compared with spatial parallelisation for simulations of Roberts flow with Rm=512 (a), (b) and Rm=4096 (c), (d). Total number of processors is calculated as number of processors in space ($\mathrm{NP}_{\mathrm{Space}}$, N_S in graphs) multiplied by number of processors for Parareal ($\mathrm{NP}_{\mathrm{Parareal}}$, N_P in graphs). Speed up and efficiency are both poor for low numbers of processors for Parareal, but as the parallelisation in space saturates, further gains can be made from Parareal, although they are small. Parareal does not offer any gain over parallelisation in space until parallel efficiency is less than 0.1, and does not come close to the theoretical maximum of $1/k$, where k is number of Parareal iterations.

4.3.2 Galloway Proctor Flow

Results for Rm= 3, 300 and 3000 for the Galloway Proctor flow are shown in Figure 4.12. Performance of Parareal is much better than for the Roberts flow. Parareal speed up is competitive with spatial parallelisation at a relatively low number of processors. In all of these cases, the number of iterations required to converge was three, so that the efficiency is bounded by 1/3, and the efficiency of Parareal stays close to this bound over a range of $\mathrm{NP}_{\mathrm{Parareal}}$, and does not fall much as the number of processors increases. The poor performance of Dedalus in parallelising the Rm=3 case is attributable to the fact that there are only 16^2 spectral modes. In the Rm = 3000 case using 32 processors in space, the results show that speed up above that of spatial parallelisation alone is possible, with an efficiency around 0.16. In all cases, Parareal has not yet reached saturation in its scaling performance, and has almost ideal scaling behaviour, except for a constant offset due to the bounds on Parareal scaling. This is shown in Fig. 4.13, where the efficiency of the method is tracked over the different Rm. Efficiency is close to the bound of 1/3, and Parareal efficiency does not fall with increasing Rm. Pure Parareal efficiency was estimated in the $R_m = 3000$ case by dividing by the efficiency of the spatial parallelisation found for that particular $\mathrm{NP}_{\mathrm{Space}}$ (32). In that case, total parallel efficiency is lower than for the other Galloway-Proctor cases due to the combination of the spatial and temporal parallelisation, and is approximately the product of the two, as expected. This reduction in overall efficiency is unavoidable, as parallel in space is more efficient than Parareal for lower numbers of processors, and Parareal only becomes competitive after spatial efficiency falls away. Also shown on this Figure are the efficiencies obtained at different Rm for the Roberts flow, highlighting the difference in performance of the method for the two cases.

4.4 Discussion

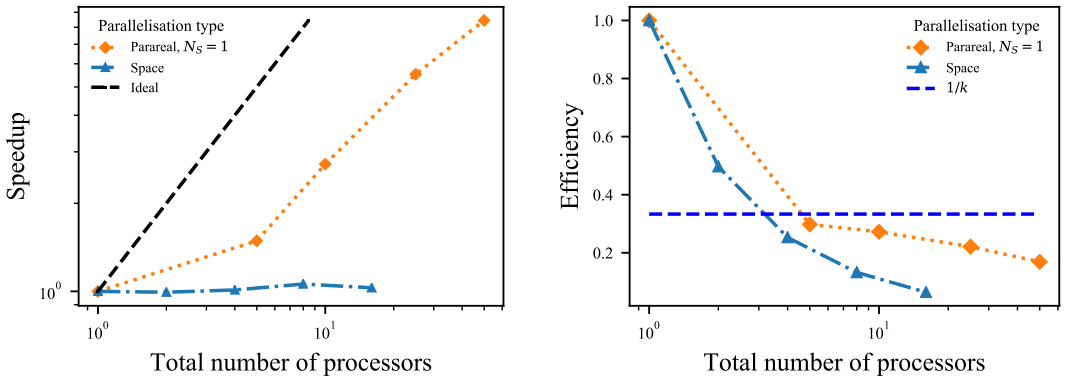
The Parareal algorithm has been found to offer parallel speed up for kinematic dynamo simulations beyond what can be achieved through spatial parallelisation alone. In the case of the simpler steady Roberts dynamo, the speed up is modest and parallel efficiencies are low. Here, owing to the steady nature of the imposed velocity, the difference in computational complexity of the coarse and fine methods is found to be too small for good performance of Parareal. The issue was that the time step size was not a limiting factor on the accuracy of the fine solver; as long as the time step was stable, it was within the accuracy required. Therefore, there was little

room to use a coarser resolution for the coarse propagator in Parareal. Performance for the time-dependent Galloway-Proctor flow was better and the efficiency stayed close to the theoretical limit over a wide range of magnetic Reynolds numbers, while scaling well to large numbers of processors. In this problem, since evolution of the magnetic field depends explicitly on the current time, the accuracy of the solution depends more on the size of the time step. This means that a time step in the coarse solver much larger than that of the fine step is possible, allowing for better Parareal performance.

Fully coupled dynamic dynamo simulation is complicated, and has non-linear dependencies, and so the accuracy of the fine solver is expected to behave more like the Galloway Proctor flow. Therefore, the good performance of Parareal for the Galloway-Proctor flow suggests that good performance may be possible also for more complex dynamos.

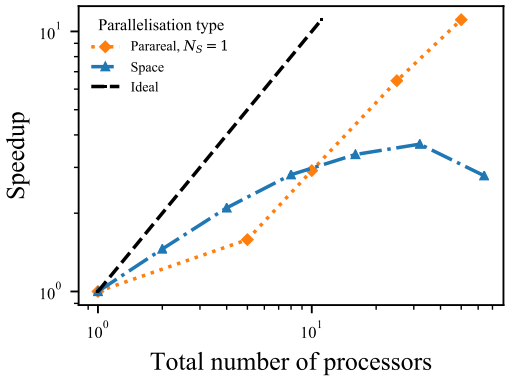
The parallel efficiency of the Parareal algorithm applied to the Galloway Proctor dynamo is close to the theoretical maximum of $1/k$. This means that the overheads due to communication are small, in comparison to the serial cost of the coarse method, pointing to an efficient implementation of the algorithm. Performance of the algorithm when applied to this problem does not appear to degrade with increased Rm, as can be seen in Figure 4.13. Performance has remained constant, with Parareal efficiency not much lower than $1/3$ for $R_m = 3$, $\mathrm{Rm} = 300$ and $\mathrm{Rm} = 3000$. This is noteworthy since highly advective problems are thought to cause problems with Parareal convergence, but this has not yet been found in the the highly advective case with Rm up to $\sim 10^3$.

The results we have found encourage us to look further at Parallel-in-time methods for dynamo simulations. Two possible avenues seem promising. The performance of other parallel-in-time methods for the Roberts and Galloway-Proctor flows could be compared to that found here for Parareal. Conversely, the performance of Parareal for a more complicated physical system could be investigated. Both of these topics offer promising and interesting research questions, and we believe that either would be a valid choice moving forward. However, due to the linearity of the kinematic dynamo problem making it quite different to the full dynamo problem, we believe that study of a non-linear system would be most informative for the dynamo community, who we feel are the main audience for this work. It is for this reason that we move on to study of Rayleigh-Bénard convection in the next chapter.

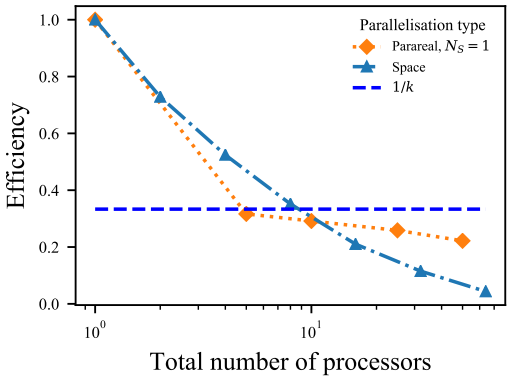


(a) Rm = 3, Speed up

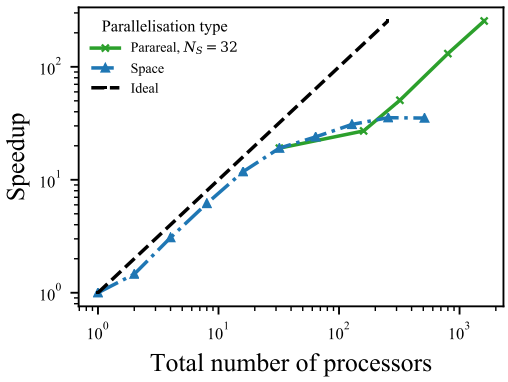
(b) Rm = 3, Efficiency



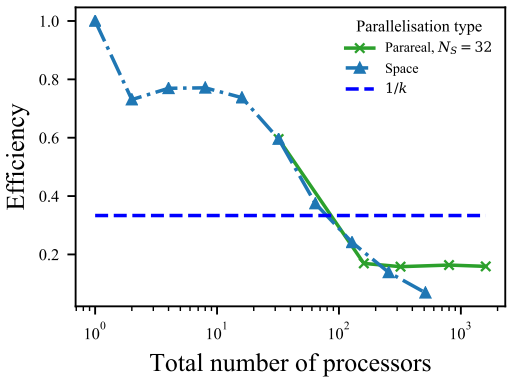
(c) Rm = 300, Speed up



(d) Rm = 300, Efficiency



(e) Rm = 3000, Speed up



(f) Rm = 3000, Efficiency

Figure 4.12: Speed up and parallel efficiency of the parareal method compared to parallelisation in space for Rm of 3, 300, and 3000, Galloway Proctor flow. Total number of processors is calculated as number of processors in space ($\mathrm{NP}_{\mathrm{Space}}$, N_S in graphs) multiplied by number of processors for Parareal ($\mathrm{NP}_{\mathrm{Parareal}}$, N_P in graphs). In the case of Rm=3000, parareal simulations were carried out with 32 processors in space, as serial runs with one processor were time intensive. Spatial resolutions required were 16^2, 128^2, and 512^2 respectively. Results here are more promising than in the Roberts flow. Parareal becomes more efficient than spatial parallelisation for smaller processor numbers, and keeps higher efficiency for longer, closer to the theoretical maximum of $1/k$ (k: number of Parareal iterations). Scaling saturation for parareal has not been reached even at 1600 processors in the Rm=3000 case.

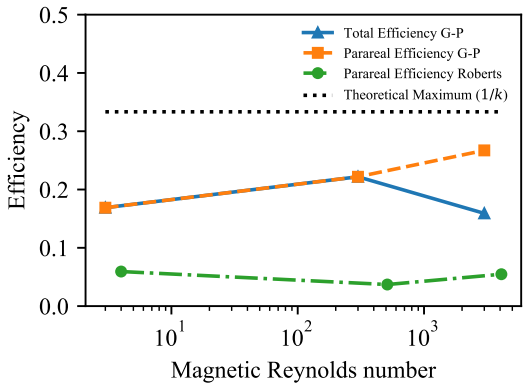


Figure 4.13: Parallel efficiency vs Rm for Galloway-Proctor and Roberts dynamos. Galloway-Proctor results show higher efficiency than the Roberts flow. Parallel efficiency of the method does not appear to degrade with increasing Rm. There is a reduction for total efficiency for Rm=3000, however, this is due to a combination of the efficiency of the spatial parallelisation with the parareal efficiency. Efficiency of parareal alone is comparable to the efficiency of the lower Rm simulations in the Galloway-Proctor case.

Chapter 5

Parareal Rayleigh Bénard Convection

In this chapter we present results on measuring the performance of the Parareal algorithm when applied to the problem of 2D Rayleigh-Bénard Convection. We investigate how the performance changes with increasing Rayleigh number, and discuss how the results indicate whether Parareal should be investigated further as an option for speeding up numerical simulations of dynamos. Chapter 1 introduced the dynamo problem, parallel-in-time methods, and the problem of Rayleigh-Bénard convection. Chapter 2 provided a review of recent parallel-in-time applications. Chapter 3 discusses the mathematical details of both Parareal and Rayleigh-Bénard convection. In the previous chapter, (Chapter 4) we reported results of our work on Parareal and the kinematic dynamo problem, with some results leading us to conclude that further work was warranted. We decided upon investigation of a more complicated physical system, rather than other parallel-in-time algorithms for the same system, leading to the work reported in this chapter. The work in this chapter was previously published in Clarke et al. (2020a).

5.1 Description of Model

Rayleigh-Bénard convection describes movement of a fluid driven by a temperature difference ΔT between two horizontal plates separated by distance h. We use the Boussinesq approximation to the Navier-Stokes equations for fluid flow in a 2D Cartesian domain. The non-dimensional Oberbeck-Boussinesq equations modelling

Rayleigh - Bénard convection can be written as

$$\frac{1}{\mathrm{Pr}}\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u})\right) = -\boldsymbol{\nabla}p + \mathrm{Ra}T\cdot\hat{\boldsymbol{z}} + \nabla^2\boldsymbol{u}, \tag{5.1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{5.2}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}T = \nabla^2 T, \tag{5.3}$$

with vertical boundaries defined as

$$T|_{z=-0.5} = 1, \quad T|_{z=0.5} = 0, \tag{5.4}$$

$$\boldsymbol{u}|_{z=-0.5} = \boldsymbol{u}|_{z=0.5} = 0, \tag{5.5}$$

and periodic horizontal boundaries. Here, $\boldsymbol{u} = (u, w)$ represents the horizontal and vertical velocity of the fluid, T represents the temperature, t represents time and p is pressure. The fundamental time scale is taken as a thermal diffusion time $\tau = h^2/\kappa$, T is scaled by ΔT, and length is scaled by h. We use a domain of size $(x = 2,\ z = 1)$, where x is the horizontal direction, and z the vertical, giving an aspect ratio of 2. We begin with a linear temperature profile with small perturbations and $\boldsymbol{u} = 0$. The Prandtl number is $\mathrm{Pr} = \nu/\kappa$ while the Rayleigh number is

$$\mathrm{Ra} = \frac{\alpha g \Delta T h^3}{\nu\kappa}, \tag{5.6}$$

where ν is the kinematic viscosity, κ is the thermal diffusivity, α is the coefficient of thermal expansion, and g is gravity.

5.1.1 Consistency Checks

The Reynolds number can be computed from the velocity of the fluid. A characteristic speed U is determined as $\overline{\langle u^2 + w^2\rangle}^{1/2}$ where the overbar denotes the time average and $\langle\rangle$ the volume average. Our parameters are chosen such that $\mathrm{Re} = U$.

The heat transported due to convection is represented by the Nusselt number

$$\mathrm{Nu}_V = \frac{1}{V}\int\left(-\frac{\partial T}{\partial z} + wT\right)\mathrm{d}V, \tag{5.7}$$

where the subscript V indicates that it has been calculated using a volume integral over the domain. A Nusselt number of 1 indicates that all heat transport is due to conduction, whilst Nusselt > 1 indicates advection is present. A larger Nusselt number indicates more heat transport by advection.

In order to confirm the accuracy of our simulations, we carry out three internal consistency checks. We calculate the Nusselt number in three ways. First, integrated over the domain volume via Equation 5.7. Second, on the bottom plate via

$$\mathrm{Nu}_b = \left\langle -\frac{\partial T}{\partial z} \right\rangle_H \Bigg|_{z=-0.5}, \tag{5.8}$$

where $\langle a \rangle_H = L_x^{-1} \int_{x=0}^{x=L_x} a \, \mathrm{d}x$ is a horizontal plane average. Third, on the top plate via

$$\mathrm{Nu}_t = \left\langle -\frac{\partial T}{\partial z} \right\rangle_H \Bigg|_{z=0.5}. \tag{5.9}$$

Conservation of energy requires

$$\mathrm{Nu} = \overline{\mathrm{Nu}_b} = \overline{\mathrm{Nu}_t} = \overline{Nu_V}, \tag{5.10}$$

(King et al., 2012). The standard test in the literature is for the Nusselt numbers calculated at different heights of the domain to be within 1% of each other (Stevens et al., 2010; King et al., 2012; Mound and Davies, 2017). In this work, the reported values have been calculated from equation 5.7.

Thus, we calculate the maximum relative difference between the bulk Nusselt number and the Nusselt numbers at the top, bottom as well as the difference between the top and bottom Nusselt number

$$\varepsilon_{\mathrm{Nu}} = \frac{\max\left(|\overline{\mathrm{Nu}_b} - \overline{\mathrm{Nu}_V}|, |\overline{\mathrm{Nu}_b} - \overline{\mathrm{Nu}_t}|, |\overline{\mathrm{Nu}_V} - \overline{\mathrm{Nu}_t}|\right)}{\overline{\mathrm{Nu}_V}}. \tag{5.11}$$

As a second consistency check, we verify that buoyancy generation is balanced with viscous dissipation. If we average over a sufficiently long time, the $\frac{D\boldsymbol{u}}{Dt}$ term of the momentum equation goes to zero. We then take the dot product of the momentum equation with $\boldsymbol{u}$ and integrate to find the energy balance

$$|\overline{\boldsymbol{u} \cdot \nabla^2 \boldsymbol{u}}| = |\overline{\boldsymbol{u} \cdot \mathrm{Ra} T \hat{\boldsymbol{z}}}| \tag{5.12}$$

where the first term represents the viscous dissipation ϵ_U, and the second term represents the buoyancy production P, not to be confused with p for pressure. The standard test in the literature is for simulations to find these quantities within 1% of each other (King et al., 2012; Mound and Davies, 2017). We check this by calculating

$$\frac{|P - \epsilon_U|}{P}. \tag{5.13}$$

As a third test, we make sure that the boundary layers are resolved with a minimum number of nodes. The thermal boundary layer can be defined using the peak value of T_{rms}, calculated as

$$T_{\text{rms}}(z) = \overline{\left\langle \sqrt{\left(T - \overline{\langle T \rangle_H}\right)^2} \right\rangle_H} \tag{5.14}$$

as in King et al. (2013). Figure 5.1 shows the relationship between T_{rms} and the thermal boundary layers, and the relationship between the viscous boundaries and the mean horizontal velocity magnitude. The thickness of the thermal boundary layer δ_T is defined by the height at which the peak value of T_{rms} occurs. The boundary layer scales with the Nusselt number as

$$\delta_T = \frac{1}{2} h \text{Nu}^{-1}, \tag{5.15}$$

(see Grossmann and Lohse (2000)). The thermal boundary layers play a significant role in the behaviour of Rayleigh-Bénard convection, and it is essential that they are fully resolved in any numerical simulation (Shishkina et al., 2010). Amati et al. (2005) showed that at least 4 grid points are required in the thermal boundary layer, while Verzicco and Camussi (2003) stated that 6 points are needed. Stevens et al. (2010) say that up to 7 points could be the minimum number of points required. In this work, we specify that at least 6 points are in the boundary layer. The number of points in the thermal boundary layer will be denoted as N_{BL}.

Figure 5.2 shows example temperature fields for the cases we study, at a snapshot in time after the flow has equilibriated. It also shows the different temperature profiles found in these cases (bottom), and compares them to the linear conductive state. We can see that as Ra increases, the profile becomes more uniform in the bulk, with a steeper temperature gradient in the boundary layers.

5.2 Validation of model

The code was validated against the data in Johnston and Doering (2009), see Figure 5.3. Both fixed flux and fixed temperature boundary conditions were simulated. We calculated a Rayleigh Nusselt scaling of $\text{Nu} = 0.135\text{Ra}^{0.286}$ from our fixed flux data, very close to the $\text{Nu} = 0.138\text{Ra}^{0.285}$ reported in Johnston and Doering (2009). The slightly higher Nusselt numbers obtained in Johnston and Doering (2009) for fixed flux cases at low Rayleigh number were also replicated. Finally, we calculated the critical Rayleigh number by running multiple simulations near Ra_c (Chandrasekhar, 1961), and checking the growth rate of the kinetic energy. We found that

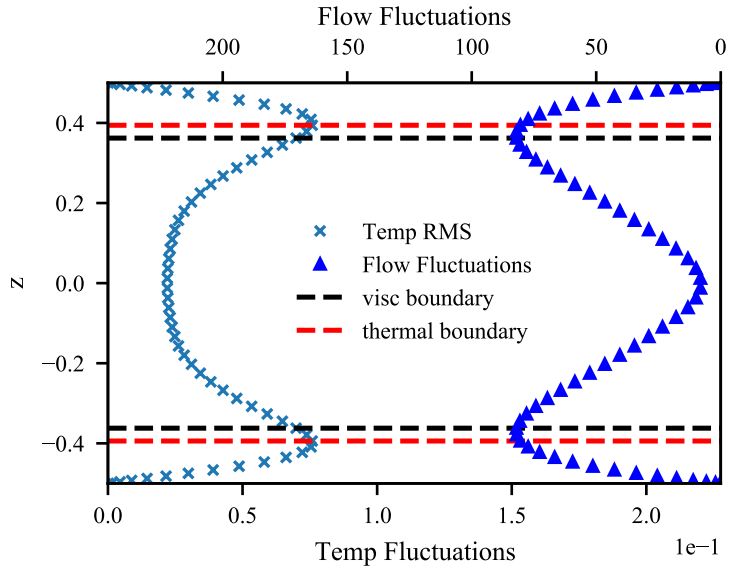


Figure 5.1: Rayleigh Bénard flow at Rayleigh number = 10^5. Temperature fluctuations (left side of graph, bottom scale) denote the T_{rms} of the temperature field (defined in text), U_{Mean} (right side of graph, top scale) denotes the magnitude of the horizontal component of the velocity. The thermal boundary layer is defined by the height at which the peak T_{rms} is found, and the viscous boundary layer is defined by the height at which the peak U_{mean} is found King et al. (2013).

it was in agreement with Chandrasekhar (1961) to within 0.1%,

5.2.1 Determining Accuracy of Fine Solution

We set a tolerance level of less than 1% for ε_{Nu} defined in Equation 5.11 and $|P - \epsilon_U|/P$ defined in Equation 5.13. We also require a minimum of 6 points in the thermal boundary layers, that is $N_{\text{BL}} \geq 6$. At each Ra we start with a low resolution ($(N_x, N_z) = (16, 8)$ for Ra $= 10^5$ and 10^6 and $(32, 16)$ for Ra $= 10^7$) and then double the resolution in both spatial directions until all three conditions are met.

For comparison, we also carry out spatial convergence tests for the L^2 norm of the temperature field, comparing results obtained from the low resolution simulations with those obtained from a high resolution simulation for each Ra. These are not used to determine the spatial resolution. We calculate the relative difference in the final state temperature field by taking the L^2 norm with the high resolution (double resolution of shown values for each Ra) final state. The second test is for

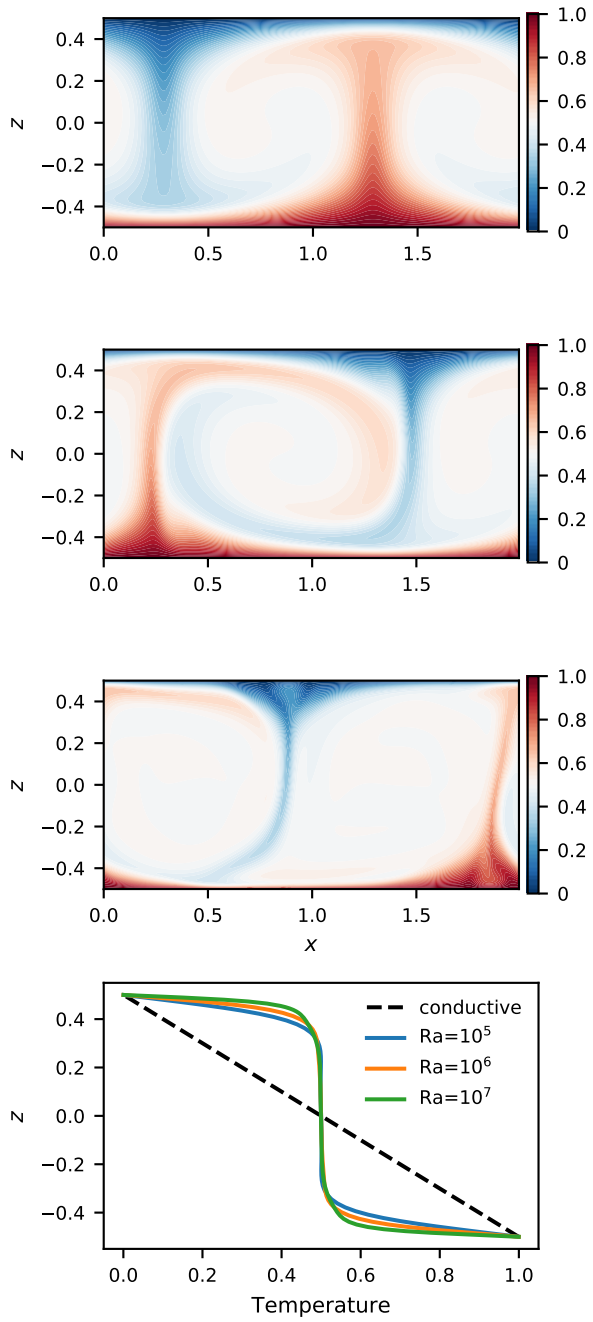


Figure 5.2: Temperature field for flows with Ra $= 10^5$ (top), 10^6 (middle top), and 10^7 (middle lower) taken after a statistically steady state has been reached. The bottom plate is fixed at $T = 1$, whilst the top plate is fixed at $T = 0$, and both top and bottom plates are no-slip. There is steady flow for Ra $= 10^5$, with more unsteady and smaller plumes at 10^6, and even more so at 10^7. At Ra $= 10^7$, there is a small amount of entrainment of fluid into the base of the plumes. The bottom figure shows temperature profiles for all three cases, compared to the purely conductive case. Boundary layers get thinner as Ra increases.

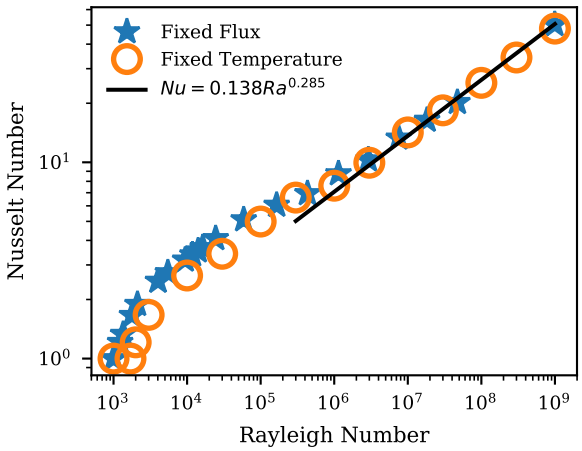


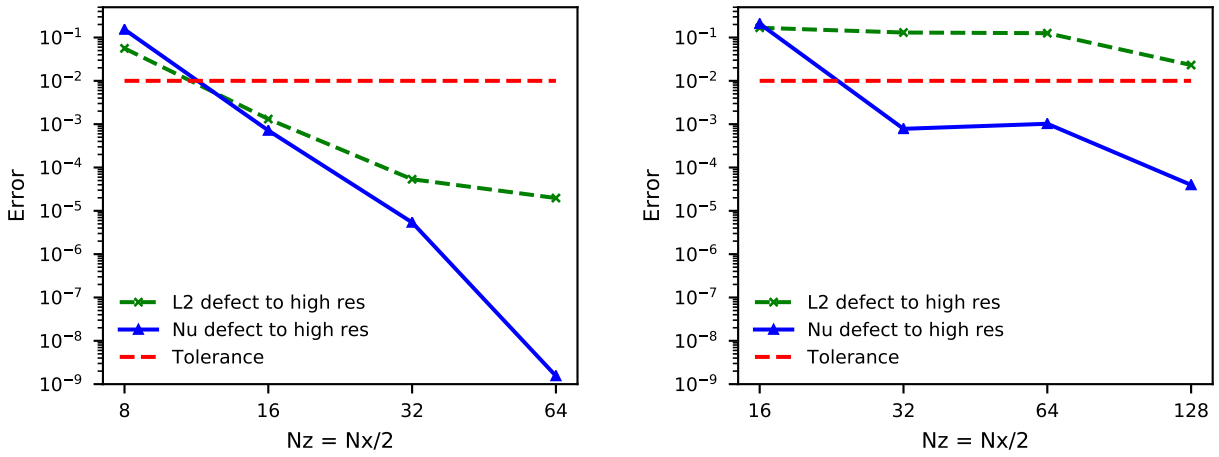
Figure 5.3: Calculated Nusselt values ($\overline{\mathrm{Nu}}_V$) compared with the scaling found in Johnston and Doering (2009). Scaling of 0.135 $\mathrm{Ra}^{0.286}$ was calculated from our data, compared to 0.138 $\mathrm{Ra}^{0.285}$ found in Johnston and Doering (2009). Fixed temperature and fixed flux boundary simulations collapse on to the same line at high Rayleigh number, in agreement with Johnston and Doering (2009) (black line).

Nu, for which we calculate

$$\varepsilon_{\mathrm{Nu}}^{\mathrm{rel}} = \frac{\left|\mathrm{Nu} - \mathrm{Nu}^{\mathrm{reference}}\right|}{\mathrm{Nu}^{\mathrm{reference}}}, \tag{5.16}$$

where $\mathrm{Nu}^{\mathrm{reference}}$ is a reference solution obtained from a high resolution simulation.

Table 5.1 shows the resolution required to meet the consistency checks discussed above. We can see that the resolution required for 6 points in the boundary layer is higher than the resolution required for the other convergence tests, except for the L^2 error for $\mathrm{Ra} = 10^7$. Figure 5.4 shows how the L^2 error compares with $\mathrm{Nu}_{\mathrm{int}}$ ϵ_{Nu}. At $\mathrm{Ra} = 10^5$, the resolution for a 1% L^2 error is the same as the resolution required for the 1% tolerance in the Nusselt numbers and buoyancy production and only half the resolution needed to have at least six nodes in the boundary layers. At $\mathrm{Ra} = 10^7$, the L^2 error is not yet below 1% even when all other tests are below tolerance, showing a significant difference in the L^2 error and the convergence tests we have set. Given the tests set in the literature for Rayleigh-Bénard convection, which do not generally include analysis of L^2 error, if the internal checks and key quantities are converged before the L^2 error, then the lower resolution is deemed sufficient. We will revisit this subject in Chapter 6, as L^2 was used as the convergence test for the kinematic dynamo in Chapter 4. The effect of timestep size on the accuracy of the solution was also investigated. However, it was found that for a given spatial resolution, the largest stable timestep was found to meet all of the accuracy criteria.



(a) $Ra = 10^5$ (b) $Ra = 10^7$

Figure 5.4: Spatial convergence of Nusselt number $\overline{\mathrm{Nu}}_V$ and L^2 errors relative to high resolution solution for Ra $= 10^5$ (left) and Ra $= 10^7$ (right). As expected, higher resolution is required for both quantities to meet the 10^{-2} tolerance for the higher Rayleigh number case. It can also be seen that the L^2 error requires much more resolution at higher Rayleigh number than the Nusselt number, where as at $Ra = 10^5$, the resolution required to give good answers for the Nusselt number and L^2 error are similar. The shown Nusselt number is calculated by averaging over time and space.

5.2.2 Duration of Simulation

We determined the duration of a simulation based on a fixed number of advective times. There are three main timescales for Rayleigh-Bénard flow which can be found from dimensional arguments; the thermal diffusive timescale, thermal advective timescale, and the viscous timescale. Here we ignore the viscous timescale, as we set Pr to 1. In the non-dimensionalisation we have chosen, the diffusive and advective timescales are linked by $\tau_{\text{diffusive}} = \text{Re} \times \tau_{\text{advective}}$. Following Mound and Davies (2017), we run our simulations for a set number (in this case 100) of advective times, after the initial transient has balanced out. However, in the Ra $= 10^5$ case, we restrict the simulation to 1 diffusive time unit, since the solution is effectively steady state.

5.2.3 Choice of Coarse Solver

There are several options for choosing a coarse solver for Parareal. These include a lower order timestepper, a larger timestep, reduced spatial resolution, reduced

Table 5.1: Resolution required to meet various convergence tests. L^2 of the temperature field, $\varepsilon_{\mathrm{Nu}}$, $\varepsilon^{\mathrm{rel}}_{\mathrm{Nu}}$, and $|P-\epsilon_U|/P$ all have tolerance values of 1%. Ra is the Rayleigh number, N_{BL} denotes the resolution required for 6 points to be in the thermal boundary layer, L^2 denotes the defect of the end state temperature field to the high res simulation, $\varepsilon_{\mathrm{Nu}}$ shows $\max(|\mathrm{Nu}_V-\mathrm{Nu}_b|, |\mathrm{Nu}_V-\mathrm{Nu}_t|, |\mathrm{Nu}_b-\mathrm{Nu}_t|)/\mathrm{Nu}_V$, $\varepsilon^{\mathrm{rel}}_{\mathrm{Nu}}$ is the Nusselt number compared with the high resolution simulation, and $|P-\epsilon_U|/P$ is the buoyancy/ dissipation internal consistency check.

Ra	Resolution (N_x, N_z) for error $\leq 1\%$				
	$N_{\mathrm{BL}} \geq 6$	L^2	$\varepsilon_{\mathrm{Nu}}$	$\varepsilon^{\mathrm{rel}}_{\mathrm{Nu}}$	$\lvert P-\epsilon_U\rvert/P$
10^5	(64,32)	(32,16)	(32,16)	(32,16)	(32,16)
10^6	(128,64)	(64,32)	(64,32)	(64,32)	(32,16)
10^7	(128,64)	(-,-)	(64,32)	(64,32)	(64,32)

physics, or a different method of solving the equations. In this work, we reduce the spatial resolution and reduce the timestep. We tested different levels of spatial coarsening to find the optimal amount for speedup. We tested coarsening factors (CF) of 2, 4, and 8, where (N_x, N_z) of the coarse solver is equal to 1/CF (N_x, N_z) of the fine method. A coarsening factor of 2 did not lead to a speedup. Convergence was quick, but the runtime of the coarse solver was too close to the that of the fine solver. A coarsening factor of 4 worked better, allowing for quick convergence along with a significant difference in the cost of the fine/coarse solvers. A factor of 8 reduction showed slow convergence, and was not pursued further.

Coarsening in space requires a method to transmit information from coarse grid to fine grid (interpolation), and back again (restriction). The order of operator for interpolation has been found to be important for the convergence of Parareal (Lunet et al., 2018); a high order method of interpolation helps the convergence of Parareal. In this work we use spectral interpolation, both because of its convergence properties, and because the use of spectral methods for spatial discretisation make it a natural choice.

When choosing a coarse time step, we found situations where a Parareal simulation could be unstable even when a stable coarse solver was combined with a stable fine solver. This is likely due to the stability of Parareal itself, which has its own stability criterion, separate to the individual solvers (see §3.3 or Staff and Rønquist (2005) for further details). This leads to lower speedups as we had to use smaller coarse time steps, making the coarse solver more costly. We also investigated using lower order timesteppers for the coarse solver, along with the reduced resolution. However, as the stability region of Runge-Kutta tends to increase with the order, we found that reduced timestep sizes were required for lower order coarse solvers. This cancelled out any speed increase from reduced computation,

Table 5.2: Spatial resolution (N_x, N_z), timestep size (in diffusion times τ_d), time-serial runtimes (seconds), and simulation duration (in τ_d) for the coarse and fine solvers at different Rayleigh number (Ra).

	Resolution		Timestep		Runtime		Duration
Ra	Coarse	Fine	Coarse	Fine	Coarse	Fine	(τ_d)
10^5	(16,8)	(64,32)	10^{-4}	2×10^{-5}	95.4	300	1.0
10^6	(32,16)	(128,64)	5×10^{-5}	2×10^{-5}	1062	2996	0.6
10^7	(32,16)	(128,64)	5×10^{-6}	2×10^{-6}	5185	14,169	0.3

thus the higher order timestepper RK443 was used in both the fine and coarse solver. Table 5.2 shows the resolutions, timesteps and runtimes of the coarse and fine solvers used in this work.

5.2.4 Determining Convergence in Parareal

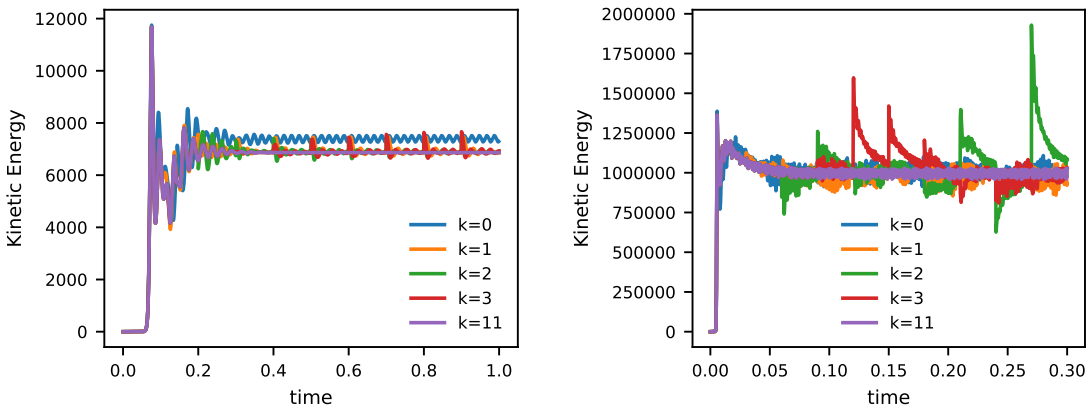
The most simple and widely used check for convergence in Parareal is to monitor the defect between two consecutive iterations (Aubanel, 2011; Berry et al., 2012; Samaddar et al., 2017). This has the benefit of being easy to implement, and can be done whilst running the simulation. However, as discussed in §5.2.1, using the L^2 can lead to substantial over-resolution of the problem if one is interested only in the averaged dynamics. Therefore, the typical online Parareal convergence test is not suitable in this case. Since, at the moment, no termination criteria for averaged dynamics has been published, we perform a fixed number of Parareal iterations and assess convergence in post processing. While useful for benchmarking, this is obviously not a reasonable approach for production runs. Research into alternative and more application-oriented termination criteria for Parareal therefore seems to be an area were further studies are urgently needed.

5.3 Results

5.3.1 Kinetic energy in the Parareal solution

Figures 5.5a, 5.5b show the kinetic energy against time, for Rayleigh numbers 10^5, 10^7, for different numbers of Parareal iterations k. The number of time slices was kept constant at 10. For Ra $= 10^5$, an initial Parareal coarse run shows significant differences from the subsequent Parareal iterations. The overall kinetic energy is higher in the low resolution coarse solver, and varies over time periodically. This

increased kinetic energy in the coarse solver is due to dissipation of the system being under resolved at the coarse resolution. The periodicity is not present in the fine solution, and the effect can be seen to reduce in the subsequent iterations. The kinetic energy quickly reduces to the correct level after the first iteration for each time slice. Subsequent iterations still have a small 'bump' in kinetic energy at the correction time, but the overall level is in accordance with the fine solver. The kinetic energy corrects quickly to the correct level (within tolerance of the fine method) at the start of each time slice, so that the time averaged value falls within tolerance values. The magnitude of the jump is also small, and does not grow significantly beyond the difference between the coarse and fine solvers. The Ra $= 10^7$ case shows problems with the Parareal convergence. The correction steps increase the error, which can be seen in the large jumps at the time slice boundaries. This is the first indication that Parareal has reached the limit of usability in this parameter space. These jumps are of far larger magnitude than those found in the lower Ra case, which is a further reason to suspect that the method is failing for Ra $= 10^7$, whilst accepting that it is working for Ra $= 10^5$.



(a) $Ra = 10^5$ (b) $Ra = 10^7$

Figure 5.5: Dimensionless kinetic energy against time for different numbers of Parareal iterations k for Ra $= 10^5$, 10^7. Time is measured in terms of the diffusion time τ_d, duration was determined as ≈ 100 advective time units after the transient settled. The coarse solver has $\frac{1}{4}$ the number of modes in x and z as the fine solver, the coarse timestep is $\approx 2\times$ the fine timestep, and the simulation used 10 time slices (see table 5.2). The coarse solver for Ra $= 10^5$ shows higher kinetic energy levels, along with periodic behaviour not present in the fine solution, which is proven to be found when $k >$ number of processors ($k = 11$ in this case). For 10^7, large jumps in the solution for $k > 0$ are due to the Parareal correction step. The error at the jumps is growing, rather than shrinking, as the iteration number increases, showing the inability of Parareal to converge in this parameter regime.

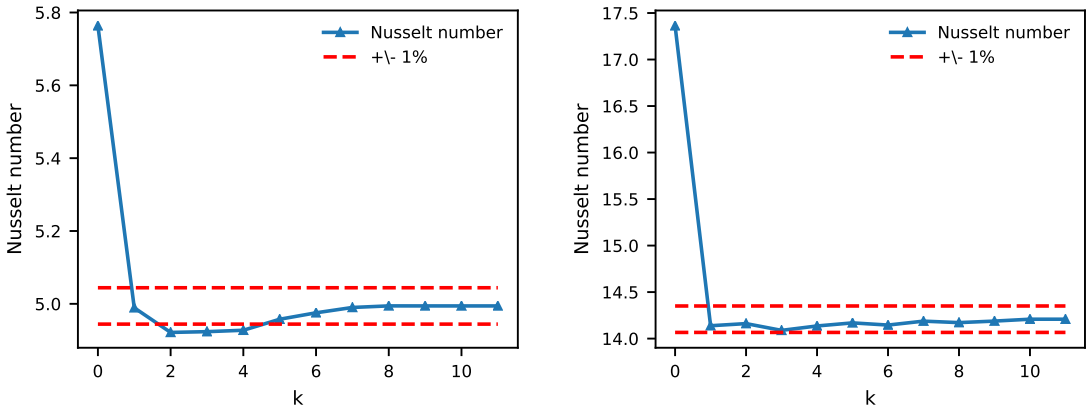
5.3.2 Parareal convergence

Figure 5.6 shows how the calculated Nusselt number changes with increasing Parareal iterations. The Nusselt number found from the initial coarse solve is outside the accuracy requirement with an error of around 10% rather than 1%. In the case of $\mathrm{Ra} = 10^5$, the Nusselt number converges to within the accuracy envelope after 1 iteration, but then in iterations 2-4 it falls back outside this region before converging again from iteration 5. We believe this is due to the well known 'hump' that can be seen for problems with dominant imaginary eigenvalues where the error does not contract monotonically (Gander and Vandewalle, 2007). For $\mathrm{Ra} = 10^7$, the Nusselt number converges after a single iteration in this case of 10 time slices. For different numbers of time slices, the Nusselt number sometimes takes more than one iteration to converge - see figs 5.7 , 5.9.

Figure 5.7 shows the comparison of the L^2 error with the error in Nusselt number for $\mathrm{Ra} = 10^6$, 10^7. In the smaller Ra case, there is smooth convergence in both the L^2 error and in the Nusselt error, although the Nusselt convergence is slightly more erratic. In the $\mathrm{Ra} = 10^7$ case, we see that the Nusselt number error falls just underneath the tolerance threshold after the first iteration. This is followed by a shallow decline in the error until the final iteration. The L^2 error behaves very differently, with a constant error of around 10% right up until the 9^{th} iteration. We see here the mismatch in the error with respect to time averaged quantities with errors with respect to snapshots of the solution (L^2).

Figure 5.8 shows the internal consistency errors ($\varepsilon_{\mathrm{Nu}}$, $|P - \epsilon_U|/P$) for all three Ra tested. In all three cases, the $|P - \epsilon_U|/P$ and $\varepsilon_{\mathrm{Nu}}$ converge to within the 1% tolerance after one iteration. However, the results for $Ra = 10^7$ show that $|P - \epsilon_U|/P$ then returns above the tolerance level, and does not fall reliably until 8 iterations have been completed.

We have also carried out numerical experiments for different numbers of time slices, from 5 to 32 time slices. Here, we would expect to see a trend where the number of iterations required to converge slowly increases with the number of time slices. In our results, we found that the number of iterations required did not behave like this for $\mathrm{Ra} = 10^7$. The number of iterations required increased and decreased with no clear pattern up to 20 time slices. Beyond this the iteration count was always higher than 1, and gradually increased with the number of time slices.

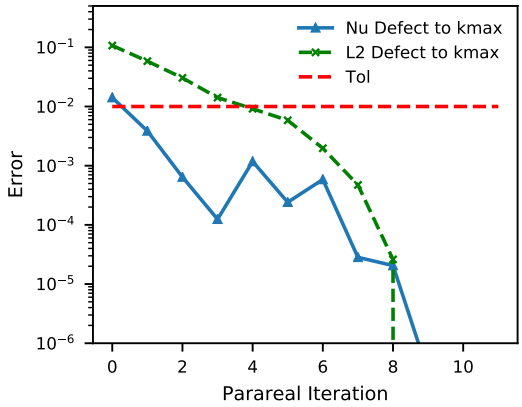


(a) $Ra = 10^5$ (b) $Ra = 10^7$

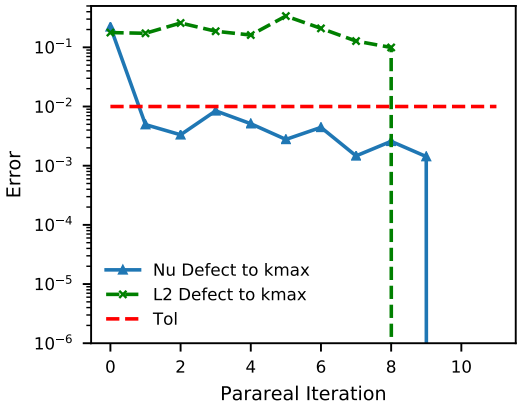
Figure 5.6: Changing Nusselt number $\overline{\mathrm{Nu}}_V$ with Parareal iteration k. There is a large error in the Nusselt number calculated from the coarse solver ($k = 0$), so that at least one iteration is required to calculate the correct Nusselt number (within 1% - dotted red lines). For the Nusselt number alone, convergence behaviour is encouraging, for Ra = 10^5 and Ra = 10^7. The simulation was carried out with 10 time slices.

5.3.3 Scaling and Performance

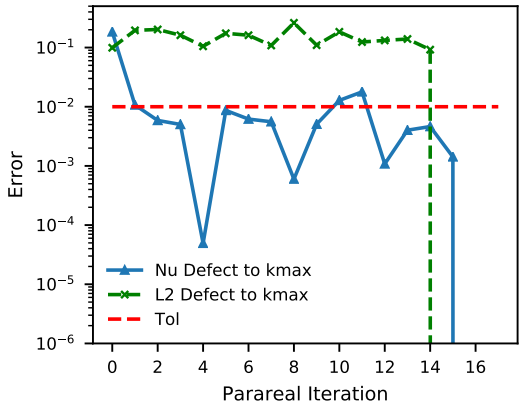
Figure 5.9 shows the scaling performance for simulations with Ra = 10^5, 10^6, 10^7. We see standard scaling behaviour for both 10^5, and 10^6, where speedup increases with processor count until the scaling limit is reached, and no further performance gains are possible. This is due to an increase in the number of Parareal iterations required at higher time slice count. We also see that performance is better at 10^6 than at 10^5, likely because the bigger problem size due to higher resolutions improves scaling. However, the performance of Parareal at Ra = 10^7 is much more mixed. This is in part due to the errors being very close to the tolerance level for all iterations after $k = 1$, see Figure 5.7b. The error does not fall with increasing iterations in the way it does for Ra = 10^5, 10^6, rather, it hovers very close to the tolerance value. Convergence behaviour with number of time slices is unpredictable in this case. For some numbers of time slices, such as in figure 5.7b, the Nusselt error falls below tolerance after one iteration and remains there. In other cases, such as five or 16 time slices, see Figure 5.7c, the error falls below the tolerance and then rises back again.



(a) Ra $= 10^6$

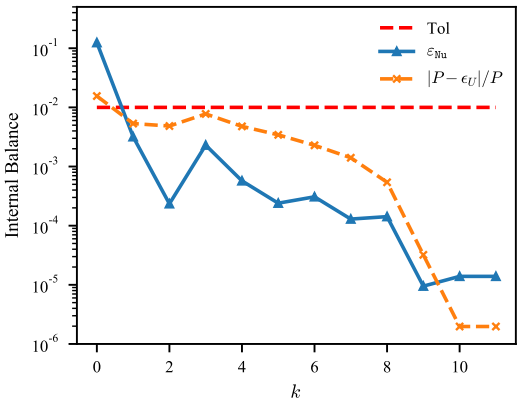


(b) Ra $= 10^7$, 10 Time slices

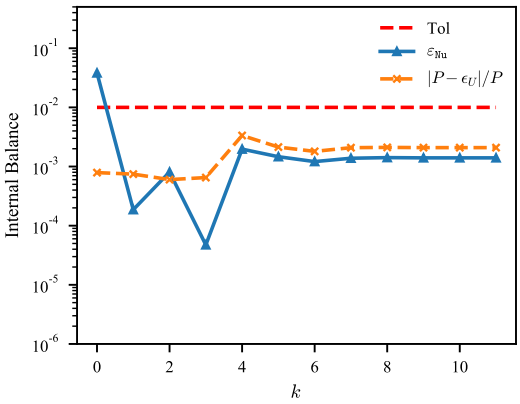


(c) Ra $= 10^7$, 16 Time slices

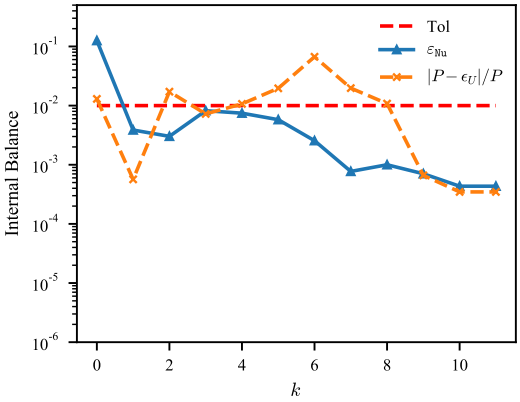
Figure 5.7: Convergence of Nusselt number $\overline{\mathrm{Nu}}_V$ and L^2 error with Parareal iteration for Ra $= 10^6, 10^7$, 10 time slices (a,b), 16 time slices(c). As $k_{\max}$ is greater than number of timeslices, the solution at $k_{\max}$ perfectly represents the serial fine solution. We can see that the L^2 error at Ra $= 10^6$ behaves as expected for good Parareal convergence, with a superlinear convergence behaviour. The Nusselt error at this Ra also shows convergence, but is more erratic. At Ra $= 10^7$, we see much worse convergence. The L^2 error does not converge until the last iteration, when k is equal to the number of time slices. The Nusselt number error behaves slightly better, but does not decrease monotonically. Figure(c) shows Ra $= 10^7$ but with 16 time slices. Here, it requires two iterations for the Nusselt number to reach the 1% tolerance.



(a) Ra $= 10^5$



(b) Ra $= 10^6$



(c) Ra $= 10^7$

Figure 5.8: Convergence of the internal checks carried out on the data of the Parareal simulations, for Ra $= 10^5, 10^6, 10^7$, 10 time slices. The internal energy balance ($|P - \epsilon_U|/P$) takes longer to converge than $\varepsilon_{\mathrm{Nu}}$. The Nusselt number is convergent for all three cases, but the internal energy balance is not convergent at the highest Rayleigh number.

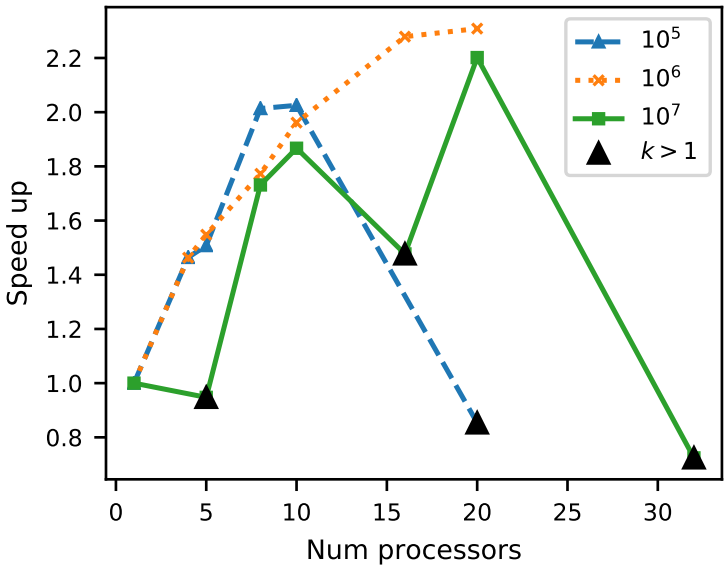


Figure 5.9: Speed up vs number of timeslices/processors for Ra $= 10^5$, 10^6, and 10^7. We can see that performance appears best for Ra $= 10^6$. Peak speedup is around $2 \sim 2.4$ for all Ra. For 10^5, and 10^6, performance is predictable, with speedup increasing with number of cores until a scaling limit is reached. For 10^7, the scaling behaviour is erratic, due to the errors being very close to the tolerance limit. This leads to more iterations being required for convergence at some processor counts, causing the smaller speedups (black triangles).

5.4 Conclusions

In this chapter, we aimed to progress from our work on the linear kinematic dynamo problem to a more complicated physical system, encompassing some of the inherent non-linearity of the full dynamo. To do this, we studied the ability of Parareal to speed up simulations of Rayleigh-Bénard convection. This work has shown that the Parareal algorithm allows for reliable speedup of simulations in a limited range of Rayleigh numbers (up to 10^6) at finite Prandtl number. The algorithm converges quickly with respect to averaged quantities like the Nusselt number and internal energy balance. Although slower, Parareal also converges with respect to the L^2 defect between subsequent iterations. Speedups of up to 2.4 are possible, with around 20 processors, with parallel efficiencies of around 0.2 for Rayleigh numbers as high as 10^6. However, in all cases, speedups were limited to at most 20 processors. Beyond that, increases in the number of required iterations balanced out any gains from using more processors.

At $Ra = 10^7$, we find that convergence of Parareal degrades substantially. The errors in Nu do not fall monotonically with increasing iteration number. For some simulations, the error falls below the tolerance level at a low number of iterations, only to increase in successive iterations. This erratic behaviour leads to irregular scaling performance at 10^7; sometimes the simulation converges in one iteration, sometimes it takes two or three. Parareal is not expected to be useful for simulations of Rayleigh-Bénard convection at Rayleigh numbers above 10^7 as we expect the performance to degrade further as the flow becomes more turbulent, in line with previous results (Steiner et al., 2015). These findings are in contrast to what Samuel (2012) found for $Ra = 10^7$ with infinite Prandtl number, where he observed a small number of iterations independent of the number of time slices being required for convergence and increasing speedup up to 40 processors. Clearly, performance of Parareal is very different in the finite versus infinite Prandtl number case, presumably due to nonlinearity in the momentum equation.

This difference in performance is caused in part by the well known general degradation of Parareal with increasing Reynolds numbers (Steiner et al., 2015). It is also caused by the choice of convergence criteria. The correction step of Parareal depends on pointwise amplitude corrections at the boundary between time slices. In Rayleigh-Bénard convection studies, the particular state of a given field at an instant in time is not of primary concern, therefore we relaxed the accuracy conditions of the fine solution, so that we did not enforce that the L^2 error be below a threshold value. In the cases of Ra $= 10^5$ and 10^6, the L^2 error is of roughly the same magnitude

as the time- and space- averaged quantities ($\varepsilon_{\mathrm{Nu}}$, $|P - \epsilon_U|/P$), used to determine accuracy of the solution. In the 10^7 simulations, we can find a good level of accuracy in the $\varepsilon_{\mathrm{Nu}}$ and $|P - \epsilon_U|/P$, whilst the L^2 error is still high in spatial convergence tests, (see Figure 5.4). As the Parareal algorithm effectively operates on the L^2 error, Parareal convergence is slow.

For the dynamo problem, where high Ra convection is expected (see §1.3.1 and table 1.2), this is not encouraging. One of the main aims of dynamo simulations is to get to more extreme parameters, and increasing the supercriticality of the system ($\mathrm{Ra}/\mathrm{Ra}_C$) would be a significant part of that aim, meaning we would need to simulate Ra much higher than in this work. Parareal is unlikely to be useful in helping dynamo simulations reach more realistic parameter regimes. In the following chapter (Chapter 6), we further discuss this issue, along with the other aims of the thesis.

Chapter 6

Conclusions

In this thesis, we aimed to determine how useful Parareal could be for speeding up simulations of kinematic dynamos and convection. This chapter discusses the key results of this thesis, whilst considering the aims of the thesis from Chapter 1. We then discuss avenues for future work.

6.1 Kinematic dynamos

In Chapter 4, we tested the performance of the Parareal algorithm for the Roberts (1972) and Galloway and Proctor (1992) dynamos. These dynamos are 2.5D, so they can be simulated on a 2D plane, allowing many simulations to be carried out with a reasonable amount of time and computing resources. The Roberts dynamo is generated by a steady flow, whilst the Galloway Proctor dynamo has a time dependent flow. Convergence tests were carried out to determine the spatial and temporal resolutions required to obtain solution accuracies of 10^{-5} in L^2 error (ε), for each of the different Rm investigated (ranging from 3 to 4096). These spatial and temporal resolutions were used as fine solvers in Parareal. Spatial and temporal coarsening were used to create the coarse solvers, with testing carried out to determine that most speed up was generated when spatial coarsening of around a half was used.

Parareal was found to offer parallel speed up for all Rm tested, for both Roberts and Galloway Proctor flows. Parallel in space and time speed ups were higher than those found in purely spatial parallelisation cases. Parareal for the Roberts dynamo offered only modest speed ups, with low parallel efficiency (< 0.1), making it unattractive to use in real world simulations. Convergence in all cases was found in three Parareal iterations, so the theoretical maximum efficiency was 1/3.

The difference between this maximum and the actual performance (parallel efficiency $< 10\%$) is due to the small difference in computational cost between the coarse and fine solvers. At the spatial resolutions used, the largest stable time step of the fine solver was sufficiently accurate to be used as the fine solution. The Galloway Proctor results showed better performance. Parareal efficiency was above 20% for all but the $\mathrm{Rm} = 3$ case, where it was just below 20%. These efficiencies are much closer to the bound of 1/3, indicating an effective implementation. The reason that performance was better in this case was that due to time dependence of the Galloway Proctor flow, smaller timesteps were needed in the fine solver to obtain the desired level of accuracy. This created a larger difference in computational complexity between the fine and coarse solvers than in the Roberts flow. We investigated Rm over 4 orders of magnitude (from 3 to 3000), and found that there was no significant change in Parareal performance as Rm increased.

6.2 Rayleigh-Bénard convection

In Chapter 5 we reported results of applying Parareal to 2D Rayleigh Bénard convection. We used fixed temperature no-slip boundary conditions, and tested performance for $\mathrm{Ra} = 10^5$, 10^6, 10^7. Accuracy of the fine solver was determined by the solution meeting a number of convergence tests: 6 points in the thermal boundary layer, $< 1\%$ error in the balance between viscous dissipation and buoyancy production, and $< 1\%$ difference in Nu calculated at different places in the domain. The coarse solver was created by using solvers with reduced spatial and temporal resolution, with testing determining that a spatial coarsening factor of 4 offered the highest Parareal efficiency. Time step coarsening was limited by Parareal stability conditions as we found cases where a combination of stable fine and stable coarse solver were unstable, as discussed in Staff and Rønquist (2005).

We found that parallel speed up was possible for small numbers of Parareal time slices. The performance was poor though, with a maximum speed up of only 2 for all Ra tested, and a scaling limit reached at 10 - 20 time slices. Above this number of time slices, the number of Parareal iterations required to converge became too large to obtain speed up. Performance of Parareal degraded with increasing Ra. At 10^7, convergence was erratic and the error in Nu did not fall monotonically with increasing iteration number. This change in performance as Ra increases is in part explained by the general degradation of Parareal with increasing Re (Steiner et al., 2015). The choice of convergence criteria was also a factor, as the L^2 errors were of similar magnitude to the errors in Nu and the energy balance at lower Ra (10^5,

10^6). At Ra $= 10^7$, however, the L^2 error was significantly higher than the errors in Nusselt and energy balance, and as the Parareal correction is effectively a point-wise amplitude correction, the large L^2 error caused slow convergence.

6.3 Interpreting our results for the non-linear dynamo problem

We showed that Parareal can offer speed up over and above saturation of spatial scaling for the kinematic dynamo. We also showed that speed up was possible for Rayleigh-Bénard convection at lower Rayleigh numbers. Parareal is not anticipated to be competitive with spatial parallelisation for Rayleigh-Bénard convection, especially at higher Ra. Given that the highest Ra treated in simulations so far is $\sim 10^{15}$ (Iyer et al., 2020), having Parareal struggle at Ra $= 10^7$ would appear to rule out its use in obtaining more extreme parameters. This contrasts with the results in Samuel (2012), who found fair performance at Ra $= 10^7$, but in that case inertia was ignored (due to infinite Prandtl number). As the Prandtl number is ~ 1 in the Earth (Matsui et al., 2016) and $\sim 10^{-7}$ in the Sun, this would not be relevant to dynamo flows.

One reason for differences in performance seen between the kinematic dynamo and RBC is the different accuracy criteria. Accuracy requirements found in the literature for Rayleigh-Bénard convection were found to be less strict than those we imposed when studying the kinematic dynamo problem. Imposing stricter accuracy on the RBC case may have left more scope to find a coarse solver with good Parareal convergence properties whilst being significantly cheaper than the fine solver, but we intended to ensure that our results would be relevant to current practitioners of convection simulations. We also set different tests for Rayleigh-Bénard convection than for the kinematic dynamo, with RBC tests concentrating on statistically averaged values due to the standard tests in the literature, and the kinematic dynamo concentrating on the point-wise L^2 error as no standard tests are found in the literature. For larger Rayleigh numbers in RBC, our tests show a significant disparity between the instantaneous L^2 error in a variable field such as temperature and the error in statistically calculated quantities such as the Nusselt number. In one example, Parareal reached a 1% error with respect to the Nusselt number in 1 iteration while the L^2 error stalled for 7 iterations and only fell below 1% after iteration 8. In a case like Rayleigh-Bénard convection, statistical quantities like the Nusselt number are typically the most informative for understanding

behaviour of the physical system and what domain scientists are interested in. The accuracy levels and convergence criteria set for the kinematic dynamo problem were quite different to those found in the literature for Rayleigh-Bénard convection. More stringent accuracy requirements for a solution is known to increase performance for Parareal (Goetschel et al., 2021). To fully understand this aspect in performance, a study of how Parareal efficiency for the kinematic dynamo changed with different specified accuracy levels would be an interesting avenue for further work.

Another reason for the differences in performance we found is due to the different types of problems being studied. The Galloway Proctor flow is more complicated than the Roberts flow, leading to smaller time steps being required to obtain the desired level of accuracy. The Roberts flow fine solver is accurate with relatively high timestep size (close to the stability limit). This means that the coarse solver timestep is only a little larger (~ 2 times) than the fine timestep. In the Galloway-Proctor flow, the accuracy of the fine solver requires a timestep lower than the stability limit – due to the time dependence of the driving flow. Therefore, the coarse solver, which only has to be stable, can have a timestep ~ 20 times larger than the fine solver in this case. This leaves more potential for speedup. Conversely, the far more complicated behaviour found in convection did not allow for better performance in Parareal. The difference in behaviour of Parareal for different problems studied in this thesis reinforces the view stated in Berry et al. (2012) that it is difficult to gauge performance of Parareal for a particular problem without carrying out some numerical tests.

Whilst standard Parareal offered some positive results in terms of the kinematic dynamo, we believe that the poor performance offered in simulations of RBC mean that it may be difficult to obtain useful performance increases for the full dynamo. Conversely, convection is suppressed by the presence of strong magnetic fields (e.g. Cattaneo et al., 2003; Stein, 2012), so the full dynamo would be less chaotic than pure RBC, possibly making simulations more amenable to good convergence in Parareal.

6.4 Further Work

For the problem of the kinematic dynamo, investigations into different coarse solvers could see improved performance. If a quicker coarse solver, using for example a different spatial discretisation, required a similar number of Parareal iterations, better performance should be possible. Conversely, a coarse solver with similar computa-

tional complexity but higher accuracy could lead to convergence in fewer iterations. It would also be instructive to compare performance of Parareal against different parallel-in-time algorithms. If restricting ourselves to the kinematic dynamo, REXI (Haut et al., 2015; Schreiber et al., 2018) and ParaEXP (Gander and Güttel, 2013) would be very good choices, due to their high suitability for linear problems. It would also be useful to examine the performance of these two methods in advance of any further progress in applying these methods to nonlinear problems. If the work was concentrating on applicability to the full dynamo problem, comparisons between Parareal, MGRIT (Friedhoff et al., 2012), and PFASST (Emmett and Minion, 2012) should be explored. PFASST would be of particular interest, as it removes the $1/k$ bound on parallel efficiency. Since good performance was found in the case of the Galloway Proctor flow, it would be insightful to see if further performance could be obtained by utilising the IPS algorithm (Berry et al., 2012). IPS allows for greater concurrency and less CPU idle time by allocating tasks to processors as the data to perform those tasks becomes available, so that coarse and fine solutions of time slices can be computed in parallel. It may also be useful to see how well this performed for the Roberts dynamo.

Given the poor performance found using standard Parareal with Rayleigh-Bénard convection, investigations into more specialist algorithms is likely the way forward. This will have the major disadvantage of sacrificing non-intrusiveness, so that current established codes would require significant modification in order to investigate the use of different parallel-in-time methods. Given how closely related the methods of Parareal and MGRIT are (Gander and Vandewalle, 2007), it is unlikely that there would be significant difference in performance between them for Rayleigh-Bénard convection problem, especially at very high Rayleigh number. However, this remains an open question, as there have been studies of advective problems where MGRIT has converged for cases where Parareal failed (De Sterck et al., 2021). PFASST offers another promising avenue for study, due to its applicability to nonlinear studies. REXI and ParaEXP are thought to be less likely candidates due them being more suited to linear problems, though the nonlinear modifications of these algorithms would be interesting to study further. Krylov-subspace-enhanced Parareal would be another option to consider, though again this would sacrifice the non-intrusiveness that makes Parareal so appealing.

Further work on Parareal could be carried out with the aim of improving the coarse solver. Recent work by Aubert (2019) showed that hyperdiffusivity in simulations can give comparable results to DNS simulations, when applied to the end of the spectrum, and when applied only to the fluid, whilst leaving the magnetic field fully resolved. They also found that using a statistically steady LES simulation

as the initial condition for a DNS simulation produced only small transients to be generated. This points to a possible model simplification that could be used as the coarse solver. The hyperdiffusivity would reduce the spatial resolution required at a given parameter range, and allow for increased time step size. This could allow for a larger difference in computational complexity between the coarse and fine solvers in Parareal, which was the limiting factor in speed up for the Rayleigh-Bénard convection simulations. This approach would lend itself to combination with the strategy used by Eghbal et al. (2017), whereby Parareal is not used to fully simulate the path from initial conditions to the simulation end time, but as a way of accelerating a simulation to a converged statistically steady state. A coarse run using hyperdiffusivity, followed by some iterations of Parareal using the end state of the coarse solver as an initial condition offers a very interesting avenue for research. As the hyperdiffusivity was applied only to the fluid, Rayleigh-Bénard convection could be a useful testbed for this approach.

For the full dynamo problem, we believe that advancements in Rayleigh-Bénard performance are a prerequisite to obtaining good performance, as this has been identified as a problem area for Parareal. The results found for the kinematic dynamo offer encouragement that if good performance can be found for convective studies, it would be possible to add in the magnetic field without a major effect on performance.

In a recent report on Supercomputing in the UK by Wilkinson et al. (2021), the research goals of the UKRI research community were grouped into seven broad themes. The work in this thesis aligns closely to two of them. The first is 'Expanding the frontiers of fundamental sciences', which includes an aim for better understanding magnetism of the solar system by investigating generation of the magnetic fields in the Sun and planets. The work in this thesis also fits in with the stated aims of 'Mathematics and Science of Computation', more specifically, as this work crosses the domains of specialists in physical applications and supercomputing algorithms, it aligns with 'Mathematics at Scale', which seeks to do just that.

Currently, the most common parallelisation strategies are to spatially decompose a physical domain and, using either shared and/or distributed memory parallelisation, compute each sub domain on a separate processor. Much innovation is currently happening however, and Parallel-in-time algorithms are one novel method that is currently growing in popularity. Other major advances are being made in parallel computing, such as the use of graphics processing units for general purpose computing (GPGPU) and specialist co-processor like hardware such as field programmable gate arrays (FPGAs). On an algorithm level, ensemble averag-

ing and rare event algorithms allow for embarrassingly parallel simulations to utilise the higher number of cores available, but don't allow for increases in numerical resolution. Much research has also been recently carried out in machine learning and Artificial Intelligence for partial differential equations, allowing for parallel computation on GPUs. We believe that parallel-in-time methods are a useful part of the toolset for improving performance on exascale computers, gains have been shown even in areas where they are deemed to have weaknesses. Examples of this include the speed up gains seen in turbulent plasma simulations by Samaddar et al. (2010).

In this work, we focussed on using non-intrusive parallel-in-time algorithms, which led to the choice of standard Parareal. The most important reason for this was that the method could be relatively quickly adopted by researchers in the dynamo area, without major modification to existing codes. However, in focusing on non-intrusive methods, we may have limited the possible gains. More invasive methods could have the potential to push the computational frontiers in dynamo modelling. If a performant parallel-in-time method, possibly a variant of Parareal, could be found for the dynamo problem, it could be of most use in allowing for investigation into long time scale effects such as geomagnetic reversals. It could do this by allowing the very large time domain to be split into chunks and computed in parallel on the new generation of exascale computers. Work on improving Parareal for advective problems could be key to this, such as that by De Sterck et al. (2021).

References

Citing pages are listed after each entry.

Agboh, W., Grainger, O., Ruprecht, D., and Dogar, M. 2020. Parareal with a learned coarse model for robotic manipulation. *Computing and Visualization in Science*. **23**(1), pp. 1–10. Cited on pp. 2 and 29.

Agboh, W. C., Ruprecht, D., and Dogar, M. R. 2019. Combining coarse and fine physics for manipulation using parallel-in-time integration. *arXiv preprint arXiv:1903.08470*. Cited on p. 29.

Ahlers, G., Grossmann, S., and Lohse, D. 2009. Heat transfer and large scale dynamics in turbulent rayleigh-bénard convection. *Reviews of modern physics*. **81**(2), pp. 503. Cited on pp. 3, 17, 19, and 49.

Alexakis, A. 2011. Searching for the fastest dynamo: Laminar abc flows. *Physical Review E*. **84**(2), pp. 026321. Cited on pp. 17 and 39.

Amati, G., Koal, K., Massaioli, F., Sreenivasan, K., and Verzicco, R. 2005. Turbulent thermal convection at high rayleigh numbers for a boussinesq fluid of constant prandtl number. *Physics of Fluids*. **17**(12), pp. 121701. Cited on p. 90.

Amdahl, G. M. Validity of the single processor approach to achieving large scale computing capabilities. In: *Proceedings of the April 18-20, 1967, spring joint computer conference*, pp. 483–485, 1967. Cited on p. 4.

Anders, E. H., Brown, B. P., and Oishi, J. S. 2018. Accelerated evolution of convective simulations. *Physical Review Fluids*. **3**(8), pp. 083502. Cited on pp. 21 and 22.

Archontis, V., Dorch, S. B. F., and Nordlund, Å. 2003. Numerical simulations of kinematic dynamo action. *Astronomy & Astrophysics*. **397**(2), pp. 393–399. Cited on pp. 39 and 71.

Ascher, U. M., Ruuth, S. J., and Wetton, B. T. 1995. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*. **32**(3), pp. 797–823. Cited on p. 61.

Ascher, U. M., Ruuth, S. J., and Spiteri, R. J. 1997. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*. **25**(2-3), pp. 151–167. Cited on pp. 61 and 62.

Atkinson, K., Han, W., and Stewart, D. E. 2011. *Numerical solution of ordinary differential equations.*, volume 108. John Wiley & Sons. Cited on pp. 56, 57, 58, and 59.

Aubanel, E. 2011. Scheduling of tasks in the parareal algorithm. *Parallel Computing*. **37**(3), pp. 172–182. Cited on pp. 8, 9, 26, 67, and 96.

Aubert, J. 2019. Approaching earth's core conditions in high-resolution geodynamo simulations. *Geophysical Journal International*. **219**(Supplement_1), pp. S137–S151. Cited on pp. 13, 15, 33, and 109.

Aubert, J., Gastine, T., and Fournier, A. 2017. Spherical convective dynamos in the rapidly rotating asymptotic regime. *Journal of Fluid Mechanics*. **813**, pp. 558–593. Cited on pp. 14, 15, and 33.

Augustson, K., Brun, A. S., Miesch, M., and Toomre, J. 2015. Grand minima and equatorward propagation in a cycling stellar convective dynamo. *The Astrophysical Journal*. **809**(2), pp. 149. Cited on p. 35.

Aumaître, S., Berhanu, M., Bourgoin, M., et al. 2008. The vks experiment: turbulent dynamical dynamos. *Comptes Rendus Physique*. **9**(7), pp. 689–701. Cited on p. 1.

Bachtiar, A., Ivers, D., and James, R. 2006. Planar velocity dynamos in a sphere. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. **462**(2072), pp. 2439–2456. Cited on p. 37.

Backus, G. 1958. A class of self-sustaining dissipative spherical dynamos. *Annals of Physics*. **4**(4), pp. 372–447. Cited on pp. 10, 37, and 48.

Baffico, L., Bernard, S., Maday, Y., Turinici, G., and Zérah, G. 2002. Parallel-in-time molecular-dynamics simulations. *Physical Review E*. **66**(5), pp. 057701. Cited on pp. 8 and 29.

Bal, G. 2005. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: *Domain decomposition methods in science and engineering*. Springer, pp. 425–432. Cited on p. 8.

Bal, G. and Maday, Y. 2002. A "parareal" time discretization for non-linear pde's with application to the pricing of an american put. In: *Recent developments in domain decomposition methods*. Springer, pp. 189–202. Cited on pp. 2 and 29.

Baudron, A.-M., Lautard, J.-J., Maday, Y., Riahi, M. K., and Salomon, J. 2014. Parareal in time 3d numerical solver for the lwr benchmark neutron diffusion transient model. *Journal of Computational Physics*. **279**, pp. 67–79. Cited on p. 29.

Bayly, B. and Childress, S. 1988. Construction of fast dynamos using unsteady flows and maps in three dimensions. *Geophysical & Astrophysical Fluid Dynamics*. **44** (1-4), pp. 211–240. Cited on p. 39.

Beaudoin, P., Charbonneau, P., Racine, E., and Smolarkiewicz, P. 2013. Torsional oscillations in a global solar dynamo. *Solar Physics*. **282**(2), pp. 335–360. Cited on p. 34.

Benedusi, P., Minion, M., and Krause, R. 2020. An experimental comparison of a space-time multigrid method with pfasst for a reaction-diffusion problem. *arXiv preprint arXiv:2006.12883*. Cited on p. 28.

Berry, L. A., Elwasif, W., Reynolds-Barredo, J. M., et al. 2012. Event-based parareal: A data-flow based implementation of parareal. *Journal of Computational Physics*. **231**(17), pp. 5945–5954. Cited on pp. 8, 26, 96, 108, and 109.

Biggin, A. J., Steinberger, B., Aubert, J., et al. 2012. Possible links between long-term geomagnetic variations and whole-mantle convection processes. *Nature Geoscience*. **5**(8), pp. 526–533. Cited on pp. 1 and 12.

Blouza, A., Boudin, L., and Kaber, S. M. 2011. Parallel in time algorithms with reduction methods for solving chemical kinetics. *Communications in Applied Mathematics and Computational Science*. **5**(2), pp. 241–263. Cited on pp. 8 and 29.

Bodenschatz, E., Pesch, W., and Ahlers, G. 2000. Recent developments in rayleigh-bénard convection. *Annual review of fluid mechanics*. **32**(1), pp. 709–778. Cited on p. 19.

Bolten, M., Friedhoff, S., Hahne, J., and Schöps, S. 2020. Parallel-in-time simulation of an electrical machine using mgrit. *Computing and Visualization in Science*. **23** (1), pp. 1–14. Cited on p. 30.

Boyd, J. P. 2001. *Chebyshev and Fourier spectral methods*. Courier Corporation. Cited on pp. 54 and 59.

Braginsky, S. I. and Roberts, P. H. 1995. Equations governing convection in earth's core and the geodynamo. *Geophysical & Astrophysical Fluid Dynamics*. **79**(1-4), pp. 1–97. Cited on p. 10.

Brandenburg, A. and Subramanian, K. 2005. Astrophysical magnetic fields and nonlinear dynamo theory. *Physics Reports*. **417**(1-4), pp. 1–209. Cited on p. 11.

Brummell, N. H., Cattaneo, F., and Tobias, S. M. 1998. Linear and nonlinear dynamo action. *Physics Letters A*. **249**(5-6), pp. 437–442. Cited on p. 41.

Brun, A. S. and Browning, M. K. 2017. Magnetism, dynamo action and the solar-stellar connection. *Living Reviews in Solar Physics*. **14**(1), pp. 4. Cited on pp. 12, 34, 35, and 53.

Brun, A. S., Strugarek, A., Varela, J., et al. 2017. On differential rotation and overshooting in solar-like stars. *The Astrophysical Journal*. **836**(2), pp. 192. Cited on p. 35.

Bullard, E. C. and Gellman, H. 1954. Homogeneous dynamos and terrestrial magnetism. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*. **247**(928), pp. 213–278. Cited on p. 75.

Burns, K. J., Vasil, G. M., Oishi, J. S., Lecoanet, D., and Brown, B. P. 2020. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*. **2**(2), pp. 023068. Cited on pp. 4 and 62.

Busse, F. 1978. Non-linear properties of thermal convection. *Reports on Progress in Physics*. **41**(12), pp. 1929. Cited on p. 49.

Busse, F. 2002. Convective flows in rapidly rotating spheres and their dynamo action. *Physics of fluids*. **14**(4), pp. 1301–1314. Cited on pp. 2, 17, and 49.

Butcher, J. C. 2008. *Numerical methods for ordinary differential equations.*, volume 2. Wiley Online Library. Cited on pp. 58 and 61.

Buvoli, T. and Minion, M. L. 2020. Imex parareal integrators. *arXiv preprint arXiv:2011.01604*. Cited on p. 8.

Castaing, B., Gunaratne, G., Heslot, F., et al. 1989. Scaling of hard thermal turbulence in rayleigh-bénard convection. *Journal of Fluid Mechanics*. **204**, pp. 1–30. Cited on p. 19.

Cattaneo, F. and Hughes, D. W. 2006. Dynamo action in a rotating convective layer. *Journal of Fluid Mechanics*. **553**, pp. 401. Cited on pp. 10 and 41.

Cattaneo, F. and Hughes, D. W. 2017. Dynamo action in rapidly rotating rayleigh–bénard convection at infinite prandtl number. *Journal of Fluid Mechanics*. **825**, pp. 385–411. Cited on p. 18.

Cattaneo, F. and Tobias, S. 2014. On large-scale dynamo action at high magnetic reynolds number. *The Astrophysical Journal*. **789**(1), pp. 70. Cited on pp. 41 and 42.

Cattaneo, F. and Tobias, S. M. 2009. Dynamo properties of the turbulent velocity field of a saturated dynamo. *Journal of Fluid Mechanics*. **621**, pp. 205. Cited on p. 41.

Cattaneo, F., Emonet, T., and Weiss, N. 2003. On the interaction between convection and magnetic fields. *The Astrophysical Journal*. **588**(2), pp. 1183. Cited on pp. 42 and 108.

Chandrasekhar, S. 1961. *Hydrodynamic and hydromagnetic stability*. Courier Corporation. Cited on pp. 10, 19, 46, 50, 51, 90, and 91.

Chapra, S. C. and Canale, R. P. 2011. *Numerical methods for engineers.*, volume 2. Mcgraw-hill New York. Cited on pp. 56 and 58.

Charbonneau, P. 2020. Dynamo models of the solar cycle. *Living Reviews in Solar Physics*. **17**(1), pp. 1–104. Cited on pp. 2, 15, and 32.

Charbonneau, P. and Steiner, O. 2012. *Solar and Stellar Dynamos: Saas-Fee Advanced Course 39 Swiss Society for Astrophysics and Astronomy*. Saas-Fee Advanced Course. Springer Berlin Heidelberg. Cited on pp. 72 and 75.

Childress, S. 1969. Théorie magnetohydrodynamique de l'effet dynamo. *Report, Department of Mechanics, Faculty of Science, University of Paris*. Cited on pp. 37 and 48.

Childress, S. and Soward, A. 1972. Convection-driven hydromagnetic dynamo. *Physical Review Letters*. **29**(13), pp. 837. Cited on p. 18.

Childress, S., Collet, P., Frisch, U., et al. 1990. Report on workshop on small-diffusivity dynamos and dynamical systems: Observatoire de nice 25–30 june 1989. *Geophysical & Astrophysical Fluid Dynamics*. **52**(4), pp. 263–270. Cited on p. 38.

Christensen, U., Aubert, J., Cardin, P., et al. 2001. A numerical dynamo benchmark. *Physics of the Earth and Planetary Interiors*. **128**(1-4), pp. 25–34. Cited on pp. 14, 17, 32, 33, 34, and 52.

Christensen, U. R. and Aubert, J. 2006. Scaling properties of convection-driven dynamos in rotating spherical shells and application to planetary magnetic fields. *Geophysical Journal International*. **166**(1), pp. 97–114. Cited on pp. 13, 14, and 33.

Christensen, U. R. and Wicht, J. 2015. Numerical dynamo simulations. *Treatise on Geophysics (Second Edition)*. **8**, pp. 245–277. Cited on pp. 10 and 13.

Christlieb, A. J., Macdonald, C. B., and Ong, B. W. 2010. Parallel high-order integrators. *SIAM Journal on Scientific Computing*. **32**(2), pp. 818–835. Cited on p. 28.

Cioni, S., Ciliberto, S., and Sommeria, J. 1997. Strongly turbulent rayleigh–bénard convection in mercury: comparison with results at moderate prandtl number. *Journal of Fluid Mechanics*. **335**, pp. 111–140. Cited on p. 19.

Clarke, A., Davies, C., Ruprecht, D., Tobias, S., and Oishi, J. S. 2020a. Performance of parallel-in-time integration for rayleigh bénard convection. *Computing and visualization in science*. **23**(1), pp. 1–13. Cited on pp. ii and 87.

Clarke, A. T., Davies, C. J., Ruprecht, D., and Tobias, S. M. 2020b. Parallel-in-time integration of kinematic dynamos. *Journal of computational physics: X*. **7**, pp. 100057. Cited on pp. ii and 69.

Clune, T. C., Elliott, J., Miesch, M., Toomre, J., and Glatzmaier, G. A. 1999. Computational aspects of a code to study rotating turbulent convection in spherical shells. *Parallel Computing*. **25**(4), pp. 361–380. Cited on p. 34.

Cooley, J. W. and Tukey, J. W. 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*. **19**(90), pp. 297–301. Cited on p. 53.

Cooper, R. G., Bushby, P., and Guervilly, C. 2020. Subcritical dynamos in rapidly rotating planar convection. *Physical Review Fluids*. **5**(11), pp. 113702. Cited on p. 42.

Cortial, J. and Farhat, C. 2009. A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems. *International Journal for Numerical Methods in Engineering*. **77**(4), pp. 451–470. Cited on p. 2.

Courant, R., Freidrichs, K., and Lewy, H. 1928. Partial differential equations of mathematical physics. *Mathematische Annalen*. **100**, pp. 32–74. Cited on pp. 15 and 59.

Cowling, T. G. 1933. The magnetic field of sunspots. *Monthly Notices of the Royal Astronomical Society*. **94**, pp. 39–48. Cited on pp. 10, 37, and 48.

Croce, R., Ruprecht, D., and Krause, R. 2014. Parallel-in-space-and-time simulation of the three-dimensional, unsteady navier-stokes equations for incompressible flow. In: *Modeling, Simulation and Optimization of Complex Processes-HPSC 2012*. Springer, pp. 13–23. Cited on pp. 2, 4, 8, 9, and 30.

Dai, X. and Maday, Y. 2013. Stable parareal in time method for first-and second-order hyperbolic systems. *SIAM Journal on Scientific Computing*. **35**(1), pp. A52–A78. Cited on p. 26.

Dalcín, L., Paz, R., and Storti, M. 2005. Mpi for python. *Journal of Parallel and Distributed Computing*. **65**(9), pp. 1108–1115. Cited on p. 62.

Dallas, V. and Tobias, S. 2018. Rotationally induced coherence in turbulent kinematic dynamos. *arXiv preprint arXiv:1805.02291*. Cited on pp. 17 and 40.

Davidson, P. A. 2001. *An Introduction to Magnetohydrodynamics*. Cambridge Texts in Applied Mathematics. Cambridge University Press. Cited on pp. 10, 43, 44, 46, and 47.

Davies, C., Pozzo, M., Gubbins, D., and Alfe, D. 2015. Constraints from material properties on the dynamics and evolution of earth's core. *Nature Geoscience*. **8** (9), pp. 678–685. Cited on pp. 1, 13, and 38.

Davies, C. J., Gubbins, D., and Jimack, P. K. 2011. Scalability of pseudospectral methods for geodynamo simulations. *Concurrency and Computation: Practice and Experience*. **23**(1), pp. 38–56. Cited on pp. 2, 12, and 14.

De Sterck, H., Falgout, R. D., Friedhoff, S., Krzysik, O. A., and MacLachlan, S. P. 2021. Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. *Numerical Linear Algebra with Applications*. pp. e2367. Cited on pp. 109 and 111.

Dennard, R. H., Gaensslen, F. H., Yu, H.-N., et al. 1974. Design of ion-implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*. **9**(5), pp. 256–268. Cited on p. 4.

DeRosa, M. L. and Toomre, J. 2004. Evolution of solar supergranulation. *The Astrophysical Journal*. **616**(2), pp. 1242. Cited on p. 12.

Dietrich, W. and Jones, C. 2018. Anelastic spherical dynamos with radially variable electrical conductivity. *Icarus*. **305**, pp. 15–32. Cited on pp. 18, 35, and 36.

Duarte, L. D., Wicht, J., and Gastine, T. 2018. Physical conditions for Jupiter-like dynamo models. *Icarus*. **299**, pp. 206–221. Cited on p. 36.

Dubrulle, B. 2008. Course 5 turbulence and dynamo. *Les Houches*. **88**, pp. 301–358. Cited on p. 13.

Dubrulle, B. 08 2011. Turbulent dynamos. *Proceedings of the International Astronomical Union*. **6**, pp. 326–338. Cited on p. 13.

Eghbal, A., Gerber, A. G., and Aubanel, E. 2017. Acceleration of unsteady hydrodynamic simulations using the parareal algorithm. *Journal of Computational Science*. **19**, pp. 57–76. Cited on pp. 9, 26, 27, 30, and 110.

Emmett, M. and Minion, M. 2012. Toward an efficient parallel in time method for partial differential equations. *Communications in Applied Mathematics and Computational Science*. **7**(1), pp. 105–132. Cited on pp. 27 and 109.

Esmaeilzadeh, H., Blem, E., Amant, R. S., Sankaralingam, K., and Burger, D. Dark silicon and the end of multicore scaling. In: *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, pp. 365–376, 2011. Cited on p. 4.

Falgout, R. D., Katz, A., Kolev, T. V., et al. 2015. Parallel time integration with multigrid reduction for a compressible fluid dynamics application. *Lawrence Livermore National Laboratory Technical Report, LLNL-JRNL-663416*. Cited on p. 30.

Farhat, C., Cortial, J., Dastillung, C., and Bavestrello, H. 2006. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. *International journal for numerical methods in engineering*. **67**(5), pp. 697–724. Cited on p. 25.

Featherstone, N. A. and Hindman, B. W. 2016. The spectral amplitude of stellar convection and its scaling in the high-rayleigh-number regime. *The Astrophysical Journal*. **818**(1), pp. 32. Cited on p. 35.

Fischer, P. F., Hecht, F., and Maday, Y. 2005. A parareal in time semi-implicit approximation of the navier-stokes equations. In: *Domain decomposition methods in science and engineering*. Springer, pp. 433–440. Cited on pp. 2, 9, and 30.

Friedhoff, S., Falgout, R. D., Kolev, T., MacLachlan, S., and Schroder, J. B. 2012. *A multigrid-in-time algorithm for solving evolution equations in parallel*. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States). Cited on pp. 27 and 109.

Friedhoff, S., Falgout, R. D., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. A Multigrid-in-Time Algorithm for Solving Evolution Equations in Parallel. In: *Presented at: Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States, Mar 17 - Mar 22, 2013*, 2013. URL http://www.osti.gov/scitech/servlets/purl/1073108. Cited on p. 2.

Friedhoff, S., Hahne, J., Kulchytska-Ruchka, I., and Schöps, S. 2019. Exploring parallel-in-time approaches for eddy current problems. In: *Progress in Industrial Mathematics at ECMI 2018*. Springer, pp. 373–379. Cited on pp. 9, 22, and 30.

Frigo, M. and Johnson, S. G. 2005. The design and implementation of FFTW3. *Proceedings of the IEEE*. **93**(2), pp. 216–231. Cited on pp. 4 and 62.

Gailitis, A., Lielausis, O., Dement'ev, S., et al. 2000. Detection of a flow induced magnetic field eigenmode in the riga dynamo facility. *Physical Review Letters*. **84** (19), pp. 4365. Cited on pp. 1 and 37.

Galloway, D. 2012. Abc flows then and now. *Geophysical & Astrophysical Fluid Dynamics*. **106**(4-5), pp. 450–467. Cited on pp. 38 and 40.

Galloway, D. J. and Proctor, M. R. 1992. Numerical calculations of fast dynamos in smooth velocity fields with realistic diffusion. *Nature*. **356**(6371), pp. 691–693. Cited on pp. 36, 38, 39, 69, 70, 71, 72, 75, and 105.

Gander, M. and Petcu, M. Analysis of a krylov subspace enhanced parareal algorithm for linear problems. In: *ESAIM: Proceedings*, volume 25, pp. 114–129. EDP Sciences, 2008. Cited on p. 25.

Gander, M. J. 2015. 50 years of time parallel time integration. In: *Multiple shooting and time domain decomposition methods*. Springer, pp. 69–113. Cited on pp. 2, 4, 5, 27, and 29.

Gander, M. J. and Güttel, S. 2013. PARAEXP: A parallel integrator for linear initial-value problems. *SIAM Journal on Scientific Computing*. **35**(2), pp. C123–C142. Cited on pp. 2, 8, 27, and 109.

Gander, M. J. and Vandewalle, S. 2007. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*. **29**(2), pp. 556–578. Cited on pp. 8, 9, 64, 67, 98, and 109.

Gander, M. J., Güttel, S., and Petcu, M. A nonlinear ParaExp algorithm. In: *International Conference on Domain Decomposition Methods*, pp. 261–270. Springer, 2017. Cited on p. 27.

Gander, M. J., Kulchytska-Ruchka, I., Niyonzima, I., and Schöps, S. 2019. A new parareal algorithm for problems with discontinuous sources. *SIAM Journal on Scientific Computing*. **41**(2), pp. B375–B395. Cited on pp. 2 and 26.

Gander, M. J., Kwok, F., and Salomon, J. 2020. Paraopt: A parareal algorithm for optimality systems. *SIAM Journal on Scientific Computing*. **42**(5), pp. A2773–A2802. Cited on p. 29.

Gastine, T. and Wicht, J. 2012. Effects of compressibility on driving zonal flow in gas giants. *Icarus*. **219**(1), pp. 428–442. Cited on p. 34.

Gastine, T., Wicht, J., Duarte, L., Heimpel, M., and Becker, A. 2014. Explaining Jupiter's magnetic field and equatorial jet dynamics. *Geophysical Research Letters*. **41**(15), pp. 5410–5419. Cited on p. 36.

Getling, A. V. 1998. *"Rayleigh-Bénard Convection: Structures and Dynamics"*., volume 11. World Scientific. Cited on p. 49.

Gilbert, A. D. 2003. Dynamo theory. In: *Handbook of mathematical fluid dynamics*. Elsevier, pp. 355–441. Cited on p. 48.

Gilman, P. 1983. Dynamically consistent nonlinear dynamos driven by convection in a rotating spherical shell. II-dynamos with cycles and strong feedbacks. *The Astrophysical Journal Supplement Series*. **53**, pp. 243–268. Cited on p. 35.

Gilman, P. A. 1975. Linear simulations of boussinesq convection in a deep rotating spherical shell. *Journal of Atmospheric Sciences*. **32**(7), pp. 1331–1352. Cited on p. 34.

Gilman, P. A. and Miller, J. 1981. Dynamically consistent nonlinear dynamos driven by convection in a rotating spherical shell. *The Astrophysical Journal Supplement Series*. **46**, pp. 211–238. Cited on p. 34.

Glatzmaier, G. A. 1984. Numerical simulations of stellar convective dynamos. I. the model and method. *Journal of Computational Physics*. **55**(3), pp. 461–484. Cited on p. 34.

Glatzmaier, G. A. 1985. Numerical simulations of stellar convective dynamos. II-field propagation in the convection zone. *The Astrophysical Journal*. **291**, pp. 300–307. Cited on pp. 18 and 35.

Glatzmaier, G. A. 2018. Computer simulations of Jupiter's deep internal dynamics help interpret what juno sees. *Proceedings of the National Academy of Sciences*. **115**(27), pp. 6896–6904. Cited on p. 36.

Glatzmaier, G. A. and Roberts, P. H. 1995a. A three-dimensional convective dynamo solution with rotating and finitely conducting inner core and mantle. *Physics of the Earth and Planetary Interiors*. **91**(1-3), pp. 63–75. Cited on pp. 17 and 32.

Glatzmaier, G. A. and Roberts, P. H. 1995b. A three-dimensional self-consistent computer simulation of a geomagnetic field reversal. *Nature*. **377**(6546), pp. 203–209. Cited on pp. 1, 14, and 32.

Glatzmaier, G. A. and Roberts, P. H. 1998. Dynamo theory then and now. *International Journal of Engineering Science*. **36**(12-14), pp. 1325–1338. Cited on pp. 10 and 11.

Goetschel, S., Minion, M., Ruprecht, D., and Speck, R. 2021. Twelve ways to fool the masses when giving parallel-in-time results. *arXiv preprint arXiv:2102.11670*. Cited on pp. 28, 67, and 108.

Gómez, D. O., Mininni, P. D., and Dmitruk, P. 2005. Parallel simulations in turbulent MHD. *Physica Scripta*. **2005**(T116), pp. 123. Cited on pp. 1 and 17.

Götschel, S. and Minion, M. L. Parallel-in-time for parabolic optimal control problems using PFASST. In: Bjørstad, P. E., Brenner, S. C., Halpern, L., et al. eds. *Domain Decomposition Methods in Science and Engineering XXIV*, pp. 363–371, Cham, 2018. Springer International Publishing. ISBN 978-3-319-93873-8. Cited on p. 29.

Grooms, I. and Julien, K. 2011. Linearly implicit methods for nonlinear PDEs with linear dispersion and dissipation. *Journal of Computational Physics*. **230**(9), pp. 3630–3650. Cited on p. 60.

Grossmann, S. and Lohse, D. 2000. Scaling in thermal convection: a unifying theory. *Journal of Fluid Mechanics*. **407**, pp. 27–56. Cited on pp. 19, 20, 50, 52, and 90.

Gubbins, D. 2008. Implication of kinematic dynamo studies for the geodynamo. *Geophysical Journal International*. **173**(1), pp. 79–91. Cited on p. 10.

Gubbins, D. and Roberts, P. 1987. Magnetohydrodynamics of the earth's core. *Geomatik*. **2**, pp. 1–183. Cited on p. 45.

Guerrero, G., Zaire, B., Smolarkiewicz, P., et al. 2019. What sets the magnetic field strength and cycle period in solar-type stars? *The Astrophysical Journal*. **880** (1), pp. 6. Cited on pp. 35 and 53.

Guervilly, C., Cardin, P., and Schaeffer, N. 2019. Turbulent convective length scale in planetary cores. *Nature*. **570**(7761), pp. 368–371. Cited on p. 33.

Günther, S., Gauger, N. R., and Schroder, J. B. 2018. A non-intrusive parallel-in-time adjoint solver with the XBraid library. *Computing and Visualization in Science*. **19**(3), pp. 85–95. Cited on p. 29.

Gunther, S., Ruthotto, L., Schroder, J. B., Cyr, E. C., and Gauger, N. R. 2020. Layer-parallel training of deep residual neural networks. *SIAM Journal on Mathematics of Data Science*. **2**(1), pp. 1–23. Cited on p. 29.

Gurrala, G., Dimitrovski, A., Pannala, S., Simunovic, S., and Starke, M. 2015. Parareal in time for fast power system dynamic simulations. *IEEE Transactions on Power Systems*. **31**(3), pp. 1820–1830. Cited on p. 30.

Hamon, F. P., Schreiber, M., and Minion, M. L. 2020. Parallel-in-time multi-level integration of the shallow-water equations on the rotating sphere. *Journal of Computational Physics*. **407**, pp. 109210. Cited on p. 31.

Hathaway, D. H. 2015. The solar cycle. *Living reviews in solar physics*. **12**(1), pp. 4. Cited on p. 12.

Haut, T. and Wingate, B. 2014. An asymptotic parallel-in-time method for highly oscillatory PDEs. *SIAM Journal on Scientific Computing*. **36**(2), pp. A693–A713. Cited on p. 28.

Haut, T. S., Babb, T., Martinsson, P. G., and Wingate, B. A. 06 2015. A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator. *IMA Journal of Numerical Analysis*. **36**(2), pp. 688–716. Cited on pp. 28 and 109.

Heimpel, M., Gastine, T., and Wicht, J. 2016. Simulation of deep-seated zonal jets and shallow vortices in gas giant atmospheres. *Nature Geoscience*. **9**(1), pp. 19–23. Cited on p. 36.

Herzenberg, A. 1958. Geomagnetic dynamos. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*. **250** (986), pp. 543–583. Cited on p. 37.

Hessenthaler, A., Nordsletten, D., Röhrle, O., Schroder, J. B., and Falgout, R. D. 2018. Convergence of the multigrid reduction in time algorithm for the linear elasticity equations. *Numerical Linear Algebra with Applications*. **25**(3), pp. e2155. Cited on p. 29.

Hollerbach, R., Galloway, D., and Proctor, M. 1995. Numerical evidence of fast dynamo action in a spherical shell. *Physical review letters*. **74**(16), pp. 3145. Cited on p. 39.

Hurlburt, N. E., Toomre, J., and Massaguer, J. M. 1984. Two-dimensional compressible convection extending over multiple scale heights. *The Astrophysical Journal*. **282**, pp. 557–573. Cited on p. 21.

Hurlburt, N. E., Toomre, J., and Massaguer, J. M. 1986. Nonlinear compressible convection penetrating into stable layers and producing internal gravity waves. *The Astrophysical Journal*. **311**, pp. 563–577. Cited on p. 21.

Iyer, K. P., Scheel, J. D., Schumacher, J., and Sreenivasan, K. R. 2020. Classical 1/3 scaling of convection holds up to Ra=10^{15}. *Proceedings of the National Academy of Sciences*. **117**(14), pp. 7594–7598. Cited on pp. 13, 19, 20, and 107.

Jackson, A., Sheyko, A., Marti, P., et al. 2014. A spherical shell numerical dynamo benchmark with pseudo-vacuum magnetic boundary conditions. *Geophysical Journal International*. **196**(2), pp. 712–723. Cited on pp. 33 and 52.

Johnsson, L. and Netzer, G. The impact of moore's law and loss of dennard scaling: Are dsp socs an energy efficient alternative to x86 socs? In: *Journal of Physics: Conference Series*, volume 762, pp. 012022. IOP Publishing, 2016. Cited on p. 4.

Johnston, H. and Doering, C. R. 2009. Comparison of turbulent thermal convection between conditions of constant temperature and constant flux. *Physical review letters*. **102**(6), pp. 064501. Cited on pp. 20, 21, 90, and 93.

Jones, C. 2007. 8.05 - thermal and compositional convection in the outer core. In: Schubert, G. ed. *Treatise on Geophysics*. Amsterdam: Elsevier, pp. 131–185. Cited on p. 13.

Jones, C. 2014. A dynamo model of Jupiter's magnetic field. *Icarus*. **241**, pp. 148–159. Cited on p. 36.

Jones, C., Boronski, P., Brun, A., et al. 2011. Anelastic convection-driven dynamo benchmarks. *Icarus*. **216**(1), pp. 120–135. Cited on pp. 2, 17, 34, and 45.

Jones, C. A. 2008. Course 2 dynamo theory. In: Cardin, P. and Cugliandolo, L. eds. *Dynamos*. Elsevier, pp. 45–135. Cited on pp. 13, 16, 17, 37, 39, 49, 72, and 74.

Jones, C. A. 2011. Planetary magnetic fields and fluid dynamos. *Annual Review of Fluid Mechanics*. **43**, pp. 583–614. Cited on pp. 10, 32, and 49.

Jones, C. A. and Roberts, P. H. 2000. Convection-driven dynamos in a rotating plane layer. *Journal of Fluid Mechanics*. **404**, pp. 311–343. Cited on pp. 13, 16, and 18.

Jones, C. A., Thompson, M. J., and Tobias, S. M. 2010. The solar dynamo. *Space Science Reviews*. **152**(1-4), pp. 591–616. Cited on p. 13.

Jones, S. E. and Gilbert, A. D. 2014. Dynamo action in the abc flows using symmetries. *Geophysical & Astrophysical Fluid Dynamics*. **108**(1), pp. 83–116. Cited on p. 39.

Kageyama, A. and Sato, T. 2004. "yin-yang grid": An overset grid in spherical geometry. *Geochemistry, Geophysics, Geosystems*. **5**(9). Cited on pp. 34 and 53.

Kageyama, A., Miyagoshi, T., and Sato, T. 2008. Formation of current coils in geodynamo simulations. *Nature*. **454**(7208), pp. 1106–1109. Cited on pp. 14 and 33.

Käpylä, P., Mantere, M., Guerrero, G., Brandenburg, A., and Chatterjee, P. 2011. Reynolds stress and heat flux in spherical shell convection. *Astronomy & Astrophysics*. **531**, pp. A162. Cited on pp. 34 and 53.

Käpylä, P. J., Mantere, M. J., and Brandenburg, A. 2012. Cyclic magnetic activity due to turbulent convection in spherical wedge geometry. *The Astrophysical Journal Letters*. **755**(1), pp. L22. Cited on pp. 18, 34, and 35.

Käpylä, P. J., Mantere, M. J., Cole, E., Warnecke, J., and Brandenburg, A. 2013. Effects of enhanced stratification on equatorward dynamo wave propagation. *The Astrophysical Journal*. **778**(1), pp. 41. Cited on pp. 34 and 35.

Kassam, A.-K. and Trefethen, L. N. 2005. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*. **26**(4), pp. 1214–1233. Cited on p. 62.

Kennedy, C. A. and Carpenter, M. H. 2003. Additive runge–kutta schemes for convection–diffusion–reaction equations. *Applied numerical mathematics*. **44**(1-2), pp. 139–181. Cited on p. 62.

Kerr, R. M. and Herring, J. R. 2000. Prandtl number dependence of nusselt number in direct numerical simulations. *Journal of Fluid Mechanics*. **419**, pp. 325–344. Cited on p. 52.

King, E., Stellmach, S., and Buffett, B. 2013. Scaling behaviour in rayleigh–bénard convection with and without rotation. *Journal of Fluid Mechanics*. **717**, pp. 449–471. Cited on pp. 21, 90, and 91.

King, E. M. and Buffett, B. A. 2013. Flow speeds and length scales in geodynamo models: the role of viscosity. *Earth and Planetary Science Letters*. **371**, pp. 156–162. Cited on p. 13.

King, E. M., Stellmach, S., and Aurnou, J. M. 2012. Heat transfer by rapidly rotating rayleigh–bénard convection. *Journal of Fluid Mechanics*. **691**, pp. 568–582. Cited on pp. 52 and 89.

Kono, M. and Roberts, P. H. 2002. Recent geodynamo simulations and observations of the geomagnetic field. *Reviews of Geophysics*. **40**(4), pp. 4–1. Cited on p. 46.

Kooij, G. L. 2017. *Towards parallel-in-time simulations of turbulent Rayleigh-Bénard convection*. University of Twente. Cited on p. 21.

Kooij, G. L., Botchev, M., and Geurts, B. J. 2015. Direct numerical simulation of nusselt number scaling in rotating rayleigh–bénard convection. *International journal of heat and fluid flow*. **55**, pp. 26–33. Cited on pp. 21 and 22.

Kooij, G. L., Botchev, M. A., and Geurts, B. J. 2018. An exponential time integrator for the incompressible navier–stokes equation. *SIAM journal on scientific computing*. **40**(3), pp. B684–B705. Cited on p. 28.

Kreienbuehl, A., Naegel, A., Ruprecht, D., et al. 2015. Numerical simulation of skin transport using parareal. *Computing and visualization in science*. **17**(2), pp. 99–108. Cited on pp. 9 and 29.

Kreienbuehl, A., Benedusi, P., Ruprecht, D., and Krause, R. 2017. Time-parallel gravitational collapse simulation. *Communications in Applied Mathematics and Computational Science*. **12**(1), pp. 109–128. Cited on p. 29.

Kundu, P. K., Cohen, I. M., and Dowling, D. R. 2016. Chapter 4 - conservation laws. In: Kundu, P. K., Cohen, I. M., and Dowling, D. R. eds. *Fluid Mechanics (Sixth Edition)*. sixth edition ed. Boston: Academic Press, pp. 109–193. Cited on pp. 45, 46, and 50.

Kupka, F. and Muthsam, H. J. 2017. Modelling of stellar convection. *Living Reviews in Computational Astrophysics*. **3**(1), pp. 1–159. Cited on p. 12.

Larmor, J. 1919. How could a rotating body such as the sun become a magnet. *Rep. Brit. Adv. Sci.* pp. 159–160. Cited on p. 10.

Li, K., Livermore, P. W., and Jackson, A. 2010. An optimal galerkin scheme to solve the kinematic dynamo eigenvalue problem in a full sphere. *Journal of computational physics*. **229**(23), pp. 8666–8683. Cited on p. 17.

Lions, J.-L., Maday, Y., and Turinici, G. 2001. Résolution d'edp par un schéma en temps «pararéel». *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*. **332**(7), pp. 661–668. Cited on p. 2.

Livermore, P. 2007. An implementation of the exponential time differencing scheme to the magnetohydrodynamic equations in a spherical shell. *Journal of Computational Physics*. **220**(2), pp. 824–838. Cited on pp. 15 and 17.

Livermore, P., Hughes, D., and Tobias, S. 2007a. The role of helicity and stretching in forced kinematic dynamos in a spherical shell. *Physics of Fluids*. **19**(5), pp. 057101. Cited on p. 40.

Livermore, P., Hughes, D., and Tobias, S. 2010. Nonlinear generation of large-scale magnetic fields in forced spherical shell dynamos. *Physics of Fluids*. **22**(3), pp. 037101. Cited on p. 41.

Livermore, P. W. and Jackson, A. 2005. A comparison of numerical schemes to solve the magnetic induction eigenvalue problem in a spherical geometry. *Geophysical and Astrophysical Fluid Dynamics*. **99**(6), pp. 467–480. Cited on p. 17.

Livermore, P. W., Jones, C. A., and Worland, S. J. 2007b. Spectral radial basis functions for full sphere computations. *Journal of Computational Physics*. **227** (2), pp. 1209–1224. Cited on p. 17.

Llewellyn Smith, S. and Tobias, S. 2004. Vortex dynamos. *Journal of Fluid Mechanics*. **498**, pp. 1–21. Cited on p. 39.

Long, R. S. 2020. *Regimes and scaling laws for convection with and without rotation*. PhD thesis, University of Leeds. Cited on p. 19.

Long, R. S., Mound, J. E., Davies, C. J., and Tobias, S. M. 2020. Scaling behaviour in spherical shell rotating convection with fixed-flux thermal boundary conditions. *Journal of Fluid Mechanics*. **889**. Cited on p. 20.

Lunet, T., Bodart, J., Gratton, S., and Vasseur, X. 2018. Time-parallel simulation of the decay of homogeneous turbulence using parareal with spatial coarsening. *Computing and Visualization in Science*. **19**(1-2), pp. 31–44. Cited on pp. 8, 9, 30, and 95.

Maday, Y. and Mula, O. 2020. An adaptive parareal algorithm. *Journal of Computational and Applied Mathematics*. pp. 112915. Cited on pp. 26 and 27.

Maday, Y. and Turinici, G. 2003. Parallel in time algorithms for quantum control: Parareal time discretization scheme. *International journal of quantum chemistry*. **93**(3), pp. 223–228. Cited on pp. 8 and 29.

Maday, Y., Salomon, J., and Turinici, G. 2007. Monotonic parareal control for quantum systems. *SIAM Journal on Numerical Analysis*. **45**(6), pp. 2468–2482. Cited on p. 8.

Maday, Y., Riahi, M.-K., and Salomon, J. 2013. Parareal in time intermediate targets methods for optimal control problems. In: *Control and optimization with PDE constraints*. Springer, pp. 79–92. Cited on p. 29.

Magoulès, F., Gbikpi-Benissan, G., and Zou, Q. 2018. Asynchronous iterations of parareal algorithm for option pricing models. *Mathematics*. **6**(4), pp. 45. Cited on p. 29.

Malkus, W. V. 1954. The heat transport and spectrum of thermal turbulence. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*. **225**(1161), pp. 196–212. Cited on p. 19.

Marti, P. and Jackson, A. 2016. A fully spectral methodology for magnetohydrodynamic calculations in a whole sphere. *Journal of Computational Physics*. **305**, pp. 403–422. Cited on p. 17.

Masada, Y., Yamada, K., and Kageyama, A. 2013. Effects of penetrative convection on solar dynamo. *The Astrophysical Journal*. **778**(1), pp. 11. Cited on p. 34.

Matilsky, L. I. and Toomre, J. 2020. Exploring bistability in the cycles of the solar dynamo through global simulations. *The Astrophysical Journal*. **892**(2), pp. 106. Cited on p. 35.

Matsui, H., Heien, E., Aubert, J., et al. 2016. Performance benchmarks for a next generation numerical dynamo model. *Geochemistry, Geophysics, Geosystems*. **17** (5), pp. 1586–1607. Cited on pp. 1, 2, 4, 12, 13, 14, 15, 17, 33, and 107.

Miesch, M. S., Elliott, J. R., Toomre, J., et al. 2000. Three-dimensional spherical simulations of solar convection. I. differential rotation and pattern evolution achieved with laminar and turbulent states. *The astrophysical journal*. **532**(1), pp. 593. Cited on p. 34.

Minion, M. 2011. A hybrid parareal spectral deferred corrections method. *Communications in Applied Mathematics and Computational Science*. **5**(2), pp. 265–301. Cited on pp. 2 and 67.

Minion, M. L. and Williams, S. A. Parareal and spectral deferred corrections. In: *AIP Conference Proceedings*, volume 1048, pp. 388–391. American Institute of Physics, 2008. Cited on p. 27.

Miyagoshi, T., Kageyama, A., and Sato, T. 2010. Zonal flow formation in the earth's core. *Nature*. **463**(7282), pp. 793–796. Cited on pp. 14 and 33.

Moore, G. E. et al. Cramming more components onto integrated circuits, 1965. Cited on p. 4.

Moore, G. E. et al. Progress in digital integrated electronics. In: *Electron devices meeting*, volume 21, pp. 11–13. Maryland, USA, 1975. Cited on pp. 4 and 52.

Mound, J. E. and Davies, C. J. 2017. Heat transfer in rapidly rotating convection with heterogeneous thermal boundary conditions. *Journal of Fluid Mechanics*. **828**, pp. 601–629. Cited on pp. 52, 89, and 94.

Muller, R., Auffret, H., Roudier, T., et al. 1992. Evolution and advection of solar mesogranulation. *Nature*. **356**(6367), pp. 322–325. Cited on p. 12.

Nelson, N. J. and Miesch, M. S. 2014. Generating buoyant magnetic flux ropes in solar-like convective dynamos. *Plasma Physics and Controlled Fusion*. **56**(6), pp. 064004. Cited on p. 35.

Nelson, N. J., Brown, B. P., Brun, A. S., Miesch, M. S., and Toomre, J. 2013. Magnetic wreaths and cycles in convective dynamos. *The Astrophysical Journal*. **762**(2), pp. 73. Cited on p. 35.

Nielsen, A. S. and Hesthaven, J. S. Fault tolerance in the parareal method. In: *Proceedings of the ACM Workshop on Fault-Tolerance for HPC at Extreme Scale*, pp. 1–8, 2016. Cited on p. 26.

Nielsen, A. S., Brunner, G., and Hesthaven, J. S. 2018. Communication-aware adaptive parareal with application to a nonlinear hyperbolic system of partial differential equations. *Journal of Computational Physics*. **371**, pp. 483–505. Cited on p. 26.

Nievergelt, J. 1964. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*. **7**(12), pp. 731–733. Cited on p. 4.

Nordlund, Å., Stein, R. F., and Asplund, M. 2009. Solar surface convection. *Living Reviews in Solar Physics*. **6**(1), pp. 1–117. Cited on p. 34.

Ong, B. W. and Schroder, J. B. 2020. Applications of time parallelization. *Computing and Visualization in Science*. **23**(1), pp. 1–15. Cited on pp. 28 and 29.

Ossendrijver, M. 2003. The solar dynamo. *The Astronomy and Astrophysics Review*. **11**(4), pp. 287–367. Cited on p. 12.

Otani, N. F. 1993. A fast kinematic dynamo in two-dimensional time-dependent flows. *Journal of Fluid Mechanics*. **253**, pp. 327–340. Cited on p. 39.

Pandey, A., Scheel, J. D., and Schumacher, J. 2018. Turbulent superstructures in rayleigh-bénard convection. *Nature communications*. **9**(1), pp. 1–11. Cited on p. 21.

Pankratius, V., Schulte, W., and Keutzer, K. 2010. Guest editors' introduction: Parallelism on the desktop. *IEEE Software*. **28**(1), pp. 14–16. Cited on p. 4.

Parker, E. N. 1955. Hydromagnetic dynamo models. *The Astrophysical Journal*. **122**, pp. 293. Cited on p. 37.

Passos, D. and Charbonneau, P. 2014. Characteristics of magnetic solar-like cycles in a 3d MHD simulation of solar convection. *Astronomy & Astrophysics*. **568**, pp. A113. Cited on pp. 34, 35, and 53.

Pedlosky, J. et al. 1987. *Geophysical fluid dynamics.*, volume 710. Springer. Cited on p. 45.

Plumley, M. and Julien, K. 2019. Scaling laws in rayleigh-benard convection. *Earth and Space Science*. **6**(9), pp. 1580–1592. Cited on p. 19.

Plunian, F. and Rädler, K. 2002. Harmonic and subharmonic solutions of the roberts dynamo problem. application to the karlsruhe experiment. *Magnetohydrodynamics*. **38**(1-2), pp. 92–103. Cited on pp. 37, 72, and 74.

Ponomarenko, Y. B. 1973. Theory of the hydromagnetic generator. *Journal of Applied Mechanics and Technical Physics*. **14**(6), pp. 775–778. Cited on p. 37.

Priest, E. R. 1982. *Solar magnetohydrodynamics.*, volume 21. Springer Science & Business Media. Cited on p. 12.

Racine, E., Charbonneau, P., Ghizaru, M., Bouchat, A., and Smolarkiewicz, P. K. 2011. On the mode of dynamo action in a global large-eddy simulation of solar convection. *The Astrophysical Journal*. **735**(1), pp. 46. Cited on pp. 34 and 35.

Reyl, C., Antonsen Jr, T. M., and Ott, E. 1996. Quasi-two-dimensional fast kinematic dynamo instabilities of chaotic fluid flows. *Physics of Plasmas*. **3**(7), pp. 2564–2578. Cited on p. 39.

Reynolds-Barredo, J. M., Newman, D. E., Sánchez, R., et al. 2012. Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations. *Journal of Computational Physics*. **231**(23), pp. 7851–7867. Cited on p. 31.

Rieutord, M. 2014. *Fluid Dynamics: An Introduction*. Graduate Texts in Physics. Springer International Publishing. Cited on p. 52.

Rincon, F. 2019. Dynamo theories. *Journal of Plasma Physics*. **85**(4). Cited on pp. 1, 2, 10, 16, 32, 37, 39, and 49.

Roberts, G. O. 1972. Dynamo action of fluid motions with two-dimensional periodicity. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*. **271**(1216), pp. 411–454. Cited on pp. 36, 37, 39, 69, 70, 71, 72, 74, and 105.

Roberts, P. H. 1967. *An introduction to magnetohydrodynamics.*, volume 6. Longmans London. Cited on p. 16.

Roberts, P. H. and Soward, A. M. 1992. Dynamo theory. *Annual review of fluid mechanics*. **24**(1), pp. 459–512. Cited on pp. 1 and 38.

Rupp, K. 42 years of microprocessor trend data. In: *GitHub*, 2018. URL https://github.com/karlrupp/microprocessor-trend-data/. Cited on p. 5.

Ruprecht, D. 2014. Convergence of parareal with spatial coarsening. *PAMM*. **14** (1), pp. 1031–1034. Cited on p. 8.

Ruprecht, D. and Krause, R. 2012. Explicit parallel-in-time integration of a linear acoustic-advection system. *Computers & Fluids*. **59**, pp. 72–83. Cited on p. 26.

Ruprecht, D., Speck, R., Emmett, M., Bolten, M., and Krause, R. Poster: Extreme-scale space-time parallelism. In: *Proceedings of the 2013 Conference on High Performance Computing Networking, Storage and Analysis Companion*, SC '13 Companion, 2013. URL http://sc13.supercomputing.org/sites/default/files/PostersArchive/tech_posters/post148s2-file3.pdf. Cited on p. 27.

Sakuraba, A. and Roberts, P. H. 2009. Generation of a strong magnetic field using uniform heat flux at the surface of the core. *Nature Geoscience*. **2**(11), pp. 802–805. Cited on p. 14.

Samaddar, D., Newman, D. E., and Sánchez, R. 2010. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *Journal of Computational Physics*. **229**(18), pp. 6558–6573. Cited on pp. 2, 9, 31, and 111.

Samaddar, D., Coster, D., Bonnin, X., et al. 2017. Temporal parallelization of edge plasma simulations using the parareal algorithm and the solps code. *Computer Physics Communications*. **221**, pp. 19–27. Cited on pp. 5, 9, 31, and 96.

Samuel, H. 2012. Time domain parallelization for computational geodynamics. *Geochemistry, Geophysics, Geosystems*. **13**(1). Cited on pp. 2, 9, 31, 103, and 107.

Schaeffer, N., Jault, D., Nataf, H.-C., and Fournier, A. 2017. Turbulent geodynamo simulations: a leap towards earth's core. *Geophysical Journal International*. **211** (1), pp. 1–29. Cited on pp. 1, 2, 13, 14, 15, and 33.

Schmalzl, J., Breuer, M., and Hansen, U. 2004. On the validity of two-dimensional numerical approaches to time-dependent thermal convection. *EPL (Europhysics Letters)*. **67**(3), pp. 390. Cited on p. 21.

Schöbel, R. and Speck, R. 2020. Pfasst-er: Combining the parallel full approximation scheme in space and time with parallelization across the method. *Computing and Visualization in Science*. **23**(1), pp. 1–12. Cited on p. 28.

Schreiber, M. and Loft, R. 2019. A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. *Numerical Linear Algebra with Applications*. **26**(2), pp. e2220. Cited on p. 31.

Schreiber, M., Peixoto, P. S., Haut, T., and Wingate, B. 2018. Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. *The International Journal of High Performance Computing Applications*. **32**(6), pp. 913–933. Cited on pp. 4, 5, 28, 31, and 109.

Schreiber, M., Schaeffer, N., and Loft, R. 2019. Exponential integrators with parallel-in-time rational approximations for the shallow-water equations on the rotating sphere. *Parallel Computing*. **85**, pp. 56–65. Cited on p. 31.

Schroder, J. B. 2017. Parallelizing over artificial neural network training runs with multigrid. *arXiv preprint arXiv:1708.02276*. Cited on p. 29.

Seshasayanan, K. and Alexakis, A. 2016. Turbulent 2.5-dimensional dynamos. *Journal of Fluid Mechanics*. **799**, pp. 246–264. Cited on p. 39.

Seshasayanan, K., Dallas, V., and Alexakis, A. 2017. The onset of turbulent rotating dynamos at the low magnetic prandtl number limit. *Journal of Fluid Mechanics*. **822**(R3). Cited on p. 40.

Shishkina, O., Stevens, R. J., Grossmann, S., and Lohse, D. 2010. Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution. *New journal of Physics*. **12**(7), pp. 075022. Cited on p. 90.

Shraiman, B. I. and Siggia, E. D. 1990. Heat transport in high-rayleigh-number convection. *Physical Review A*. **42**(6), pp. 3650. Cited on p. 19.

Siggia, E. D. 1994. High rayleigh number convection. *Annual review of fluid mechanics*. **26**(1), pp. 137–168. Cited on p. 52.

Soderlund, K. and Stanley, S. 2020. The underexplored frontier of ice giant dynamos. *Philosophical Transactions of the Royal Society A*. **378**(2187), pp. 20190479. Cited on p. 36.

Soderlund, K., Heimpel, M., King, E., and Aurnou, J. 2013. Turbulent models of ice giant internal dynamics: Dynamos, heat transfer, and zonal flows. *Icarus*. **224** (1), pp. 97–113. Cited on p. 36.

Sommeijer, B. P. 1993. Parallel-iterated runge-kutta methods for stiff ordinary differential equations. *Journal of Computational and Applied Mathematics*. **45** (1-2), pp. 151–168. Cited on p. 28.

Spalart, P. R., Moser, R. D., and Rogers, M. M. 1991. Spectral methods for the navier-stokes equations with one infinite and two periodic directions. *Journal of Computational Physics*. **96**(2), pp. 297–324. Cited on p. 62.

Speck, R., Ruprecht, D., Krause, R., et al. A massively space-time parallel n-body solver. In: *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–11. IEEE, 2012. Cited on pp. 5, 27, and 30.

Speck, R., Ruprecht, D., Krause, R., et al. 2014. Integrating an n-body problem with sdc and pfasst. In: *Domain Decomposition Methods in Science and Engineering XXI*. Springer, pp. 637–645. Cited on p. 30.

Sreenivasan, B. and Jones, C. A. 2006. The role of inertia in the evolution of spherical dynamos. *Geophysical Journal International*. **164**(2), pp. 467–476. Cited on pp. 14 and 32.

Staff, G. A. and Rønquist, E. M. 2005. Stability of the parareal algorithm. In: *Domain decomposition methods in science and engineering*. Springer, pp. 449–456. Cited on pp. 95 and 106.

Stanley, S. and Bloxham, J. 2004. Convective-region geometry as the cause of uranus' and neptune's unusual magnetic fields. *Nature*. **428**(6979), pp. 151–153. Cited on p. 36.

Stanley, S. and Bloxham, J. 2006. Numerical dynamo models of uranus' and neptune's magnetic fields. *Icarus*. **184**(2), pp. 556–572. Cited on p. 36.

Stein, R. F. 2012. Solar surface magneto-convection. *Living Reviews in Solar Physics*. **9**(1), pp. 1–51. Cited on pp. 42 and 108.

Steiner, J., Ruprecht, D., Speck, R., and Krause, R. 2015. Convergence of parareal for the navier-stokes equations depending on the reynolds number. In: *Numerical Mathematics and Advanced Applications-ENUMATH 2013*. Springer, pp. 195–202. Cited on pp. 8, 9, 30, 103, and 106.

Stellmach, S. and Hansen, U. 2004. Cartesian convection driven dynamos at low ekman number. *Physical Review E*. **70**(5), pp. 056312. Cited on pp. 18 and 42.

Stevens, R. J., Verzicco, R., and Lohse, D. 2010. Radial boundary layer structure and nusselt number in rayleigh–bénard convection. *Journal of fluid mechanics*. **643**, pp. 495–507. Cited on pp. 52, 89, and 90.

Stevens, R. J., Blass, A., Zhu, X., Verzicco, R., and Lohse, D. 2018. Turbulent thermal superstructures in rayleigh-bénard convection. *Physical review fluids*. **3** (4), pp. 041501. Cited on p. 21.

Stieglitz, R. and Müller, U. 2001. Experimental demonstration of a homogeneous two-scale dynamo. *Physics of Fluids*. **13**(3), pp. 561–564. Cited on pp. 1, 37, and 42.

Strugarek, A., Beaudoin, P., Charbonneau, P., Brun, A., and Do Nascimento, J.-D. 2017. Reconciling solar and stellar magnetic cycles with nonlinear dynamo simulations. *Science*. **357**(6347), pp. 185–187. Cited on p. 34.

Tilgner, A. and Brandenburg, A. 2008. A growing dynamo from a saturated roberts flow dynamo. *Monthly Notices of the Royal Astronomical Society*. **391**(3), pp. 1477–1481. Cited on p. 41.

Tilgner, A. and Busse, F. 1995. Subharmonic dynamo action of fluid motions with two-dimensional periodicity. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*. **448**(1933), pp. 237–244. Cited on p. 37.

Tobias, S. 2021. The turbulent dynamo. *Journal of Fluid Mechanics*. **912**. Cited on pp. 10, 11, 12, 15, 16, 32, 39, 49, and 75.

Tobias, S. M. and Cattaneo, F. 2013. Shear-driven dynamo waves at high magnetic reynolds number. *Nature*. **497**(7450), pp. 463–465. Cited on pp. 1 and 41.

Trefethen, L. 2000. *Spectral Methods in MATLAB*. Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104). Cited on pp. 54 and 55.

Tritton, D. J. 1977. *Physical fluid dynamics*. Springer Science & Business Media. Cited on p. 50.

Väisälä, M. S., Pekkilä, J., Käpylä, M. J., et al. 2021. Interaction of large-and small-scale dynamos in isotropic turbulent flows from gpu-accelerated simulations. *The Astrophysical Journal*. **907**(2), pp. 83. Cited on p. 17.

Van der Houwen, P. and Sommeijer, B. 1993. Analysis of parallel diagonally implicit iteration of runge-kutta methods. *Applied numerical mathematics*. **11**(1-3), pp. 169–188. Cited on p. 28.

Van Der Houwen, P. J. and Sommeijer, B. P. 1990. Parallel iteration of high-order runge-kutta methods with stepsize control. *Journal of Computational and Applied Mathematics*. **29**(1), pp. 111–127. Cited on p. 28.

van der Houwen, P. J. and Sommeijer, B. P. 1991. Iterated runge–kutta methods on parallel computers. *SIAM journal on scientific and statistical computing*. **12** (5), pp. 1000–1028. Cited on p. 28.

van der Poel, E. P., Stevens, R. J., and Lohse, D. 2013. Comparison between two- and three-dimensional rayleigh–bénard convection. *Journal of Fluid Mechanics*. **736**, pp. 177–194. Cited on p. 21.

Verzicco, R. and Camussi, R. 1999. Prandtl number effects in convective turbulence. *Journal of Fluid Mechanics*. **383**, pp. 55–73. Cited on pp. 20 and 21.

Verzicco, R. and Camussi, R. 2003. Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell. *Journal of Fluid Mechanics*. **477**, pp. 19–49. Cited on p. 90.

Warnecke, J., Käpylä, P. J., Mantere, M. J., and Brandenburg, A. 2012. Ejections of magnetic structures above a spherical wedge driven by a convective dynamo with differential rotation. *Solar Physics*. **280**(2), pp. 299–319. Cited on pp. 34 and 35.

Warnecke, J., Käpylä, P. J., Mantere, M. J., and Brandenburg, A. 2013. Spoke-like differential rotation in a convective dynamo with a coronal envelope. *The Astrophysical Journal*. **778**(2), pp. 141. Cited on p. 34.

Weiss, N. 2001. Turbulent magnetic fields in the sun. *Astronomy & Geophysics*. **42** (3), pp. 3–10. Cited on p. 38.

Weiss, N. 2002. Dynamos in planets, stars and galaxies. *Astronomy & Geophysics*. **43**(3), pp. 3–9. Cited on p. 1.

Wicht, J., Stellmach, S., and Harder, H. 2010. Numerical dynamo simulations: from basic concepts to realistic models. In: *Handbook of geomathematics*. pp. 779–834. Cited on p. 2.

Wicht, J., Gastine, T., and Duarte, L. D. 2019. Dynamo action in the steeply decaying conductivity region of Jupiter-like dynamo models. *Journal of Geophysical Research: Planets*. **124**(3), pp. 837–863. Cited on p. 36.

Wilkinson, M., Lawrence, B., Prescott, A., et al. Jun 2021. UKRI supercomputing science case. Cited on pp. 4 and 110.

Yadav, R. K. and Bloxham, J. 2020. Deep rotating convection generates the polar hexagon on saturn. *Proceedings of the National Academy of Sciences*. **117**(25), pp. 13991–13996. Cited on p. 36.

Zeldovich, Y. B. 1957. The magnetic field in the two-dimensional motion of a conducting turbulent fluid. *Sov. Phys. JETP*. **4**, pp. 460–462. Cited on pp. 37 and 48.