MAGNETIC DYNAMOS: HOW DO THEY EVEN WORK?

MAGNETIC DYNAMOS: HOW DO THEY EVEN WORK?

By BENJAMIN B. H. Jackel, H.Sc. M.Sc

A Thesis Submitted to the School of Graduate Studies in Partial Fulfilment of the Requirements for the Degree Doctor of Philosophy

McMaster University DOCTOR OF PHILOSOPHY (2013) Hamilton, Ontario (Physics and Astronomy)

TITLE: Magnetic Dynamos: How Do They Even Work?

AUTHOR: Benjamin B.H. Jackel, H.Sc, M.Sc (McMaster University)

SUPERVISOR: Professor Ethan T. Vishniac

NUMBER OF PAGES: ix, 115

Abstract

The origin of cosmic magnetic fields is a important area of astrophysics. The process by which they are created falls under the heading of dynamo theory, and is the topic of this thesis. Our focus for the location of where these magnetic fields operate is one the most ubiquitous objects in the universe, the accretion disk. By studying the accretion disk and the dynamo process that occurs there we wish to better understand both the accretion process and the dynamo process in stars and galaxies as well.

We analyse the output from a stratified zero net flux shearing box simulation performed using the ATHENA MHD code in collaboration with Shane Davis. The simulation has turbulence which is naturally forced by the presence of a linear instability called the magnetorotational instability (MRI). We utilise Fourier filtering and the tools of mean field dynamo theory to establish a connection between the calculated EMF and the model predictions of the dynamically quenched alpha model. We find a positive correlation for both components parallel to the large scale magnetic field and the azimuthal components.

We have explored many aspects of the theory including additional contributions from magnetic buoyancy and an effect arising from the large scale shear and the current density. We also directly measure the turbulent correlation time for the velocity and magnetic fields both large scale and small. We can also observe the effects of the dynamo cycle, with the azimuthal component of the large scale magnetic field flipping sign in this analysis.

We find a positive correlation between the divergence of the eddy scale magnetic helicity flux and the component of the electromotive force parallel to the large scale magnetic field. This correlation directly links the transfer of magnetic helicity to the

dynamo process in a system with naturally driven turbulence. This highlights the importance of magnetic helicity and its conservation even in a system with triply periodic boundary conditions.

Acknowledgements

I would first of all like to thank my advisor, Ethan Vishniac for all of the support over the years even through all of the moving around and hassle. Secondly, Shane Davis whose simulations this project has depended on, and for taking part in the weekly discussions, performing new simulations and hosting the data.

The support of Gabe Davenyi, Dan Gallino and Matt Minnick for help proof reading, but also bouncing ideas about digital signal processing, fluid mechanics, statistics and much much more.

And finally, but most importantly, my wife Margaret who has put up with me being a student since forever and supporting me this whole time. Thank you.

Contents

List of Figures

Declaration of Academic Achievement

Under the guidance of my advisor, Ethan Vishniac, I carried out the analyses presented in this dissertation. The simulation data was provided by Shane Davis on the SciNet supercomputer at the Canadian Institute for Theoretical Astrophysics.

Chapter 1

Introduction

The disk is a very common object in the Universe. We see these disks in a very wide range of scales, from the structure of spiral galaxies, disks around supermassive black holes in active galactic nuclei, stars accreting material off a companion, and in the formation of stars and planets. Theoretical calculations and simulations extend this list to include places where we don't have the capability to observe directly but can infer the presence of disks. This includes mergers between compact objects like white dwarfs and neutron stars, around stellar mass black holes, and the final moments of a core collapse supernova. This list is not exhaustive, but highlights how prevalent disks really are in astrophysics.

To see why the disk shape is so common, we only need some fairly basic physics. If we consider a spherical blob of gas, it will tend to collapse if the internal pressure is not high enough. If the cloud is not rotating this will happen in an isotropic way. More complicated treatments of this collapse process give rise to Bonnor-Ebert (Bonnor, 1956; Ebert, 1955) spheres and the Lane-Emden equation (Lane, 1870; Emden, 1907).

Random motions and/or turbulence will impart a net angular momentum and so the collapse process plays out somewhat differently. Instead of collapsing isotropically, the centripetal force will act to prevent collapse in directions perpendicular to the angular momentum vector. In the parallel directions, viscous forces in the gas will damp vertical motions causing material to collect on the rotationally supported plane. The thickness of this plane is determined by the gas pressure and the vertical component gravity. The material in the plane, under the influence of gravity and internal pressure gradients, will be in an orbit around the mass accumulating in the centre, giving rise to the disk shape structure we see everywhere.

When the central object is the dominant source of gravity, the motion of the material in the disk can be described as Keplerian rotation where material looses angular momentum though the viscous forces present. As the angular momentum is transferred outward, the material falls towards the central object. We call this process accretion and so these disks are called accretion disks.

Disks are so common in the Universe, and yet we lack a deep understanding of how they behave. For instance, jets and outflows are often observed in tandem with disks and although observations for the launch of jets in these systems are not yet sufficiently resolved, theoretical models involving magnetized outflows from disks (eg. Blandford & Payne (1982)) or involving the interaction between rotating magnetized stars and the disk are strongly favoured (see eg. Pudritz et al [PPVI] for a review). Observations show these disks to be highly turbulent (Horne, 1995) and threaded with magnetic fields (Donati et al., 2005). Worse, the picture presented in our simple example has some worrying problems. If the particles in the disk are in orbits, then the Rayleigh stability criterion states that the disk should be stable against perturbations. We have a rough idea of the composition of the gas and so have an estimate for what the viscosity of the gas should be, for example Spitzer (1962). The problem is when we calculate an accretion timescale using this viscosity we get a timescale that is too large by orders of magnitude. Additionally, the magnetic field present in these disks is too large if it were simply condensed from the background interstellar field as the gas collapsed.

While there is much to understand, some progress has been made. Shakura & Sunyaev (1973) used a parameter, α, to relate the viscosity to the pressure scale height and sound speed in the disk. The viscosity is then argued to be enhanced by turbulence and furthermore, magnetohydrodynamic (MHD) turbulence. The disk structure can then be solved for observable quantities as functions of the α parameter. As a result, the α parameter is often used to describe the accretion rate, angular momentum transport, and turbulent stress in the disk.

In the Shakura & Sunyaev model, magnetic fields were also important for both transporting angular momentum and as a source of magnetohydrodynamic turbulence. The source of the turbulence was later found by Balbus & Hawley (1991) in their rediscovery of a plasma fluid instability originally proposed by Chandrasekhar (1960) and Velikhov (1959). This instability, called the magnetorotational instability (MRI) relies on the interaction between the magnetic field, differential rotation, and the plasma. The MRI is incredibly important as it provides a source of MHD turbulence, transports angular momentum, and plays a central role in driving a magnetic dynamo.

The primary method of studying the dynamo and the MRI is to use a shearing box simulation (Hawley et al., 1995). This work attempts to study the link between the MRI and a mean field dynamo model by testing the predictions of the model against such a local scale simulation. The goal is to be able to calculate quantities describing the dynamo and accretion process using the large scale properties of the disk for use in a global simulation where it is infeasible to resolve the microphysics.

Although the focus of this work is understanding the dynamo process in accretion disks, we note also that dynamos are a ubiquitous process in the universe, and the standard approach to understanding them has so far proved disappointing. A deeper understanding of the accretion disk dynamo may well prove helpful in improving our understanding of stellar and galactic dynamos.

The following work will be divided into three major topics. The first we will explain the background information needed to understand the models being developed. We will cover hydrodynamics and magnetohydrodynamics, the standard Shakura & Sunyaev model, and finally the mean field theory. The following chapter discusses the techniques used to analyse the data and details of the simulation used to produce the data. The third major chapter presents the results of this analysis.

Chapter 2

Theory

2.1 Hydrodynamics

The goal of this chapter is to develop the $\alpha - \Omega$ dynamo model by way of mean field dynamo theory. To do this we will need the tools of magnetohydrodynamics which relies on both electrodynamics and hydrodynamics.

2.1.1 Navier Stokes Equations

Before we start the derivation, there are some important assumptions which need to be made. Mostly these fall under what is referred to as the Continuum Hypothesis. This states that the fluid and its properties are continuous. The validity of this assumption rests on comparing the mean free path of a particle to a characteristic length scale of the problem.

Astrophysical systems cover a very large range of scales and densities, so we must be careful about applying the continuum hypothesis. For instance, stars and accretion disks are nearly perfect fluids owing to their relatively small scales and high densities.

On the other hand, the interstellar medium and solar wind are relatively diffuse and so the continuum hypothesis only holds on suitably large scales.

With these caveats in mind, we will start at the very beginning by considering a distribution function of particles in phase space, $f(\boldsymbol{x}, \boldsymbol{v}, t)$. Then we will construct an equation to calculate the probability of finding a single particle in this phase space volume with a given position, $\boldsymbol{x}$ and velocity, $\boldsymbol{v}$ as

$$\iint f(\boldsymbol{x}, \boldsymbol{v}, t)\, \mathrm{d}^3\boldsymbol{x}\, \mathrm{d}^3\boldsymbol{v} = N, \tag{2.1}$$

where N is the total number of particles.

We assume all the particles are the same, that they collide, and that their total number, momentum and energy is conserved. From this we can describe how the phase space density changes due to collisions (Louiville's Theorem states the phase space volume does not change) with

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \left(\frac{\partial f}{\partial t}\right)_{col}. \tag{2.2}$$

Using some calculus, we can define the total differential of $f(\boldsymbol{x}, \boldsymbol{v}, t)$ as

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + \frac{\partial x_i}{\partial t}\frac{\partial f}{\partial x_i} + \frac{\partial v_i}{\partial t}\frac{\partial f}{\partial v_i}, \tag{2.3}$$

where we have used Einstein summation notation. Switching to vector notation we get

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \boldsymbol{g} \cdot \frac{\partial f}{\partial \boldsymbol{v}} = \left(\frac{\mathrm{d}f}{\mathrm{d}t}\right)_{\mathrm{col}}. \tag{2.4}$$

Here, $\boldsymbol{g} = \frac{\partial \boldsymbol{v}}{\partial t}$ and is a place holder for forces. This equation is referred to as the Boltzmann equation and is where we will derive the Navier-Stokes equations from by taking moments.

The first moment we will take is to multiply Equation (2.4) by the particle mass, m, and integrate over $\mathrm{d}^3\boldsymbol{v}$ (with units given per volume) which gives

$$\int m\frac{\partial f}{\partial t}\,\mathrm{d}^3\boldsymbol{v} + \frac{\partial}{\partial \boldsymbol{x}}\int m\boldsymbol{v}f\,\mathrm{d}^3\boldsymbol{v} + \int m\frac{\partial(\boldsymbol{g}f)}{\partial \boldsymbol{v}}\,\mathrm{d}^3\boldsymbol{v} = \int m\,(\mathrm{d}ft)_{\mathrm{col}}\,\mathrm{d}^3\boldsymbol{v}. \tag{2.5}$$

To make sense of this, we note that the mass density is given by $\rho = \int mf\mathrm{d}^3\boldsymbol{v}$ and so we can identify the first term as $\frac{\partial\rho}{\partial t}$.

If we define the average of the velocity as,

$$\boldsymbol{U} = \frac{1}{\rho}\int mVf\,\mathrm{d}^3\boldsymbol{v}, \tag{2.6}$$

then the integral in the second term becomes, $\nabla\cdot(\rho\boldsymbol{U})$. If we can assume that $f\rightarrow 0$ as $\boldsymbol{v}\rightarrow\infty$, then the third term vanishes by use of the divergence theorem. Finally, the right hand side can be set to zero by enforcing conservation of particle number, momentum and energy for colliding particles (molecular chaos theorem). Putting this together yields,

$$\frac{\partial\rho}{\partial t} + (\boldsymbol{U}\cdot\nabla)\,\rho + \rho\nabla\cdot\boldsymbol{U} = 0 \tag{2.7}$$

referred to as the continuity equation which represents conservation of mass.

Next we will derive the conservation of momentum equation. In this case, we will multiply equation (2.4) by $m\boldsymbol{v}$ and integrate again. Through a similar process used for the continuity equation, this gives,

$$\rho\frac{\partial\boldsymbol{U}}{\partial t} + \rho\,(\boldsymbol{U}\cdot\nabla)\,\boldsymbol{U} = \nabla\cdot\boldsymbol{\sigma} + \boldsymbol{f}. \tag{2.8}$$

We have left equation (2.8) in a general form which is usually referred to as the Cauchy

Momentum equation. In this form, $\boldsymbol{\sigma}$ is the stress tensor while $\boldsymbol{f}$ represents body or external forces such as gravity and electromagnetic forces.

The final step is to multiply by $m\boldsymbol{v}^2$ and integrate to get the conservation of energy equation,

$$\frac{\partial \epsilon}{\partial t} + \boldsymbol{U} \cdot \nabla \epsilon = -\frac{p}{\rho} \nabla \cdot \boldsymbol{U} - \frac{1}{\rho} \nabla \cdot \boldsymbol{F} + \frac{1}{\rho} \Psi, \tag{2.9}$$

where ϵ is the specific internal energy, $\boldsymbol{F}$ is the conduction heat flux, and Ψ is the viscous dissipation rate.

The stress tensor is assumed to be isotropic and we define the diagonal elements to give the pressure, p. The other parts of the stress tensor can be used to derive the viscosity, which we will neglect here. Putting this together with the conservation equations gives,

$$\frac{\partial}{\partial t} \rho + \nabla (\rho \boldsymbol{U}) = 0 \tag{2.10a}$$

$$\rho \frac{\partial}{\partial t} \boldsymbol{U} + \rho \left(\boldsymbol{U} \cdot \nabla \right) \boldsymbol{U} = -\nabla p + \rho \boldsymbol{f} \tag{2.10b}$$

$$\frac{\partial}{\partial t} E + \nabla \cdot \left(\boldsymbol{U} \left(E + p \right) \right) = 0. \tag{2.10c}$$

2.1.2 Helicity

The final part of this section will deal with a quantity known as the helicity. The reasoning behind this is to set up its magnetic analogue which will be discussed in Section 2.2.4.

The helicity is a topological quantity of a fluid defined as,

$$H = \int_V \boldsymbol{U} \cdot \boldsymbol{\omega} \, \mathrm{d}V, \tag{2.11}$$

where $\boldsymbol{\omega} = \nabla \times \boldsymbol{U}$ is the vorticity. Physically it describes how twisted, knotted or linked the vortex lines in a fluid are.

This quantity is important since if the fluid is inviscid, incompressible and acted on by conservative forces, it is a conserved quantity. It is noteworthy to point out that in a real system, the helicity is destroyed by viscosity and so the conservation is broken. Worse, since the helicity has units of $\frac{E}{L}$ where E is in units of energy, we see that as the scale size decreases, the energy required for a given helicity decreases. This scaling is generally referred to as cascading, whereby if some quantity is injected at some higher scale it will cascade to smaller scales (the most recognizable example is the well known 5/3 law where energy in a turbulent cascade has a one dimensional power spectrum that scales as $k^{-5/3}$). Since the energy per logarithmic interval has a shallower dependence on eddy size than the helicity, it is trivial for the energy cascade to carry helicity to ever decreasing length scales. Finally, at the smallest scales, viscous losses are strongest and so the helicity is very efficiently destroyed by even small amounts of dissipation.

Nevertheless, fluid motions are somewhat easier to understand conceptually than magnetic phenomenon which will help us gain some insight into its magnetic analogue. A physical analogy of the conservation of helicity can be seen in Figure 2.1.

Figure 2.1: As an analogy, consider an elastic band with a twist in it. Suppose now we choose left handed and right handed twists to have opposite sign and define the helicity as the addition of all the twists. Then we try to untwist the band, without cutting it, we will end up creating a twist in the opposite sense. The net result is to leave the helicity in the band unchanged.

Finally, the helicity shares many aspects with the current helicity and magnetic helicity which will be discussed in the following section. All of these helicities play an important role in the dynamo process discussed in Section 2.5.

2.2 Magnetohydrodynamics

To develop the mean field dynamo theory we will need to utilise the tools of magnetohydrodynamics (MHD). This is a rich field which describes the evolution of the velocity and magnetic fields in a conducting plasma. It couples the equations developed in Section 2.1 with Maxwell's equations.

2.2.1 Assumptions

In addition to the assumptions of the Continuum Hypothesis, the magnetic field imposes some new constraints. In the directions that are perpendicular to the magnetic field, the Larmour radius and the skin depth must be small compared to the smallest length scale of the problem. The scale in the parallel direction must be large compared to the Landau damping scale. Finally, the ion gyration timescale must be short compared to the system's timescale. If the collision time scale is short compared any plasma time scale, then the microphysical plasma time and length scales are irrelevant.

The final assumption to discuss is that of the resistivity, η. The resistivity is simply defined as the inverse of the Ohmic conductivity. If η is negligible, then we are in the regime of Ideal MHD, else we are dealing with Resistive MHD.

The simulation used to produce the data discussed in this work assumes Ideal MHD, however numerical effects will act to mimic a resistive term giving rise to *numerical* resistivity. As such, in this section we will derive the equations for Resistive MHD so that we can follow the effects of the resistivity on our model.

2.2.2 Maxwell's Equations and Electrodynamics

The MHD equations, in part, describe the evolution of the magnetic field. The key to this evolution lies with Maxwell's equations and then later merging them with equations (2.10). Maxwell's equations are a collection of laws used to describe the dynamics of electric and magnetic fields and their relations to charged particles.

Maxwell's original work to model electricity and magnetism was based on fluid flow and later a mechanical model, it is perhaps not surprising then that the mathematics fit so nicely with those of fluid mechanics. Maxwell's contribution was to collect the then separate laws from Gauss, Ampere, Faraday, Coulomb, Poisson, Biot, and Savarte along with his own addition of a displacement current to create a complete model. The model predicted the presence of electromagnetic waves through use of the wave equation and even a calculation of the speed of light.

The full set of equations (including the Lorentz force law) are as follows:

$$\nabla \cdot \boldsymbol{E} = \frac{\rho_c}{\varepsilon_0} \tag{2.12a}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{2.12b}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{2.12c}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \left(\boldsymbol{J} + \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t} \right) \tag{2.12d}$$

$$\boldsymbol{F} = q \left(\boldsymbol{E} + \boldsymbol{U} \times \boldsymbol{B} \right). \tag{2.12e}$$

Equation (2.12a) is Gauss' law describing the electric field, $\boldsymbol{E}$ from a distribution of charge with density, ρ_c. Equation (2.12b) enforces the lack of a magnetic monopole. Faraday's law, Equation (2.12c), demonstrates the creation of an electric field from a changing magnetic field, $\boldsymbol{B}$. In a similar fashion, Ampere's law Equation (2.12d), describes the generation of a magnetic field from a current, $\boldsymbol{J}$ and changing electric field. Finally, though not usually included with the modern set of Maxwell's equations, is the Lorentz force law which describes the force on a particle with charge, q and velocity, $\boldsymbol{U}$, by the electric and magnetic fields. The units used here are SI so that ε_0 is the permittivity and μ_0 is the permeability of free space.

One final piece of information from electromagnetism will be needed to arrive at the MHD equations, that is, the generalised Ohm's law:

$$\eta \boldsymbol{J} = \boldsymbol{E} + \boldsymbol{U} \times \boldsymbol{B} \tag{2.13}$$

where η is the resistivity.

2.2.3 Derivation

To derive the MHD equations, it would be entirely reasonable to start with the Boltzmann equation and follow the same process we did in Section 2.1. For the sake of brevity we will choose a more direct approach.

First we must make some more assumptions about the fluid, namely that it is now a plasma with an equal number of positive and negative ions to remain overall neutral. For simplicity, we will assume that the plasma is fully ionised, though this is not strictly required. Additionally, we will set $\mu_0 = 1$ for ease of notation.

Next we will make a small change to Equation (2.12e) by calculating instead the force density,

$$\boldsymbol{F} = \rho_c \boldsymbol{E} + \boldsymbol{J} \times \boldsymbol{B}. \tag{2.14}$$

Then we add this force to Equation (2.10b) through the body force term $\boldsymbol{f}$,

$$\rho \frac{\partial \boldsymbol{U}}{\partial t} + \rho \left(\boldsymbol{U} \cdot \nabla \right) \boldsymbol{U} = -\nabla p + \rho_c \boldsymbol{E} + \boldsymbol{J} \times \boldsymbol{B} + \rho \boldsymbol{f}, \tag{2.15}$$

where we have left room in the $\boldsymbol{f}$ term in case we want to add gravity or other forces. We now note that the plasma is neutral ($\rho_c = 0$) and so the term with the electric field is zero. Now we use a vector identity along with Equation (2.12d),

$$\boldsymbol{J} \times \boldsymbol{B} = \left(\nabla \times \boldsymbol{B} \right) \times \boldsymbol{B} = \frac{1}{2} \nabla B^2 - \left(\boldsymbol{B} \cdot \nabla \right) \boldsymbol{B},$$

to get,

$$\rho \frac{\partial \boldsymbol{U}}{\partial t} + \rho \left(\boldsymbol{U} \cdot \nabla \right) \boldsymbol{U} + \left(\boldsymbol{B} \cdot \nabla \right) \boldsymbol{B} = -\nabla p^* + \rho \boldsymbol{f}, \tag{2.16}$$

where we have defined the effective pressure as $p^* = p + \frac{1}{2} B^2$.

Adding the Lorentz force couples Maxwell's equations to the Navier-Stokes equations, but now we need an equation to describe the evolution of the magnetic field. This is accomplished through Faraday's law, Equation (2.12c), and Ohm's law, Equation (2.13). We first take the curl of Ohm's law,

$$\nabla \times \boldsymbol{E} + \nabla \times \left(\boldsymbol{U} \times \boldsymbol{B} \right) = \eta \nabla \times \boldsymbol{J}. \tag{2.17}$$

We note that, $\nabla \times \boldsymbol{J} = \nabla \times \left(\nabla \times \boldsymbol{B} \right)$, and $\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$ to get,

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times \left(\boldsymbol{U} \times \boldsymbol{B} \right) + \eta \nabla \times \left(\nabla \times \boldsymbol{B} \right). \tag{2.18}$$

Finally, another vector identity,

$$\nabla \times \left(\nabla \times \boldsymbol{B} \right) = \nabla \cdot \boldsymbol{B} + \nabla^2 \boldsymbol{B},$$

and Equation (2.12b) yields,

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B}. \tag{2.19}$$

Through similar manipulation, we can get the conservation of energy equation as,

$$\frac{\partial E}{\partial t} + \nabla \cdot (\boldsymbol{U}(E + p^*) + \boldsymbol{B} \times (\boldsymbol{U} \times \boldsymbol{B})) = 0. \tag{2.20}$$

Collecting the conservation equations we arrive at the full set of resistive MHD equations,

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho \boldsymbol{U}) = 0 \tag{2.21a}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{U}) + \rho (\boldsymbol{U} \cdot \nabla) \boldsymbol{U} + (\boldsymbol{B} \cdot \nabla) \boldsymbol{B} = -\nabla p^* + \rho \boldsymbol{f} \tag{2.21b}$$

$$\frac{\partial}{\partial t}B = \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B} \tag{2.21c}$$

$$\frac{\partial}{\partial t}E + \nabla \cdot [\boldsymbol{U}(E + p^*) + \boldsymbol{B} \times (\boldsymbol{U} \times \boldsymbol{B})] = 0. \tag{2.21d}$$

If we set the resistivity to zero in Equation (2.21c) the equations reduce to the ideal MHD equations. Ideal MHD is notable since the magnetic field lines become frozen into the fluid. Even though there is a small amount of numerical resistivity expected in the simulation, the approximation of ideal MHD provides us with a useful insight into the dynamics. Resistive MHD on the other hand, allows for diffusion of the field lines across the fluid. Additionally, field lines are allowed to cross and reconnect which has important implications in for example, the solar corona and plasma confinement.

2.2.4 Magnetic Helicity

One interesting aspect of MHD is the striking similarities between concepts in hydrodynamics and magnetic phenomenon. One of those similarities is the analogue of the kinetic helicity we introduced in Equation (2.11). This analogue is the magnetic helicity,

$$H_M = \int_V \boldsymbol{A} \cdot \boldsymbol{B} \, \mathrm{d}V, \tag{2.22}$$

where $\boldsymbol{A}$ is the vector potential and $\boldsymbol{\nabla} \times \boldsymbol{A} = \boldsymbol{B}$. From here on, we will write the magnetic helicity with subscript M and the kinetic helicity with subscript K. Like the kinetic helicity, the magnetic helicity is a topological invariant in ideal MHD.

The conservation of both the kinetic and magnetic helicities is broken when there is a finite viscosity or resistivity. The key difference is that the magnetic helicity converges to being conserved in the limit of zero resistivity, while the kinetic helicity does not. Following our explanation of how the kinetic helicity conservation is broken, the magnetic helicity has units of EL. This scaling means that the magnetic helicity is concentrated on *large* scales, effectively protecting it from the dissipative effects of resistivity acting on the small scale. We call this type of scaling an inverse cascade.

Perhaps the biggest difference between the kinetic and magnetic helicities, is that due to the definition of H_M relying on the vector potential, it is not a gauge invariant quantity. There are ways of dealing with this though, such as a perfectly conducting or zero net flux boundaries. Another way involves using the current helicity,

$$h_c = \boldsymbol{J} \cdot \boldsymbol{B}, \tag{2.23}$$

where we have now introduced the concept of helicity *density*. Out of simplicity, we will now refer to the kinetic, magnetic, and current helicities as,

$$h_K = \boldsymbol{U} \cdot \boldsymbol{\omega},$$
$$h_M = \boldsymbol{A} \cdot \boldsymbol{B},$$
$$h_c = \boldsymbol{J} \cdot \boldsymbol{B},$$

respectively.

These quantities will prove critical in the development of the mean field dynamo model and indeed this entire work.

2.2.5 Dimensionless Quantities

To tie up the sections on hydrodynamics and magnetohydrodynamics, we will discuss the dimensionless forms of these equations. Or rather, some dimensionless quantities associated with them. We will derive perhaps the most common of these, the Reynolds number and then simply state some of the others which will be relevant.

We begin by writing the *viscid* Navier-Stokes momentum equation,

$$\rho \frac{\partial \boldsymbol{U}}{\partial t} + \rho \left(\boldsymbol{U} \cdot \boldsymbol{\nabla} \right) \boldsymbol{U} - \mu \nabla^2 \boldsymbol{U} = -\nabla p + \rho \boldsymbol{f}, \tag{2.24}$$

where μ is the dynamic viscosity. This equation is essentially Newton's second law, $\boldsymbol{F} = m\boldsymbol{a}$ multiplied by the density, ρ, and so every term has units of $\rho \frac{L}{T^2}$ or $\rho \frac{V^2}{L}$. To make the equation dimensionless we choose a characteristic length scale, L, and velocity, V and then multiply by $\frac{L}{\rho V^2}$. Then we define dimensionless quantities,

$$\boldsymbol{U'} = \frac{\boldsymbol{U}}{V},\ p' = \frac{1}{\rho V^2},\ \boldsymbol{f'} = \frac{L\boldsymbol{f}}{V^2},\ \frac{\partial}{\partial t'} = \frac{L}{V}\frac{\partial}{\partial},\ \nabla' = L\nabla.$$

Putting this together yields,

$$\frac{\partial \boldsymbol{U'}}{\partial t'} + \left(\boldsymbol{U'} \cdot \nabla'\right) \boldsymbol{U'} - \frac{\mu}{\rho L V} \nabla'^2 \boldsymbol{U'} = -\nabla' p' + \boldsymbol{f'}.$$

This expression looks almost the same as our Navier-Stokes equation, except for the $\frac{\mu}{\rho L V}$ term which we will define as,

$$Re \equiv \frac{\rho L V}{\mu}, \tag{2.25}$$

or the Reynolds number. In words, the Reynolds number tells us the relative strength of inertial forces to viscous forces.

Often, the Navier-Stokes equation will be written as,

$$\frac{\partial \boldsymbol{U}}{\partial t} + \left(\boldsymbol{U} \cdot \boldsymbol{\nabla}\right) \boldsymbol{U} - \frac{1}{Re} \nabla^2 \boldsymbol{U} = -\nabla p + \boldsymbol{f}, \tag{2.26}$$

with the reasoning that fluids with the same Reynolds numbers and same geometry are similar. The whole idea behind creating scale models in wind tunnels is that if by adjusting the speed of the wind you can match the Reynolds number, then the behaviour of the full scale can be inferred.

For our purposes, however, the Reynolds number has another important property. That is, at large Reynolds numbers the fluid flow becomes turbulent. There is no precise number where this occurs, but for terrestrial systems the boundary is at $Re \approx 10^3$ and fully developed turbulence at $Re \approx 10^5$. Astrophysical flows, particularly Keplerian flows found in accretion disks are thought to behave differently than terrestrial fluid flows. For instance, even though accretion disks from simple scaling estimates have Reynolds numbers of $> 10^{10}$, the Coriolis force may stabilise a purely hydrodynamic flow. In the presence of even a weak magnetic field however, the plasma becomes highly unstable and turbulent, owing to the MRI (Balbus & Hawley, 1991). There are a host of other dimensionless quantities used in all areas of fluid mechanics and MHD, we will define here only the relevant ones.

Reynolds Number	$Re \equiv \frac{\rho L V^2}{\mu}$	Ratio of inertial to viscous forces
Prandtl Number	$Pr \equiv \frac{c_p \mu}{k}$	Ratio of viscous to thermal diffusion rate
Rossby Number	$Ro \equiv \frac{V}{Lf}$	Ratio of inertial to Coriolis forces
Magnetic Reynolds Number	$R_m \equiv \frac{VL}{\eta}$	Ratio of magnetic advection to magnetic diffusion
Magnetic Prandtl Number	$Pr_m \equiv \frac{\nu}{\eta}$	Ratio of viscosity to resistivity

2.3 Shakura and Sunyaev α Model

We take a brief diversion from our discussion of MHD before we introduce the mean field dynamo theory and the α model of accretion, Shakura & Sunyaev (1973).

The motivation for this model is to parametrise angular momentum transport in an accretion disk. If we consider a disk of material, then this disk will be stable against perturbations if Rayleigh's stability criterion is satisfied, namely

$$\frac{\partial (R^2 \Omega)}{\partial R} > 0. \tag{2.27}$$

If the disk is Keplerian, meaning each piece is in an orbit, then this criterion is always satisfied and the disk should be stable. The viscosity of the fluid will define a timescale for accretion or equivalently the angular momentum transport. Arguments based on the properties of the material in the disk give an estimate of this timescale that is longer than observed values by orders of magnitude (using for instance the values of molecular viscosity from Spitzer (1962)). Shakura & Sunyaev noted that an effective viscosity enhanced by turbulence and magnetic fields could explain the discrepancy.

They began by assuming that the stress tensor scales with the sound speed,

$$W_{R\phi} = \alpha c_s^2. \tag{2.28}$$

From this, we can define an effective viscosity, $\nu = \alpha c_s H$, where c_s is the sound speed and H is the scale height. Since the Maxwell stress tensor can be included here, magnetic fields were certain to be important but their role was not well understood.

In addition to providing a measure for the accretion rate, the α parameter can be used to estimate other observables of the disk such as the midplane temperature and density by choosing an opacity law. Even though it is rather dubious to parametrise turbulence by a single value, α is still useful in describing the observable properties of the disk. The model we wish to develop is motivated by being able to define an α value based on the large scale properties as a function of position in the disk. Thus providing a more fine grained approach with the intent of explaining local and transient phenomenon in observed disks, or use as a subgrid model in global simulations of accretion disks.

2.4 The Magnetorotational Instability

To explain the anomalous viscosity, Balbus & Hawley (1991) proposed a linear plasma instability called the magnetorotational instability (MRI). This instability, originally discovered by Velikhov (1959) and Chandrasekhar (1960) by analysing Couette flow in a rotating hydromagnetic system was later applied to the astrophysical accretion disk flow by exploring the weak field limit. The MRI functions on the principle that a restoring force can be unstable in a differentially rotating system. The restoring force in the accretion disk is due to the magnetic field applying tension between fluid elements and the differential rotation comes from the Keplerian velocity profile. A visual representation can be seen in Figure 2.2.

More formally, the equations of motion for a fluid element at radius R_0 in orbit with angular velocity Ω can be solved using a linear perturbation approach to give solutions of the form $e^{i\omega t}$. Here ω satisfies,

$$\omega^2 = 4\Omega_0^2 + R\frac{\mathrm{d}\Omega^2}{\mathrm{d}R}, \tag{2.29}$$

where $\Omega_0 = \Omega(R_0)$. For ω to be real, $\frac{\partial(R^2\Omega)}{\partial R} > 0$, which for an accretion disk with a Keplerian velocity profile is satisfied. This result is referred to as the Rayleigh Criterion. The types of solutions obtained this way are referred to as epicycles, wherein a perturbed object will perform a retrograde orbit about its equilibrium point as it orbits the central object with frequency κ.

To see how the magnetic tension (denoted by the placeholder, K) can destabilise, we can do the same stability analysis. The resulting equation for ω is now,

$$\omega^4 - (2K + \kappa^2)\omega^2 + K(K + R\frac{\mathrm{d}\Omega^2}{\mathrm{d}R}) = 0, \tag{2.30}$$

where κ^2 is epicyclic frequency. If K is small enough, then ω will have imaginary value corresponding to exponential growth solutions. In the presence of a *weak* magnetic field, the stability criterion is now,

$$\frac{\partial(\Omega^2)}{\partial \ln R} > 0, \tag{2.31}$$

which is never satisfied in a Keplerian system. At some point depending on the conditions in the disk, the magnetic energy will achieve a maximum value and we say the MRI is saturated. Though it is unclear what factors might affect the saturation, in general the MRI saturates when the magnetic energy is in equipartition with the turbulent energy and is some fraction of the local gas pressure which corresponds to an $\alpha \lesssim 1$.

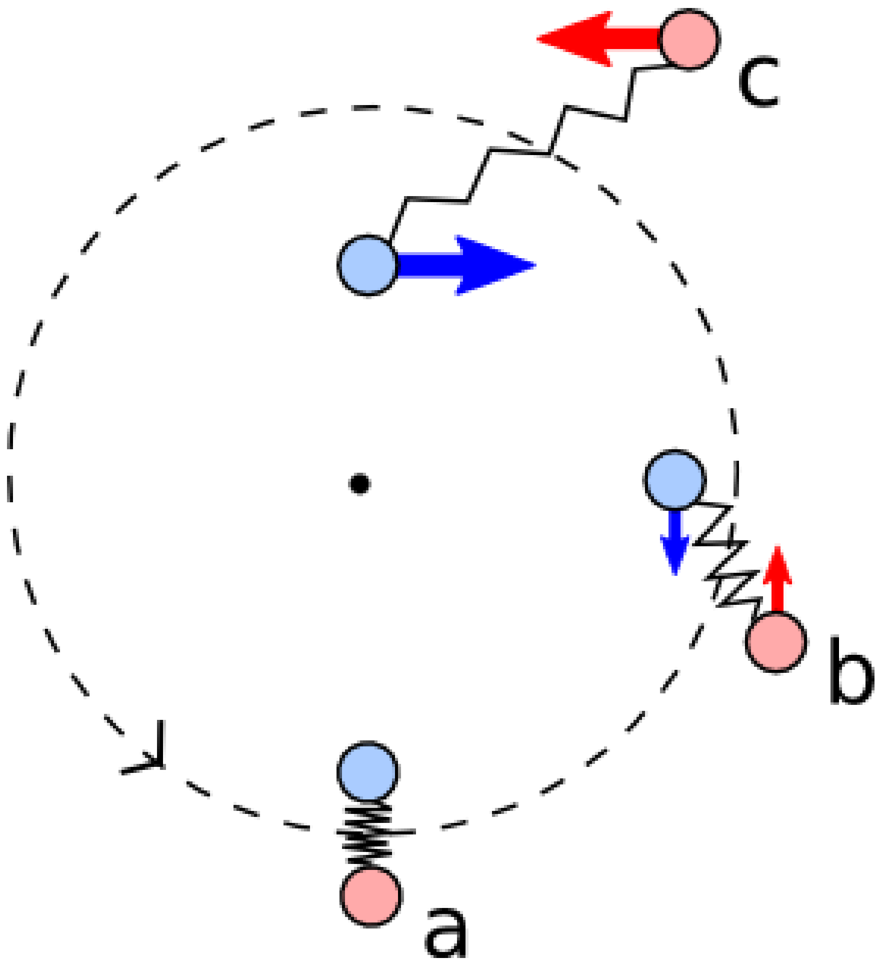


Figure 2.2: Here we consider adjacent fluid elements in orbit threaded by a magnetic field parallel to the angular momentum vector. If one of the fluid elements is displaced, the magnetic tension between the elements will increase. This tension will lead to the inner element falling to a lower orbit while the outer element will move to a higher orbit. The net result will be to increase the magnetic tension more, leading to instability.

2.5 Mean Field Dynamo

In this section we will develop the theory behind the model we employ. The basic idea is that we take the MHD equations and define them for, in our case, an accretion disk and separate them into two parts. The goal in doing this is to explain the large scale phenomena in terms of the small scale or turbulent behaviour.

2.5.1 Mean Field Separation

Mean field theory has a rich history, and what follows shares similarities to Large Eddy Simulations (Deardorff, 1970) and Reynold Averaged Navier Stokes. In the pioneering works by Moffatt (1978); Parker (1979); Krause & Raedler (1980), mean field theory was applied to the MHD equations. The first step of this process is to define what we mean by separating the equations into two parts. and separation meant that a field, $\boldsymbol{B}$ could be separated into a mean part and a fluctuating part,

$$\boldsymbol{B} = \overline{\boldsymbol{B}} + \boldsymbol{b}.$$

What we intend to do here is instead of a mean and fluctuating parts we will use large and small scale parts. The goal in separating the scales is to isolate the effects of the MRI on the large scale dynamo, as such the dividing line will be between the large scale dynamo peak and the MRI forcing scale as described in Section 3.0.7. In what follows, upper case letters will denote the total quantities, lower case will signify small scale or fluctuating quantities, and a bar will represent large scale. Before we can apply this to the MHD equations, we need to define the rules for this process. Typically, these are referred to as the Reynolds averaging rules (Monin & I'Aglom, 1971; Reynolds, 1895).

1. Linearity:

$$\overline{\boldsymbol{F}+\boldsymbol{G}} = \overline{\boldsymbol{F}} + \overline{\boldsymbol{G}}$$
$$c\overline{\boldsymbol{F}} = \overline{c\boldsymbol{F}}$$
$$\overline{c} = c$$

2. Derivatives:

$$\overline{\partial_t \boldsymbol{F}} = \partial_t \overline{\boldsymbol{F}}$$
$$\overline{\partial_x \boldsymbol{F}} = \partial_x \overline{\boldsymbol{F}}$$

3. Products:

$$\overline{\boldsymbol{F}\boldsymbol{G}} = \overline{\boldsymbol{F}}\;\overline{\boldsymbol{G}}$$

4. Additional properties:

$$\overline{\overline{\boldsymbol{F}}} = \overline{\boldsymbol{F}}$$
$$\overline{\boldsymbol{f}} = 0$$
$$\overline{\overline{\boldsymbol{F}\boldsymbol{G}}} = \overline{\boldsymbol{F}\boldsymbol{G}}$$
$$\overline{\overline{\boldsymbol{F}}\boldsymbol{g}} = 0$$
$$\overline{\partial_t \boldsymbol{f}} = 0$$
$$\overline{\partial_x \boldsymbol{f}} = 0$$

With these rules in mind we begin by defining the large and small scale quantities,

$$\boldsymbol{B} = \overline{\boldsymbol{B}} + \boldsymbol{b},\ \boldsymbol{U} = \overline{\boldsymbol{U}} + \boldsymbol{u},\ \boldsymbol{J} = \overline{\boldsymbol{J}} + \boldsymbol{j}.$$

The two equations we will need are the momentum equation and the induction equation. Averaging the induction equation gives,

$$\begin{aligned}\frac{\partial \overline{\boldsymbol{B}}}{\partial t} &= \nabla \times \left(\overline{\boldsymbol{U} \times \boldsymbol{B}}\right) + \eta \nabla^2 \overline{\boldsymbol{B}} \\ &= \nabla \times \left(\overline{\boldsymbol{U}} \times \overline{\boldsymbol{B}}) + \overline{\boldsymbol{u} \times \boldsymbol{b}}\right) + \eta \nabla^2 \overline{\boldsymbol{B}}.\end{aligned} \tag{2.32}$$

It is useful to define $\overline{\boldsymbol{u} \times \boldsymbol{b}}$ as the turbulent electromotive force or EMF or $\mathcal{E}$. If we now subtract Equation (2.32) from Equation (2.21c) and combine terms we get,

$$\frac{\partial \boldsymbol{b}}{\partial t} = \overline{\boldsymbol{B}} \cdot \nabla \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \overline{\boldsymbol{B}} + \boldsymbol{G}, \tag{2.33}$$

where $\boldsymbol{G} = \nabla \times \left(\boldsymbol{u} \times \boldsymbol{b} - \overline{\boldsymbol{u} \times \boldsymbol{b}}\right)$.

If we apply the same treatment to the momentum equation, we get

$$\frac{\partial \overline{\boldsymbol{U}}}{\partial t} = -\overline{\boldsymbol{U}} \cdot \nabla \overline{\boldsymbol{U}} - \overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}} + \overline{\boldsymbol{B}} \cdot \nabla \overline{\boldsymbol{B}} + \overline{\boldsymbol{b} \cdot \nabla \boldsymbol{b}} - \nabla(\overline{P} + \overline{p}) - \frac{1}{2}(\overline{B^2} + \overline{b^2}), \tag{2.34}$$

and

$$\begin{aligned}\frac{\partial \boldsymbol{u}}{\partial t} = &- (\overline{\boldsymbol{U}} \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\overline{\boldsymbol{U}} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \overline{(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}} \\ &+ (\overline{\boldsymbol{B}} \cdot \nabla)\boldsymbol{b} + (\boldsymbol{b} \cdot \nabla)\overline{\boldsymbol{B}} + (\boldsymbol{b} \cdot \nabla)\boldsymbol{b} - \overline{(\boldsymbol{b} \cdot \nabla)\boldsymbol{b}} \\ &- \nabla p + \frac{1}{2}\nabla \overline{b^2} - \frac{1}{2}\nabla b^2 - \nabla \boldsymbol{b} \cdot \overline{\boldsymbol{B}}\end{aligned}$$

2.5.2 Transport Coefficients

At this stage we will derive the standard approach in mean field dynamo theory. That is, we wish to be able to describe the electromotive force in terms of the large scale magnetic field, $\overline{\boldsymbol{B}}$. The common choice is to assume that the EMF can be expanded in terms of derivatives of $\overline{\boldsymbol{B}}$,

$$\mathcal{E}_i = \alpha_{ij}\overline{B}_j + \beta_{ijk}\partial_k\overline{B}_j + \dots . \tag{2.35}$$

Where α and β are pseudo-tensors and are referred to as the turbulent transport coefficients.

If we assume the turbulence is homogeneous and isotropic, then the tensors reduce to scalars and Equation (2.35) reduces to (keeping only the first derivative),

$$\overline{\boldsymbol{\mathcal{E}}} = \alpha\overline{\boldsymbol{B}} + \beta\overline{\boldsymbol{J}}, \tag{2.36}$$

where $\alpha = \delta_{ij}\alpha_{ij}$ and $\beta = \beta_{ijk}\epsilon_{ijk}$. Because of the properties of $\overline{\boldsymbol{B}}$ and $\overline{\boldsymbol{\mathcal{E}}}$, α is really a pseudo-scalar, meaning $\alpha = -\alpha$ when $\boldsymbol{x} \to -\boldsymbol{x}$.

At this point the first major approximation comes in, namely we assume that $\boldsymbol{G}$ in Equation (2.33) is small which is valid when $\boldsymbol{b} \ll \overline{\boldsymbol{B}}$. This is usually referred to as the first order smoothing approximation, or FOSA. Equation (2.33) then becomes,

$$\frac{\partial \boldsymbol{b}}{\partial t} = \nabla \times \left(\boldsymbol{u} \times \overline{\boldsymbol{B}}\right). \tag{2.37}$$

From here, the EMF can be solved by integrating this equation, taking the cross product with $\boldsymbol{u}$ and filtering,

$$\overline{\boldsymbol{\mathcal{E}}} = \overline{\boldsymbol{u} \times \int_0^t \nabla \times \left(\boldsymbol{u} \times \overline{\boldsymbol{B}}\right) \, \mathrm{d}t}. \tag{2.38}$$

This equation can be solved and give the simplest form of the transport coefficients as $\alpha = -\frac{1}{3}\tau_c \overline{\boldsymbol{u} \cdot \boldsymbol{\omega}}$ and $\beta = -\frac{1}{3}\tau_c \overline{u^2}$, where τ_c is the correlation time derived from approximating the integral.

There are a number of other approximation schemes to arrive at the transport coefficients, such as Eddy Damped Quasi-normal Markovian (EDQNM) approximation and Minimal Tau (MTA); a thorough discussion of these and other topics can be found in Brandenburg & Subramanian (2005).

We recognise now that α is just the kinetic helicity from Equation (2.11). This is the crucial piece in what is referred to as the $\alpha - \Omega$ dynamo, where a poloidal field is generated from a toroidal one. The α term plays the role of transforming and amplifying the toroidal field into a poloidal one while the Ω effect is the shear turning the poloidal into a toroidal.

2.5.3 Catastrophic Quenching

This form of the $\alpha - \Omega$ dynamo enjoyed a lot of success in solar and geo dynamo models. There are some serious problems with the model with criticism from Gruzinov & Diamond (1994) and Cattaneo & Hughes (1996) about where the kinematic dynamo assumptions are valid, for example.

One problem is that the FOSA assumption breaks when $\boldsymbol{b}$ was found to grow much more quickly than $\overline{\boldsymbol{B}}$ (Kulsrud & Anderson, 1992). Additionally, we made the assumption that the turbulence was homogeneous and isotropic, but it can not be isotropic, else the average of the EMF would be zero. Homogeneity must be broken as well to have any growth from $\nabla \times \left(\boldsymbol{u} \times \overline{\boldsymbol{B}}\right)$, Blackman & Field (1999).

The biggest concern with this model, however, is that of *catastrophic α quenching*. Vainshtein & Cattaneo (1992) noted that the α effect was sharply suppressed in the high magnetic Reynolds number regime resulting in quenching of the dynamo for even small mean fields. Since astrophysical systems typically have enormous values of Rm while simultaneously having large magnetic fields, this pointed to a failure of the model. Vainshtein & Cattaneo (1992) worked out a correction factor of the form,

$$\alpha = \frac{\alpha_0}{1 + R_m \overline{B}^2 / B_{eq}^2}, \tag{2.39}$$

where B_{eq}^2 is the equipartition energy density.

2.5.4 Dynamic Quenching

The physical reasoning behind why the α effect quenches in these types of simulations is largely due to the conservation of magnetic helicity which we defined in Section 2.2.4. The reason has to do with how both the kinetic and magnetic helicity conservation behaves in the presence of viscosity and resistivity (recall, numerical viscosity and resistivity are present in simulations regardless of whether ideal MHD is employed). Viscosity and resistivity dissipate energy most efficiently at small scales. Kinetic helicity cascades down to small scales while the magnetic helicity cascades to larger scales, this means that h_K is most affected by dissipation while h_M remains largely untouched. This concept also helps explains the α effect's dependence on Magnetic Reynolds number.

To see the link between magnetic helicity conservation and the α effect, we take a look at the magnetic α effect (Pouquet et al., 1976). It was soon discovered that adding this term backreacted on the kinetic α term which acts as a natural, dynamic way of quenching α. The α term in Equation (2.36) is now,

$$\alpha = \alpha_K + \alpha_M = -\frac{1}{3}\tau_c \overline{\boldsymbol{u} \cdot \boldsymbol{\omega}} + \frac{1}{3\rho}\tau_c \overline{\boldsymbol{j} \cdot \boldsymbol{b}}. \tag{2.40}$$

We can relate the α term to the magnetic helicity since $\overline{\boldsymbol{j} \cdot \boldsymbol{b}} \approx k_f^2 \overline{\boldsymbol{a} \cdot \boldsymbol{b}}$ (Mitra et al., 2010), where $\overline{\boldsymbol{a} \cdot \boldsymbol{b}}$ is the small scale magnetic helicity. This now ties in with our discussion of the conservation of the kinetic and magnetic helicities as well as how their spectral densities cascade. That is, the conservation of magnetic helicity places a critical constraint on the dynamo theory. Therefore, any successful dynamo theory must deal with the conservation of magnetic helicity or else it will pile up on small scales and strongly quench the dynamo. We will see that the transfer of magnetic helicity from small scales to large, i.e. $\overline{\boldsymbol{a} \cdot \boldsymbol{b}}$ to $\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}$ also depends on the EMF. So a nonzero EMF facilitates the accumulation of small scale magnetic helicity with a sign which tends to cancel a nonzero EMF.

2.5.5 Magnetic Helicity Flux Driven Dynamo

In this section we will explore how the magnetic helicity flux driven by the current helicity in the α_M term can drive a dynamo (Vishniac & Cho, 2001). Since we are working with a gauge dependant quantity, it is important to define which gauge we are working with. That is, the Coulomb Gauge where, $\nabla \cdot \boldsymbol{A} = 0$ and

$$\boldsymbol{A}(\boldsymbol{r}) = \int \frac{\boldsymbol{J}(\boldsymbol{r'})}{4\pi|\boldsymbol{r} - \boldsymbol{r'}|}\, \mathrm{d}^3\boldsymbol{r'}. \tag{2.41}$$

This gauge choice preserves an approximately constant proportionality between the small scale current and magnetic helicities.

We can define the time evolution of the vector potential as,

$$\frac{\partial \boldsymbol{A}}{\partial t} = \boldsymbol{U} \times \boldsymbol{B} - \nabla\Phi - \eta \boldsymbol{J}, \tag{2.42}$$

where $\nabla^2\Phi = \nabla \cdot (\boldsymbol{U} \times \boldsymbol{B})$. Now, using the induction equation we can write

$$\frac{\partial \boldsymbol{A} \boldsymbol{\cdot} \boldsymbol{B}}{\partial t} = -\eta \boldsymbol{J} \boldsymbol{\cdot} \boldsymbol{B} - \nabla \cdot \left[\boldsymbol{B}\Phi + \boldsymbol{A} \times (\boldsymbol{U} \times \boldsymbol{B}) - \eta \left(\boldsymbol{A} \times \boldsymbol{J}\right)\right]. \tag{2.43}$$

We define the divergence term as a flux of the magnetic helicity density, $\boldsymbol{J_H}$ to get

$$\partial_t(\boldsymbol{A} \boldsymbol{\cdot} \boldsymbol{B}) = -\eta \boldsymbol{J} \boldsymbol{\cdot} \boldsymbol{B} - \nabla \cdot \boldsymbol{J_H}. \tag{2.44}$$

Following our scale separation procedure, we can define the large and small scale evolution equations,

$$\partial_t(\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}) + \nabla \cdot \overline{\boldsymbol{J_H}} = 2\overline{\boldsymbol{\mathcal{E}}} \cdot \overline{\boldsymbol{B}} - 2\eta \overline{\boldsymbol{J}} \cdot \overline{\boldsymbol{B}}, \tag{2.45}$$

and

$$\partial_t(\boldsymbol{a} \boldsymbol{\cdot} \boldsymbol{b}) + \nabla \cdot \boldsymbol{j_h} = -2\overline{\boldsymbol{\mathcal{E}}} \cdot \overline{\boldsymbol{B}} - 2\eta \boldsymbol{j} \boldsymbol{\cdot} \boldsymbol{b}, \tag{2.46}$$

Where $2\overline{\boldsymbol{\mathcal{E}}} \cdot \overline{\boldsymbol{B}}$ appears in both equations with opposite sign, its role is to transfer magnetic helicity between scales. The large and small scale magnetic helicity fluxes are defined as,

$$\overline{\boldsymbol{J_H}} = \left(\boldsymbol{\mathcal{E}} + \nabla\overline{\Phi} + \eta \overline{\boldsymbol{J}}\right) \times \overline{\boldsymbol{A}}, \tag{2.47}$$

and

$$\boldsymbol{j_h} = (\boldsymbol{\epsilon} + \nabla\phi + \eta \boldsymbol{j}) \times \boldsymbol{a}. \tag{2.48}$$

In these expressions, $\nabla^2\overline{\Phi} = \nabla \cdot (\overline{\boldsymbol{U}} \times \overline{\boldsymbol{B}} + \boldsymbol{\mathcal{E}})$ and $\nabla^2\phi = \nabla \cdot \boldsymbol{\epsilon}$, where

$$\boldsymbol{\epsilon} = \overline{\boldsymbol{U}} \times \boldsymbol{b} + \boldsymbol{u} \times \overline{\boldsymbol{B}} + \boldsymbol{u} \times \boldsymbol{b} - \overline{\boldsymbol{u} \times \boldsymbol{b}}. \tag{2.49}$$

The importance of these quantities rests largely with the conservation of magnetic helicity. Accumulation of magnetic helicity will result in a contribution to the EMF with the opposite sign as the kinematic term leading to the quenching discussed in Section 2.5.3. If there is a non zero flux of magnetic helicity, then we expect a balance between the divergence of this flux and the EMF.

Chapter 3

Methods

In much of the work that follows, we will be using methods of Fourier analysis to both solve differential equations, decompose the data into two parts, and to perform various filtering techniques to the data. To that end, we need to define what the Fourier Transform is and some of its properties.

The Fourier Transform (FT) stems from the study of Fourier Series where any periodic function can be described by a possibly infinite set of sine and cosine waves. The FT is then the generalisation of a Fourier Series by removing the condition of periodicity, using the complex exponential form of the sine and cosine, and using an integral instead of summation.

The FT and its inverse are defined as,

$$\begin{aligned} \hat{f}(\boldsymbol{k}) &= \int_{-\infty}^{\infty} f(\boldsymbol{x}) e^{-i\boldsymbol{x}\cdot\boldsymbol{k}} \, \mathrm{d}\boldsymbol{x}, \\ f(\boldsymbol{x}) &= \int_{-\infty}^{\infty} \hat{f}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}} \, \mathrm{d}\boldsymbol{k}, \end{aligned} \tag{3.1}$$

Where we have defined the Fourier Transform on a 3-Dimensional domain so that

$\boldsymbol{x} = (x, y, z)$ and $\boldsymbol{k} = (k_x, k_y, k_z)$ and we denote $\mathrm{d}\boldsymbol{x} = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ and $\mathrm{d}\boldsymbol{k} = \mathrm{d}k_x\,\mathrm{d}k_y\,\mathrm{d}k_z$.

Traditionally, the FT transforms a function of amplitude versus time to amplitude versus frequency. In the notation presented in Equation (3.1), we instead transform a function defined in the spatial domain to one defined by wavenumbers, $\boldsymbol{k}$. An example of the box car function and its Fourier transform are given in Figure 3.1. The notation used in this work will use bold letters to denote vector quantities and the hat symbol to represent transformed quantities. For simplicity, we will sometimes refer to the transformed domain as k-space.

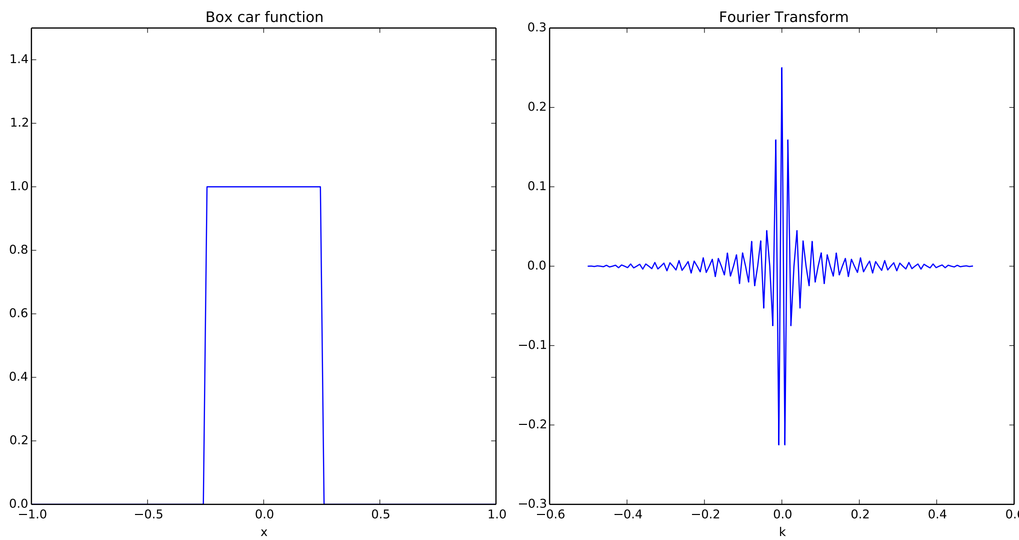


Figure 3.1: On the left is an example of a box car function centered around $x = 0$. On the right is the Fourier Transform of the given box car function, note that it is a sinc function ($f(x) = sin(x)/x$). This example will be of particular importance once we discuss the filtering techniques using the Fourier Transform.

3.0.6 Differential Equations with Fourier Transforms

To understand why the FT is so useful, we will explore some of the techniques used in this work and explain the salient properties as we go. The first technique is using the FT to solve differential equations. The essential point is that by taking the Fourier Transform we can turn a partial differential equation into an ordinary equation or an ordinary differential equation into a algebraic equation. We can show this by way of example using the Poisson Equation (in 1 dimension),

$$\nabla^2\Phi(x) = \rho(x).$$

The first step is to take the Fourier Transform of this equation,

$$-k^2\hat{\Phi}(k) = \hat{\rho}(k).$$

It is straightforward to show that the FT of $f'(x)$ is $ik\hat{f}(k)$ (using integration by parts),

$$\mathcal{F}f'(x) = \int_{\infty}^{\infty} f'(x)e^{-ixk}\,\mathrm{d}x = ik\int_{\infty}^{\infty} f(x)e^{-ixk}\,\mathrm{d}x = ik\hat{f}(k)$$

Given this, the Poisson equation is transformed into an algebraic equation which is easily solved. The final step is to invert transform to get the solution in the spatial domain. This last step can be difficult if $\rho(x)$ is complicated, luckily numerical techniques exist which we will discuss in Section 3.0.8.

3.0.7 Filtering Techniques

The next technique used in this work is digital filtering, specifically lowpass and highpass filtering. There is a vast wealth of resources on digital signal processing but by necessity we will need to restrict our attention to some fairly simple concepts. In simple terms, a lowpass filter is one in which given an input signal, only "low" frequencies are allowed to be transmitted. Conversely, a highpass filter is one in which "high" frequencies are transmitted. The "low" and the "high" frequencies passed are set by the filter design according to a cutoff frequency, f_c. We will use $\mathcal{L}$ to denote a lowpass filter operation and δ to represent a highpass filter. These filters have the property that $\mathcal{L} = 1 - \delta$ and vice versa. This property also implies that given a signal, A, it can be decomposed into two parts, $A = \mathcal{L}A + \delta A$.

The type of filter we have used in this work is referred to as an "ideal filter". It is named so because it provides the sharpest possible cutoff of frequencies or scales. It is also the least complicated to implement. In general terms, the implementation of a filter is quite straight forward. The transformed filtered output is simply the product of the Fourier Transformed signal with the filter function. The ideal lowpass filter function is then actually just the box car function from Figure 3.1, it cuts off any frequency with $|f| > f_c$ where f_c is half the width of the box car function.

At this point it is worthwhile to look at another interesting property of the FT, namely multiplication in k-space is equivalent to convolution in the spatial domain, or

$$\mathcal{F}f \cdot \mathcal{F}g = \mathcal{F}(f * g).$$

The consequences of our filter shape are now apparent, the ideal lowpass filter is the same as convolving our signal with a sinc function. This is usually referred to as ringing due to the oscillatory nature of the sinc function. How undesirable the ringing in the filtered output is depends on the goals of the filter. In our case, a sharp cutoff in k-space is the most important effect and so the ringing is an acceptable trade off.

The value of the cutoff frequency used is chosen by examining the magnetic power spectrum and noting where in k-space the MRI ($kH/(2\pi) = 4$) versus the large scale dynamo ($kH/(2\pi) = 0.8$) are operating as shown in Figure 3.2. We choose a value of $\frac{kH/(2\pi)}{=}2$ as the cutoff according to this criterion as well as maximising the correlations found in Chapter 4.

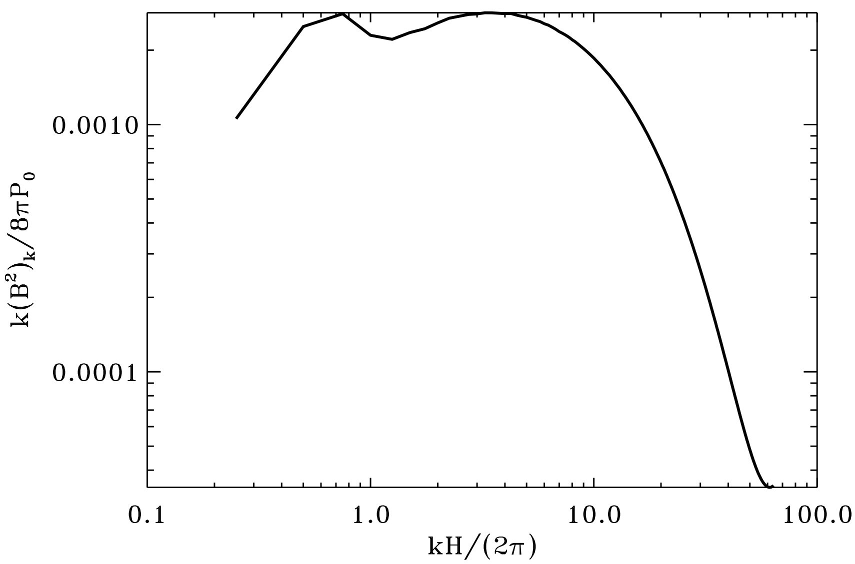


Figure 3.2: Power spectrum of the magnetic energy, weighted by the wavenumber k averaged over 250 orbits. The second peak in this spectrum at $\frac{k}{2\pi} = 4$ corresponds to the scale where the MRI is injecting turbulence. The first peak we associate with the large scale dynamo. The choice of cutoff frequency is designed to separate the effects of these two processes. Figure courtesy of Dr. Shane Davis.

3.0.8 The Fast Fourier Transform

The discussion of Fourier Transforms has so far assumed that the functions have been continuous and on an infinite domain. When working with data from a numerical simulation, this assumption no longer holds. Fortunately, the properties of the continuous Fourier Transform still apply to its discrete version,

$$\begin{aligned} \hat{f}_n &= \sum_{j=0}^{N-1} e^{-2\pi ijn/N} f_j, \\ f_j &= \frac{1}{N} \sum_{n=0}^{N-1} e^{-2\pi ijn/N} f_n. \end{aligned} \tag{3.2}$$

Though this is entirely correct, there is a major flaw with this approach to calculating a discrete transform. Since we need to sum to N for each value of n or j, the scaling of this calculation goes as $\mathcal{O}N^2$ which is expensive computationally. Instead, we use the Fast Fourier Transform (Cooley & Tukey, 1965), which exploits certain symmetries to arrive at a solution which scales as $\mathcal{O}N \log N$. The particular implementation of the FFT for this work uses the Numpy library in Python.

One important consequence of using a discrete transform, is that the domain on which we transform a function is finite. Since the Fourier Transform assumes an infinite domain, the resulting transform will then be periodic with a period equal to the domain size. Fortunately, in our case the data happens to be periodic as well, so Fourier Analysis will be a good fit.

3.1 Shearing Box Formalism

The central theme in accretion disk theory is the magnetorotational instability. There is much that is not known about this instability and so many simulations have been set up to investigate it. A major difficulty is that the turbulence generated by the MRI covers many scales which means performing a global simulation of an accretion disk is impractical given modern computational power. A local simulation is therefore the best option and early simulations were most interested in finding which simple quantities could predict the saturation amplitude of the MRI. A simulation performed in Hawley et al. (1995) used a shearing box type geometry to successfully determine the dependence of the saturated state of the MRI on certain physical quantities such as the initial mean magnetic field. This simulation laid the groundwork for many other local simulations used to study the properties of the MRI and the dynamo process.

3.1.1 Description

In the shearing box formalism we consider a piece of an accretion disk at some radius, R_0, which is small compared to the total radius of the disk and whose dimensions are small compared to R_0. The piece is corotating at the orbital frequency, Ω. Finally adopting a cartesian coordinate system where we map x to the radial, y to the azimuthal and z to the vertical coordinate, we can write the MHD equations as,

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho \boldsymbol{U}) = 0 \tag{3.3}$$

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} + (\boldsymbol{B} \cdot \nabla)\boldsymbol{B} = -\nabla p^* + \rho\Omega^2 (2qx\hat{\boldsymbol{x}} - z\hat{\boldsymbol{z}}) - 2\Omega\hat{\boldsymbol{z}} \times \rho\boldsymbol{u} \tag{3.4}$$

$$\frac{\partial}{\partial t}B = \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) \tag{3.5}$$

$$\frac{\partial}{\partial t}E + \nabla \cdot [\boldsymbol{U}(E + p^*) + \boldsymbol{B} \times (\boldsymbol{U} \times \boldsymbol{B})] = 0. \tag{3.6}$$

We have introduced a the total derivative notation, $\frac{\mathrm{D}}{\mathrm{D}t} \equiv \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla$, and the shear parameter $q = -\frac{\mathrm{d}\ln\Omega}{\mathrm{d}\ln r}$. For a Keplerian accretion disk, $q = 3/2$. The last term of Equation (3.4) corresponds to the Coriolis force while the second to last term represents the effective gravitational + centrifugal potential. To further specify the equations, the equation of state is chosen to be isothermal so that,

$$p = c_s^2 \rho, \tag{3.7}$$

where c_s is the sound speed. Adittionally, in the simulation used throughout this work, we have used the parameters set out in Davis et al. (2010). That is, the equation of state is isothermal with a sound speed of $c_s = 5 \times 10^{-7}$, orbital frequency $\Omega = 10^{-3}$ and $\rho_0 = 1$.

Where the shearing box derives its name is in how the boundaries are defined. We define $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ to be the radial, azimuthal and vertical direction respectively. In the azimuthal and vertical directions the box is strictly periodic, but to simulate the shearing environment the radial direction is shearing periodic. This can be expressed as (Hawley et al., 1995),

$$\begin{aligned}
f(x, y, z) &= f(x + L_x, y - q\Omega L_x t, z), \\
f(x, y, z) &= f(x, y + L_y, z), \\
f(x, y, z) &= f(x, y, z + L_z),
\end{aligned}$$

where L_x, L_y, L_z are the box dimensions and f is any function defined in the box. The azimuthal velocity across the x-boundary is a special case and is defined as,

$$v_y(x, y, z) = v_y(x + L_x, y - q\Omega L_x t, z) + q\Omega L_x. \tag{3.8}$$

At $t = \frac{nL_y}{q\Omega L_x}$ where n is a positive integer, the box becomes purely periodic. These special points will become important for dealing with Fourier decomposition. A visual representation of the shearing effect is given in Figure 3.3.

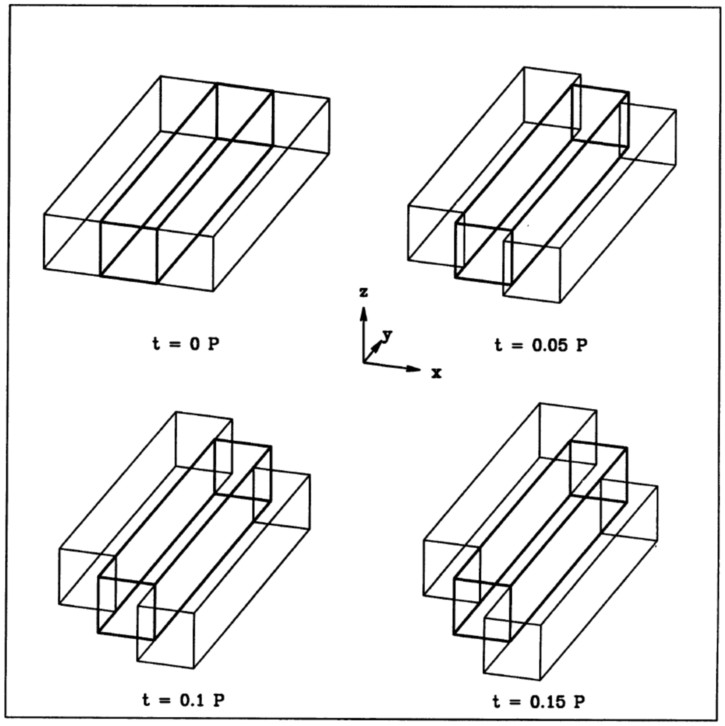


Figure 3.3: The boxes at $t = 0$ begin strictly periodic and over time slide past each other. At some later time, given by the periodic points, the boxes will go back to a configuration like at $t = 0$. Image from Hawley et al. (1995).

The shearing periodic boundary also places special constraints on Fourier Transforms. Since the FT assumes periodic functions, and the radial direction is not, then we need to modify our approach slightly. Normally we can define the wavevectors, k as we do for the vertical and azimuthal directions,

$$k_y = \frac{2\pi n_y}{L_y}, \tag{3.9}$$

$$k_z = \frac{2\pi n_z}{L_z}, \tag{3.10}$$

where $n_y, n_z = \ldots, -1, 0, 1, \ldots$. For the radial direction, k is now a function of time and we define it as,

$$k_x(t) = \frac{2\pi n_x}{L_x} + q\Omega k_y t. \tag{3.11}$$

When taking Fourier Transforms, whether to filter or solve differential equations, it is important that we do a change of variables $y \to y' - q\Omega x(t - t_n)$ where t_n is the nearest periodic point before transforming. Then change back after doing whatever Fourier technique. This process is referred to as mapping and remapping.

3.1.2 Stratification and Zero Net Flux

An important consideration when investigating these local simulation is the initial configuration of the magnetic field which threads it. In Hawley et al. (1995), the initial field orientation was chosen to be vertical to emulate the effect of the dipole field from the central object piercing the disk or an advected field. They found that the saturation state of the MRI depended on the strength and orientation of this initial field. If zero net flux is chosen, simulations find that the saturation level is independent of the initial configuration. For this reason, this simulation was performed with zero net flux. The initial configuration of the magnetic field is sinusoidal in the radial direction. For simplicity, many shearing box simulations neglect vertical gravity. As computing power increased higher resolution has been possible and worryingly a dependence of the MRI activity on the box resolution has been found (Pessah et al., 2007; Fromang & Papaloizou, 2007; Pessah et al., 2007; Käpylä & Korpi, 2011; Lesur & Longaretti, 2007). While many simulations did neglect vertical gravity some did not, for instance Brandenburg et al. (1995); Stone et al. (1996) include the effects of vertical gravity. In these simulations however, the effects of increasing resolution and long integration times were not established and so the convergence problem was not noticed. A possible explanation might be that unstratified simulations lack an outer scale or that a secondary instability may be at work (Vishniac, 2009).

Including vertical gravity allows the pressure and density to change as a function of vertical position. A pressure scale height is defined as the distance where the pressure falls by a factor of e and is given by,

$$H = \frac{c_s}{\Omega}, \tag{3.12}$$

where c_s is the sound speed and Ω is the orbital frequency. Additionally, stratification allows for magnetically buoyant motion of plasma in the vertical direction with implications for instability or inclusion in the dynamo process. In support of this idea, stratified simulations have now begun to show MRI activity that is independent of resolution (Davis et al., 2010; Shi et al., 2010). Here we consider a zero net flux shearing box where vertical gravity has been included. This choice means that the density is stratified and is a function of z is given by,

$$\rho = \rho_0 e^{-\frac{z^2}{H^2}}. \tag{3.13}$$

Vertical gravity forces a special consideration for the gravitational potential, since the vertical periodic boundary presents some problems. To keep discontinuities from arising, the potential must be smoothly reversed at the vertical boundary. This is done by applying a smoothing function,

$$\left(\left[(\xi \mp 1)^2 + \xi^2\lambda^2\right]^{1/2} \mp \xi\right)^2, \tag{3.14}$$

to the gravitational potential,

$$\Phi(z) = \frac{\Omega^2 z^2}{2}. \tag{3.15}$$

Here $\xi = z_0/z$, z_0 is the midplane or $L_z/2$, and $\lambda = 0.1$ is the width of the smoothing function.

The particular shearing box we are investigating here has a resolution given by 128 cells per scale height and an aspect ratio of $1H \times 4H \times 4H$. An estimate of the numerical resistivity is $\eta = 5.8 \times 10^{-8}$ through looking at the errors in the induction equation as calculated.

3.1.3 Analysis Procedure

The simulations were computed in collaboration with Shane Davis using the ATHENA MHD code, details for this code can be found in Gardiner & Stone (2005); Stone & Gardiner (2010); Stone et al. (2008). The parameters used in the simulation are identical to those in Davis et al. (2010), particularly the run which they refer to as S128R1Z4.

An estimate of the scalar resistivity and found to be approximately 5.8×10^{-8}. This value was found by measuring the errors in the induction equation and was computed by Dr. Shane Davis. It should be noted that numerical resistivity does not necessarily have the same properties as the scalar resistivity introduced in Section 2.2.

The outputs analysed correspond to a point in the simulation when the dynamo has reached a saturated state after around 50 orbits. The outputs span a timescale which corresponds to approximately 4.3 orbits.

An output containing the magnetic and velocity field as well as the density at every cell location is generated at each timestep for a total of 426 files. Each file is read into memory sequentially and ran through the analysis pipeline.

The filtering is done by first performing an FFT operation, multiplying by the filter then performing an inverse FFT resulting in a filtered quantity (either high or low pass depending on the filter used). In this way the filter design is modular and can accommodate different filtering techniques such as the Hamming and Gaussian filter designs.

Spatial derivatives are performed by using the method outlined in Section 3.0.6 and includes the divergence, gradient and curl operations. Temporal derivatives are performed using a central difference and 3 consecutive output files.

After each quantity is calculated, a horizontal average (over the x-y plane) is done and is saved to the hard drive. These files can be read in again if plotting or other analysis is desired.

The specific PYTHON functions used to generate these quantities can be found in the Appendix.

Chapter 4

Results and Discussion

4.1 Mean Field Assumptions

We begin the discussion of the shearing box simulation by testing some of the assumptions of mean field dynamo theory. Starting with the induction equation and checking that the standard choices about which terms to drop are justified.

The plots generated in this chapter have been horizontally averaged (over the $x-y$ plane) and expressed as functions of the vertical direction, z. Finally, the quantities being plotted are the time average over 4.3 orbits comprised of 426 individual outputs from the simulation. This provides us with a sample that is large compared to the correlation timescale and to provide robust time averages. The correlation timescale will be formally calculated in Section 4.2.4, however the sample comprises a total of $\sim 27t\Omega$ compared to the correlation timescale of approximately $0.2t\Omega$. Our first test will be to check that we can separate the large scale induction equation from the small and to test the lowpass filter. We use the Pearson Correlation Coefficient (r) to describe the correlation between the curves with +1 being a perfect correlation and -1 a perfect anti-correlation. The eventual correlation we will perform later in the chapter involves the component parallel to the large scale magnetic field. Since the azimuthal field is by far the strongest component, we will perform correlations of vector quantities in the y-direction. The results of this test of scale separation is shown in Figure 4.1.

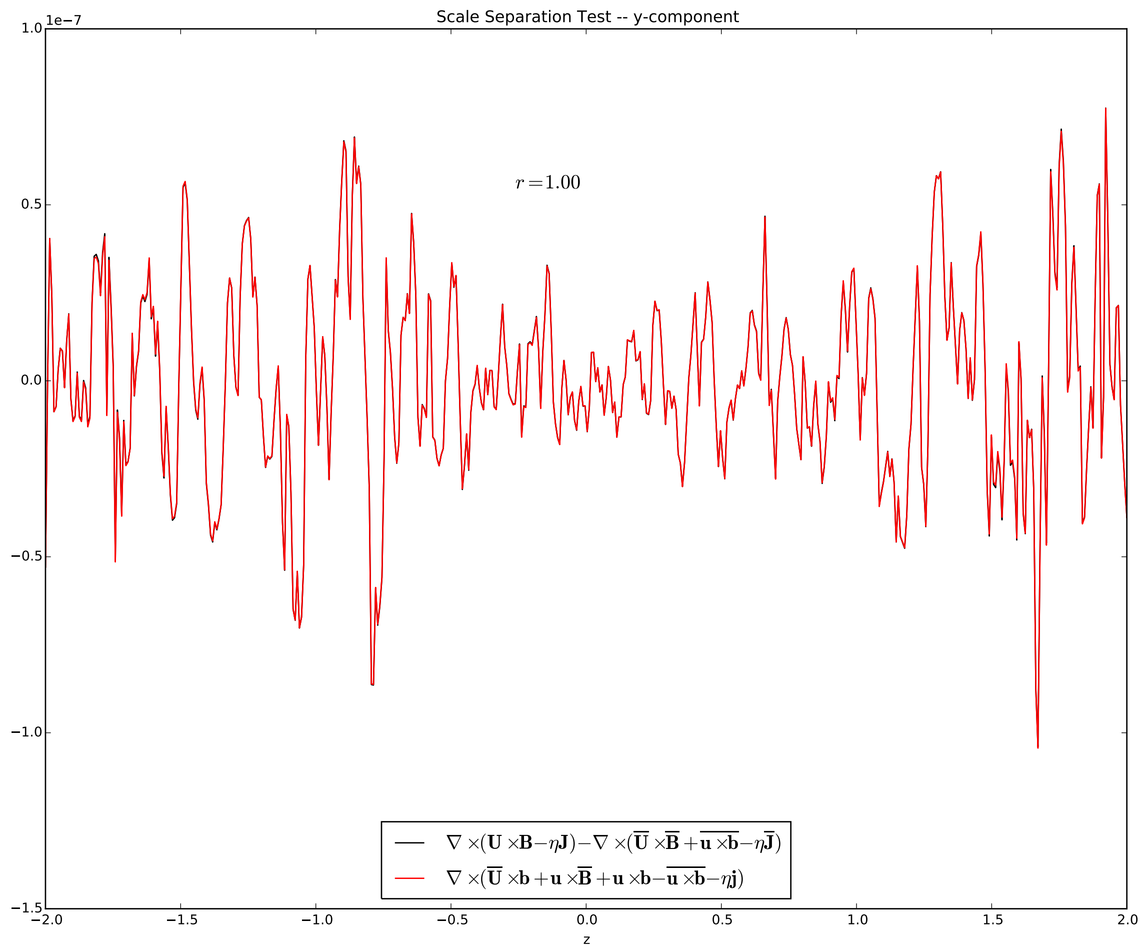


Figure 4.1: Presented here is the test of the scale separation procedure for the y-component of the induction equation. This test is done by subtracting the large scale induction equation, defined in Equation (2.32) from the total. The result is compared to the small scale induction equation as defined in Equation (2.33). We see a very strong agreement here, with correlation of $k_\tau = 0.99$ suggesting the filters are doing a good job of separating large scale from small.

Next we will test which terms are negligible, such as those dropped in FOSA. That is, we wish to see if $\partial_t \boldsymbol{b} = \nabla \times \left(\boldsymbol{u} \times \overline{\boldsymbol{B}}\right)$ is a good approximation to the small scale induction equation. The results of this test indicate that terms dropped are significant, resulting in a decrease of 0.25 in the correlation. This shown in Figure 4.2.

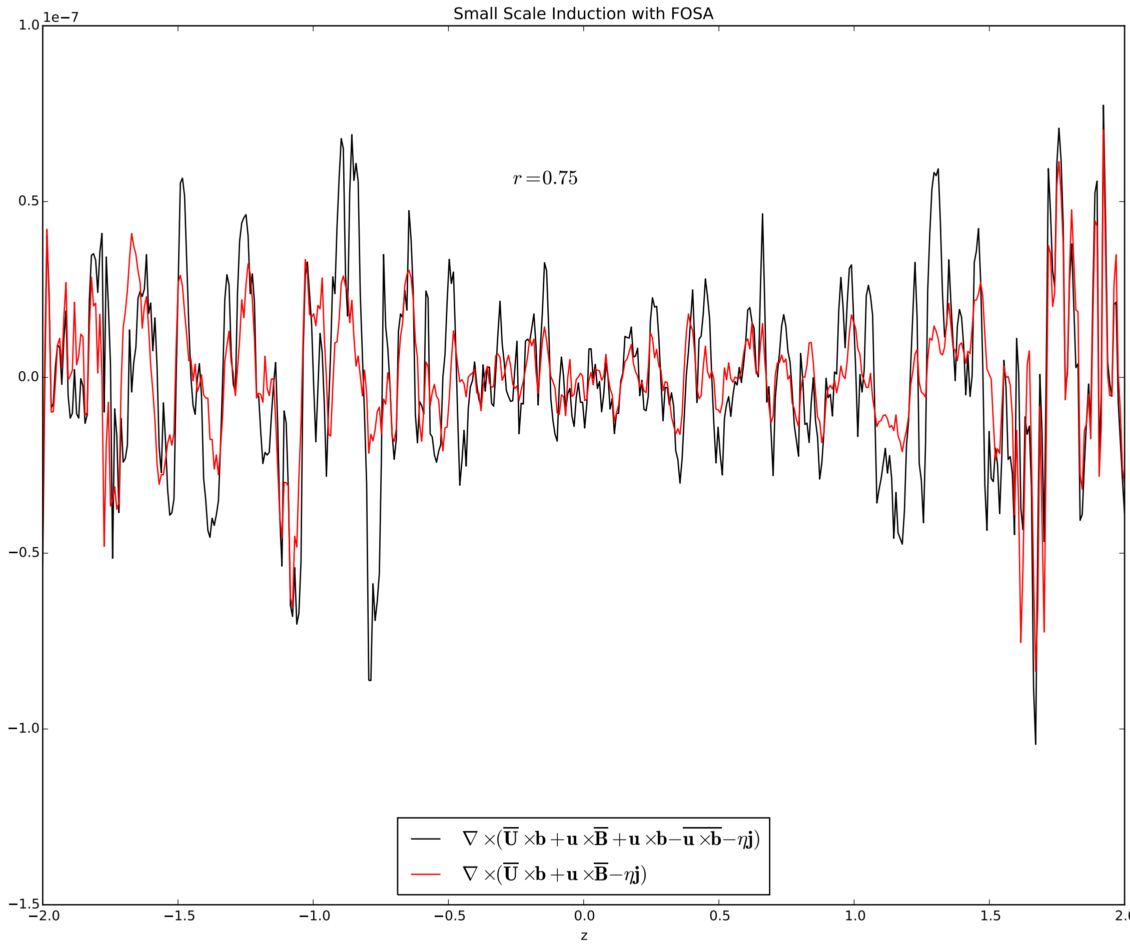


Figure 4.2: Here we see that the first order smoothing approximation results in some loss of information at a cost of approximately 25% correlation.

Next, we investigate the effects of numerical resistivity on the small scale induction equation. The test involves correlating the small scale induction equation with the resistive term against the equation without the resistive term. We can see from Figure 4.3 that the resistivity is a very minor effect for the small scale induction equation.

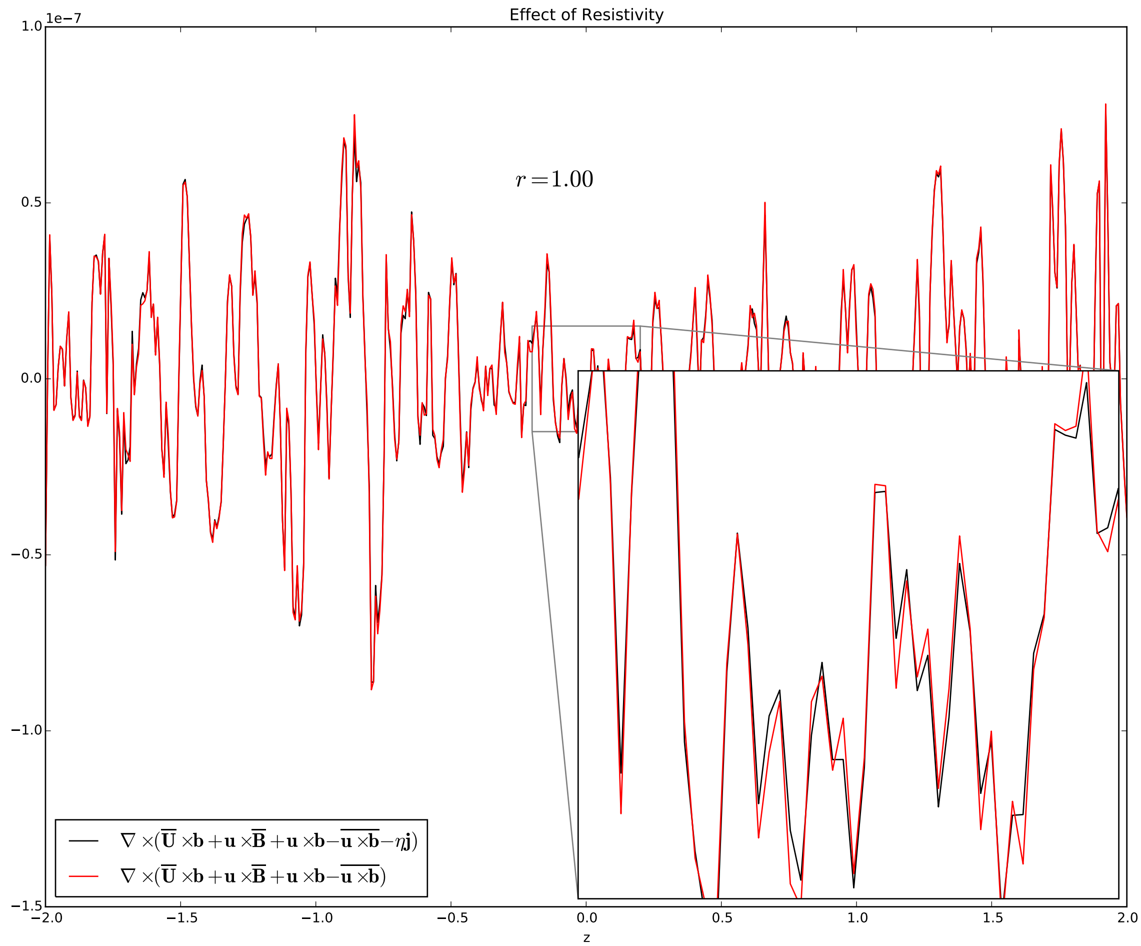


Figure 4.3: Plotted is the small scale induction equation with and without resistivity included. The resistivity adds only a very small amount to the small scale induction equation. Even in the inset, zoomed in by a factor of 5, we see the curves nearly lie ontop of one another.

4.2 EMF Expansion

An additional test we can perform at this stage is to check some of the assumptions that go into the calculation of the transport coefficients, α and β. In Section 2.5.2, we described the standard method used to get $\alpha = -\frac{1}{3}\tau_c\overline{\boldsymbol{u}\cdot\boldsymbol{\omega}}$ by integrating

$$\partial_t \boldsymbol{\mathcal{E}} = \overline{\boldsymbol{u} \times \dot{\boldsymbol{b}}}. \tag{4.1}$$

What we will do here is calculate the whole expression,

$$\frac{\partial \boldsymbol{\mathcal{E}}}{\partial t} = \overline{\boldsymbol{u} \times \dot{\boldsymbol{b}}} + \overline{\dot{\boldsymbol{u}} \times \boldsymbol{b}}, \tag{4.2}$$

where dots represent time derivatives. Each of the terms can be expanded by plugging in the small scale induction and momentum equations giving,

$$\overline{\boldsymbol{u} \times \dot{\boldsymbol{b}}} = \boldsymbol{u} \times \left[\nabla \times \left(\overline{\boldsymbol{U}} \times \boldsymbol{b} + \boldsymbol{u} \times \overline{\boldsymbol{B}} + \boldsymbol{u} \times \boldsymbol{b} - \overline{\boldsymbol{u} \times \boldsymbol{b}} - \eta \boldsymbol{j}\right)\right], \tag{4.3}$$

and,

$$\begin{aligned}\overline{\dot{\boldsymbol{u}} \times \boldsymbol{b}} =& [-(\overline{\boldsymbol{U}} \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\overline{\boldsymbol{U}} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \overline{(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}} \\ &+ (\overline{\boldsymbol{B}} \cdot \nabla)\boldsymbol{b} + (\boldsymbol{b} \cdot \nabla)\overline{\boldsymbol{B}} + (\boldsymbol{b} \cdot \nabla)\boldsymbol{b} - \overline{(\boldsymbol{b} \cdot \nabla)\boldsymbol{b}} \\ &- \nabla p^* + 3/2\Omega u_x \hat{y} - 2(\boldsymbol{\Omega} \times \boldsymbol{u})] \times \boldsymbol{b},\end{aligned} \tag{4.4}$$

where the effective pressure is given by

$$p^* = p - \frac{1}{2}\overline{b^2} + \frac{1}{2}b^2 + \boldsymbol{b} \cdot \overline{\boldsymbol{B}}, \tag{4.5}$$

and the last two terms correspond to the shear and Coriolis forces respectively.

These tests are given in Figures 4.4, 4.5, and 4.6

To test this aspect, we take a numerical derivative of the turbulent velocity and magnetic fields crossed with the appropriate quantity and compare against Equation (4.3) and Equation (4.4). The specific time in the simulation these tests were performed was arbitrary, however the result is indicative for any time during the simulation. Since the numerical time derivatives of $\boldsymbol{b}$ and $\boldsymbol{u}$ are good approximations of the small scale induction and momentum equations, we expect only a small deviation with errors due to the numerical time derivative and the inherent chaotic motions in the turbulent quantities.

For the first test in Figure 4.4, there is a good match suggesting there are no issues with the treatment of the small scale induction equation and this step of the transport coefficient calculation. In Figure 4.5 however, we see a much lower correlation. It is unclear why this is, however it does point to a problem in the calculation of the coefficients and that the turbulence may be causing too much noise in this test or there is an error in the calculation of the small scale momentum equation for the radial/vertical components. Further investigation into this behaviour to find the underlying cause is warranted. Finally, in Figure 4.6 the previous tests are combined which provides some insight into the values of the correlation coefficients found later in this chapter. That is, the low correlation observed for the test involving the small scale momentum equation may be affecting the calculation of the transport coefficients suggesting that this is a physical result.

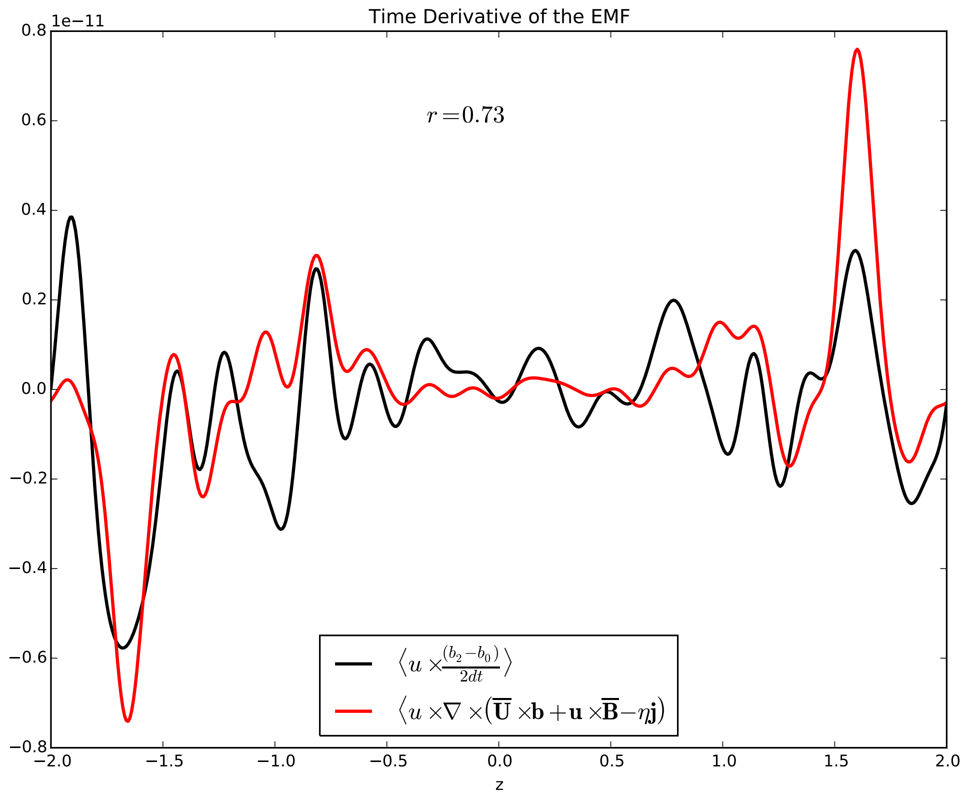


Figure 4.4: Here the cross product of the turbulent velocity and the small scale induction equation are plotted against the cross product of the turbulent velocity and the numerical time derivative of the turbulent magnetic field. The strong correlation ($r = 0.73$) suggests there are no problems with the treatment of the small scale induction equation and the transport coefficients calculated from this term.

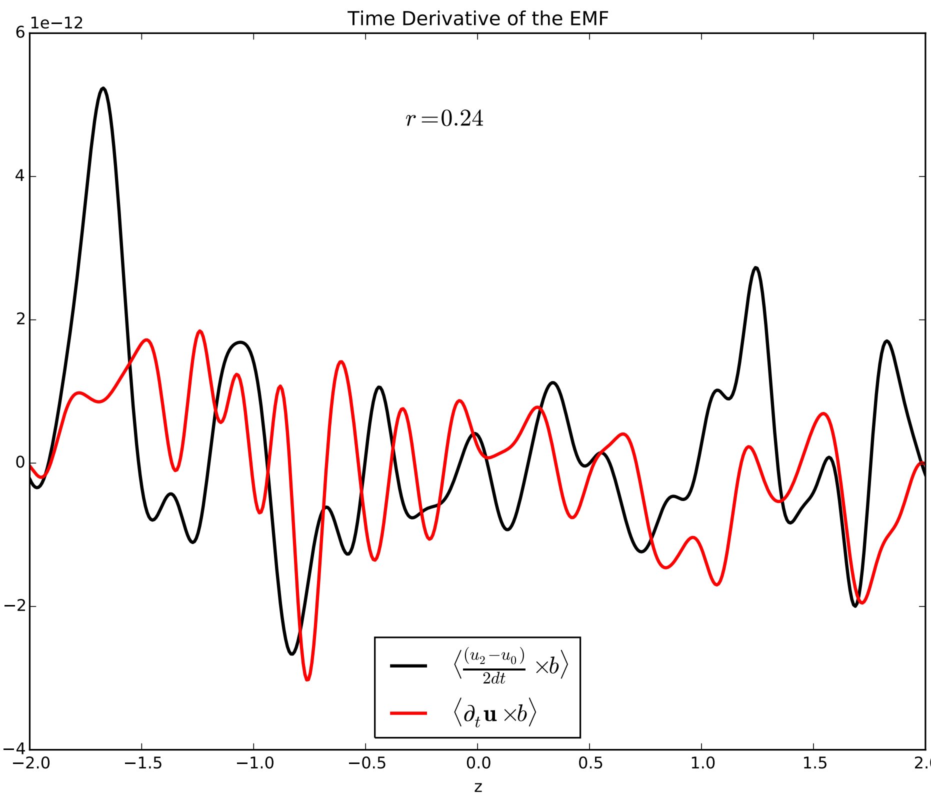


Figure 4.5: Here the cross product of the turbulent magnetic field and the small scale momentum equation are plotted against the cross product of the turbulent magnetic field and the numerical time derivative of the turbulent velocity. A weak correlation ($r = 0.24$) suggests there may be an issue with the calculation of the small scale momentum equation or the turbulent motions may be at fault. Transport coefficients calculated based on these assumptions may also be affected.

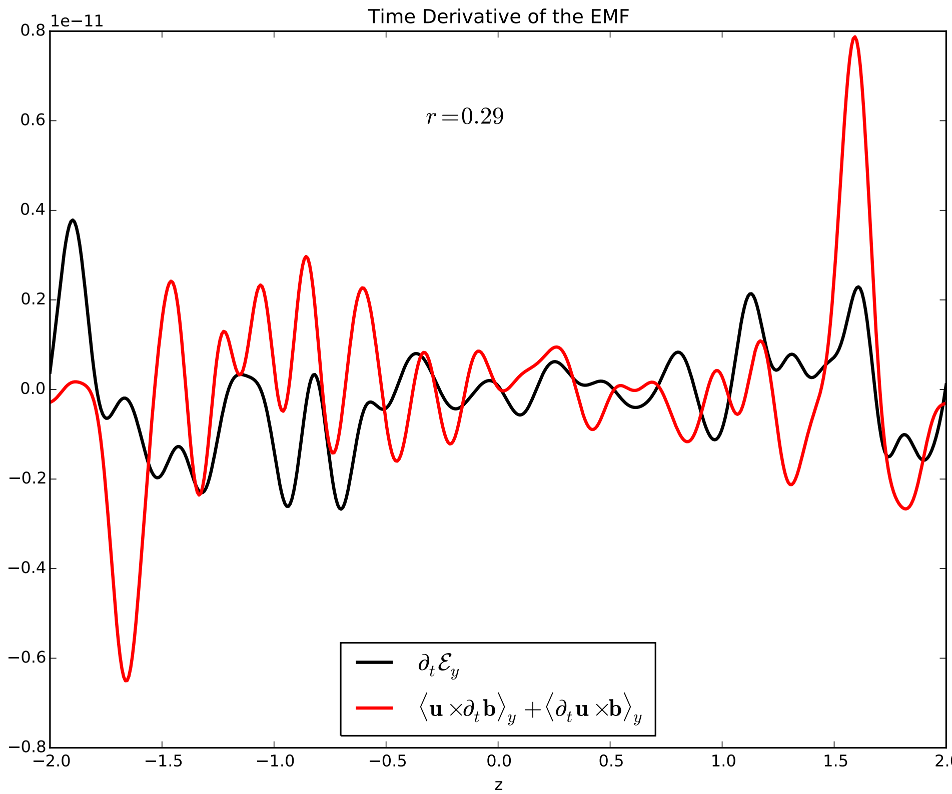


Figure 4.6: The time derivative of the EMF with the calculations using the induction and momentum equations. We see a weak correlation owing to the weak correlation observed from the term involving the momentum equation.

4.2.1 Mean Field Predictions

In this section we will calculate the mean field α model predictions and compare them against the EMF measured from the simulation. The preliminary values for τ are determined in Section 4.2.4 and then a least squares regression is done within a factor of 2 of that value to maximise the correlation. In this procedure, one value is chosen for all of the data sets which are then time averaged.

In preparation for comparing the divergence of the magnetic helicity flux, we will dot the expressions for the mean field with the large scale magnetic field. In our first test we plot the standard α prediction against the measured EMF (dotted with the large scale magnetic field) and can be seen in Figure 4.7. Since this model does not account for the conservation of magnetic helicity, we should see the effects of quenching. Indeed, a weak anti-correlation indicates the presence of the back reaction on the EMF.

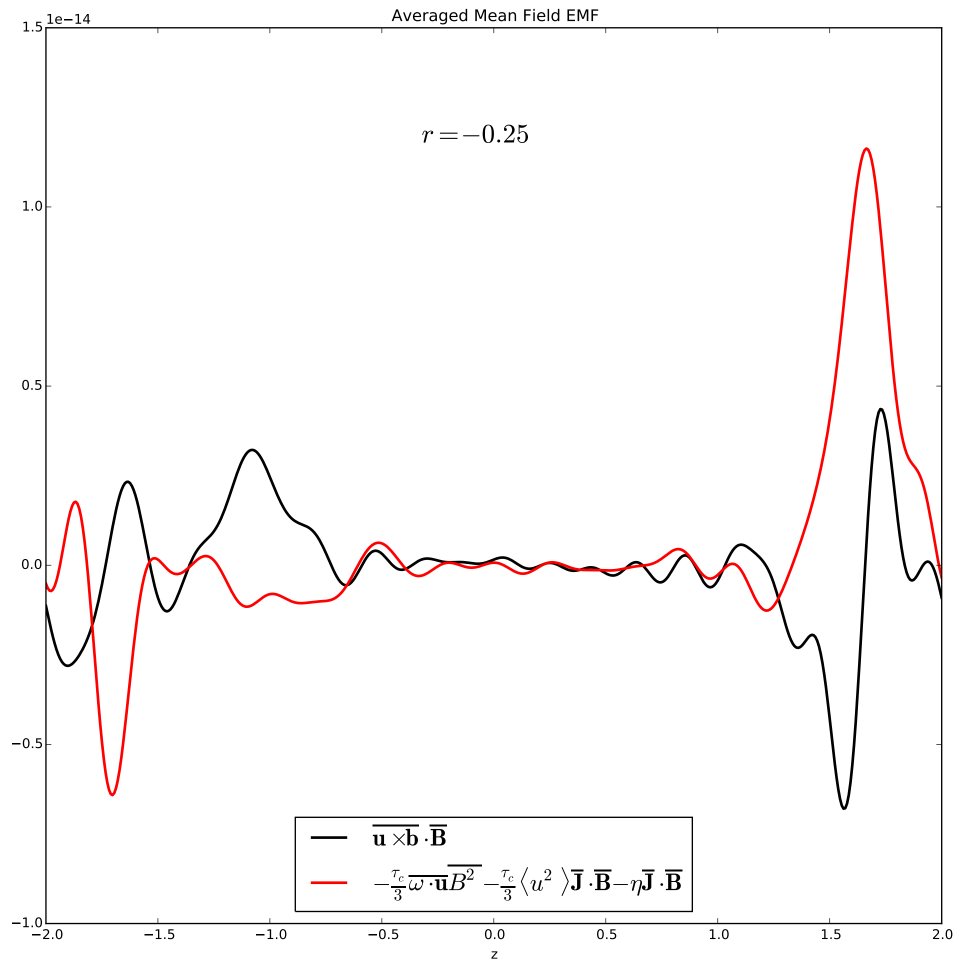


Figure 4.7: Presented here is the standard mean field α model overplotted with the measured EMF from the simulation. This plot represents a time average of the sample data. The standard model in this case is based simply the kinetic helicity which we expect should be a poor approximation based on the arguments about catastrophic quenching. Indeed we see an anti-correlation with the EMF.

The dynamical quenching model adds the magnetic α term and we again dot this with the large scale magnetic field in Figure 4.8.

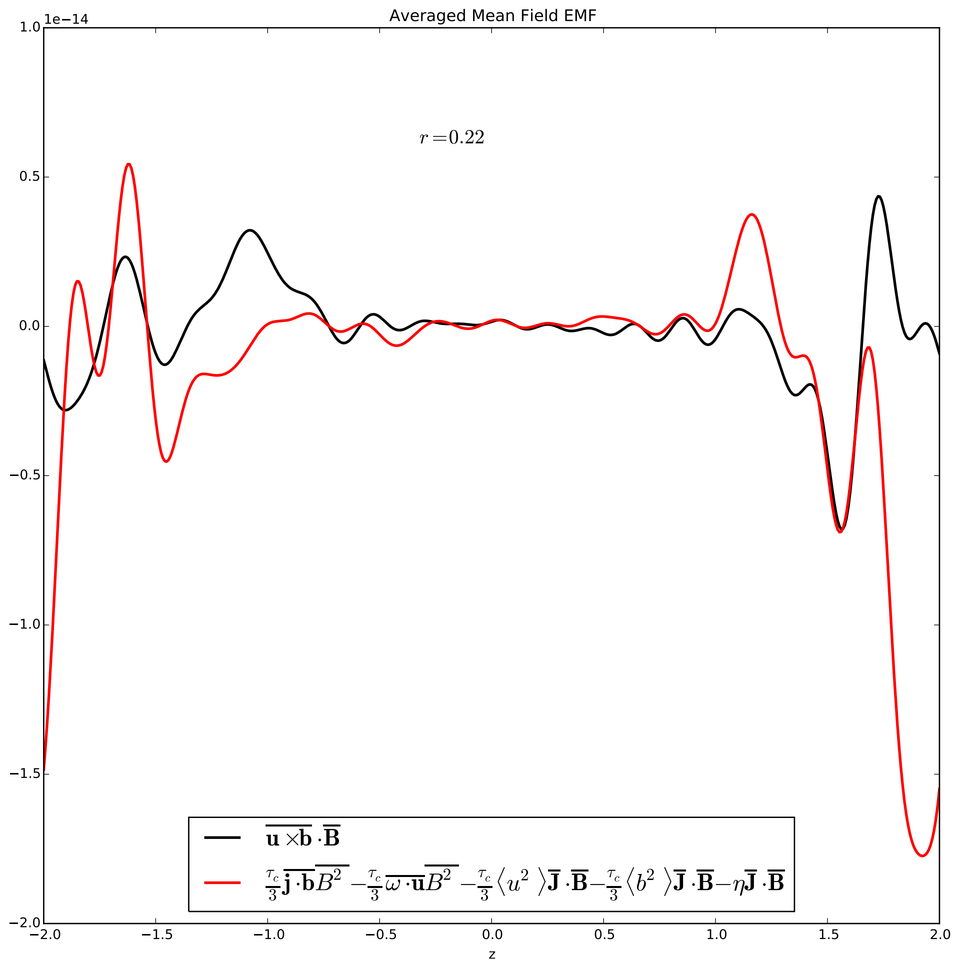


Figure 4.8: Here we present the time averaged mean field model including the magnetic α effect. We now show a positive correlation between the measured EMF and the model predictions suggesting the magnetic α is the dominant contribution to the dynamo.

With the inclusion of the magnetic α term, we now see a positive weak correlation. The strength of the correlation may be, in part due to the effects seen in Section 4.2. Another possibility might be that the assumption of isotropy is invalid and so we explore this idea in the next section.

4.2.2 Anisotropy

Up to this point, the mean field model makes the assumption that the turbulence is isotropic, but we know that for the EMF to be non-zero it must be at least weakly anisotropic. We will therefore drop this assumption and focus on the azimuthal direction. First we explain where the turbulent dissipation term that depends on $\overline{b^2}$ comes from. If we take the time derivative of the EMF and plug in the definition of the small scale momentum equation into $\overline{\dot{\boldsymbol{u}} \times \boldsymbol{b}}$, there is a term which is $\overline{(\boldsymbol{b} \cdot \boldsymbol{\nabla} \boldsymbol{B} - \nabla P) \times \boldsymbol{b}}$. By assuming that the components of the magnetic field are strongly correlated only with themselves, the first term becomes $-\epsilon_{ijk}\overline{b_j^2}\partial_j B_k$ or $\overline{b^2}\overline{\boldsymbol{J}}$. The second term can be simplified to be $2(k_i^2/k^2)b_j\partial_j B_i$ by assuming the small scale fluid is approximately incompressible. Putting these two terms together yields,

$$- \epsilon_{ijk}(1 - 2k_k^2/k^2)b_j^2\partial_j B_k. \tag{4.6}$$

Our expression for the azimuthal EMF now becomes

$$(\overline{\boldsymbol{u} \times \boldsymbol{b}})_y = \tau_c \left(\alpha_{yy}^M + \alpha_{yy}^K\right) \overline{B_y} - \tau_c \left(\overline{u_z^2} + C_x\overline{b_z^2}\right) \partial_z \overline{B_x} + \tau_c \left(\overline{u_x^2} + C_z\overline{b_x^2}\right) \partial_x \overline{B_z} - \eta \overline{J_y}, \tag{4.7}$$

where $\alpha_{yy}^M = 2\overline{b_x \partial_y b_z}$ and $\alpha_{yy}^K = -2\overline{u_x \partial_y u_z}$. Additionally, C is a function to describe the shape of eddies given by $1 - 2\frac{k_i^2}{k^2}$. In the isotropic limit C_0 reduces to a factor of $\frac{1}{3}$. For the anisotropic case we can estimate it by replacing k_i^2 with the components of the magnetic field that are perpendicular to it or $\overline{b_j^2}$ and $\overline{b_k^2}$. Because $\nabla \cdot \boldsymbol{B} = 0$, the term $\frac{k_z^2}{k^2}\overline{b_z^2}$ will be systematically smaller than the perpendicular components. As an estimate, parallel components will have a factor of 1/5 while perpendicular will have a factor of 2/5 in the isotropic limit. With these estimations we can construct an eddy shape factor as

$$C_x = \left(1 - 4\frac{\overline{b_y^2 b_x^2}}{\overline{b_z^2 b_y^2} + 2\overline{b_x^2 b_y^2} + 2\overline{b_y^2 b_z^2}}\right) \tag{4.8}$$

$$C_z = \left(1 - 4\frac{\overline{b_y^2 b_z^2}}{\overline{b_x^2 b_y^2} + 2\overline{b_x^2 b_z^2} + 2\overline{b_y^2 b_z^2}}\right). \tag{4.9}$$

The results for Equation (4.7) are given in Figure 4.9. We find that the azimuthal EMF is weakly correlated with the anisotropic mean field model for the whole data set and at some points is strongly correlated. This suggests that the anisotropy may not be playing a large role, and that the turbulence inherent in the EMF is the dominant source of error. This makes sense since the EMF is an instantaneous measurement, while the model is more of a temporal average.

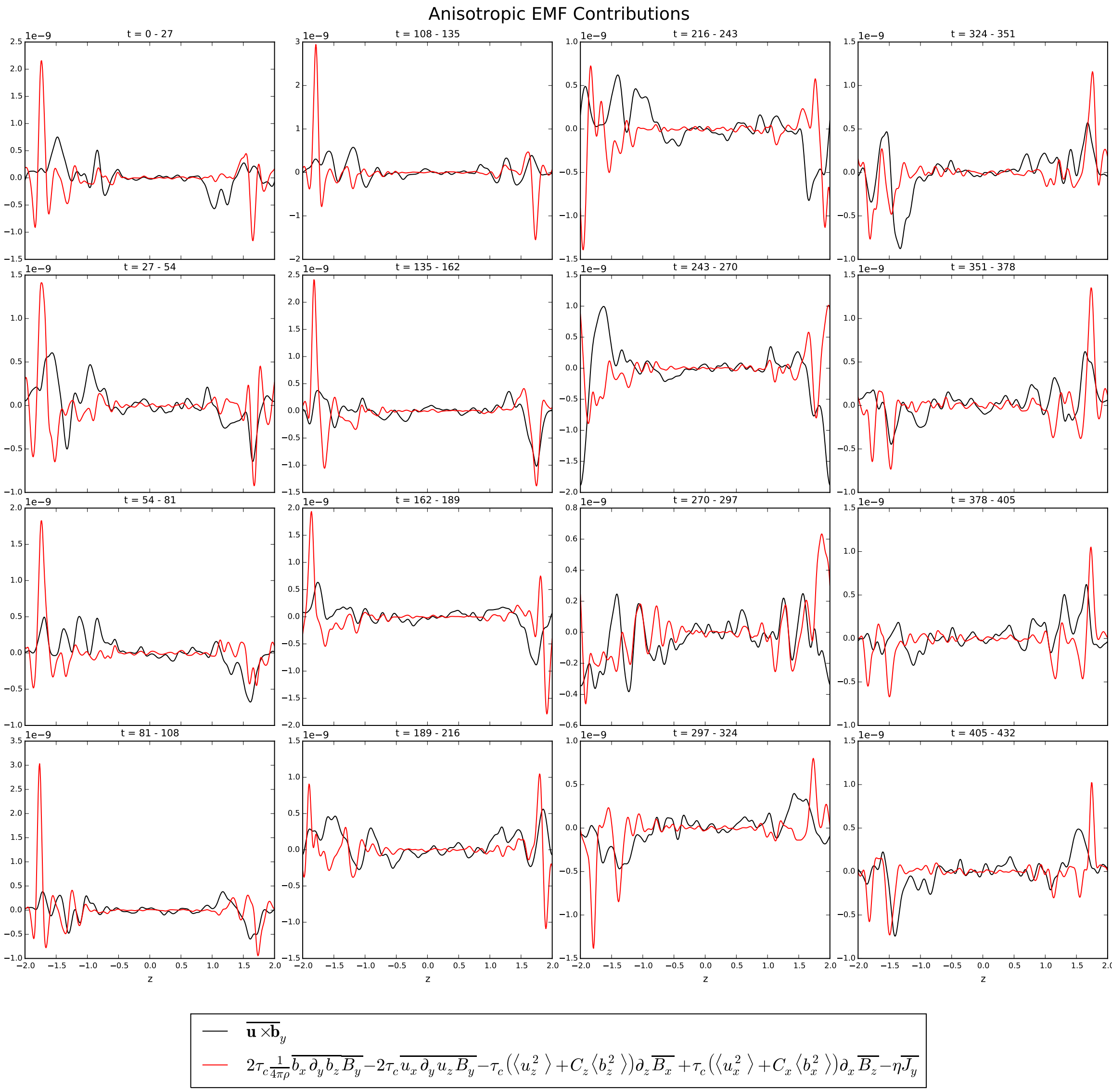


Figure 4.9: The calculated anisotropic EMF from the model overplotted with the EMF calculated small scale fields. Each panel is an average over 27 time slices comprising a total of 426 times. We can see a correlation between the model and the calculated EMF which we will quantify in the averaged plot.

If we take an average of all the calculation in an attempt to increase the signal to noise we get Figure 4.10. One reason for the weak correlation may be that we are missing additional terms, for instance the magnetic buoyancy or higher order terms in the EMF expansion. We look for these in the next section.

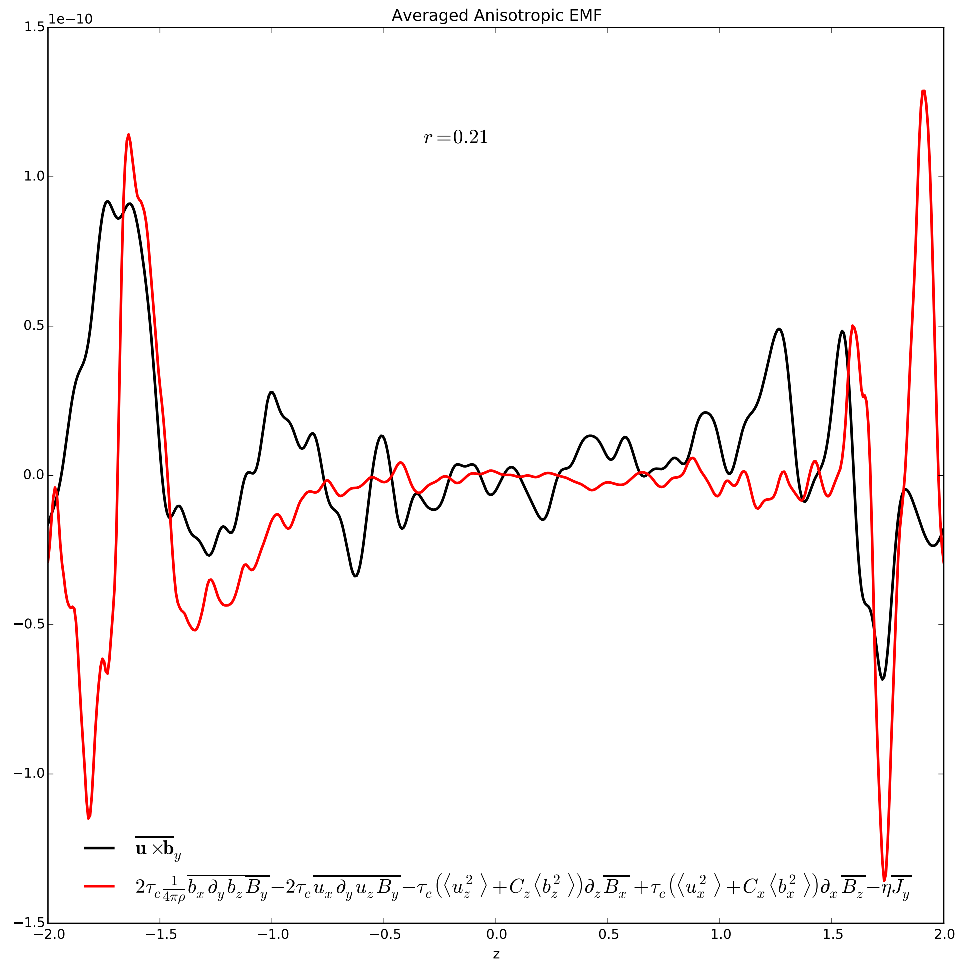


Figure 4.10: The calculated EMF overplotted with the values calculated using the model, averaged over all 426 time slices. A correlation exists between the direct calculation of the EMF and the model calculation.

4.2.3 Additional Contributions

Since the simulation performed is a stratified one, there is a potentially important contribution in the form of magnetic buoyancy. Sticking with our azimuthal view of the EMF, we can estimate this contribution as large scale vertical motions weighted by the magnetic pressure, or

$$\boldsymbol{\mathcal{E}}_{buoyancy} = \frac{\overline{U_z B^2 B_x}}{\overline{B^2}}. \tag{4.10}$$

In our calculation of the turbulent transport coefficients, there was a term which we associated with the Coriolis force, $2\boldsymbol{\Omega} \times \boldsymbol{u}$. This term, in principle, may have an effect on the dynamo. Indeed, Rogachevskii & Kleeorin (2003) have provided evidence for a "shear current" inducing an EMF. We make an additional term in our calculation of the EMF based on this shear current which works out to be

$$\kappa = \Omega \tau_c^2 \left(\frac{k_z^2 - k_y^2}{k^2} - \frac{3k_x^2 k_y^2}{k^4} \right) \overline{b_z^2} \partial_z \overline{B_y}. \tag{4.11}$$

If we assume that k_y^2/k^2 is small then this reduces to,

$$\kappa = \Omega \tau_c^2 \frac{k_z^2}{k^2} \overline{b_z^2} \partial_z \overline{B_y}. \tag{4.12}$$

The contributions to the EMF from buoyancy and the shear current are summarised in Figure 4.11. From this, we conclude that the effects of magnetic buoyancy and the shear current effect are negligible. The magnetic buoyancy may still play a role in the dynamo, however it does not appear to make a dominant contribution to the EMF. Similarly, the shear current effect may be important in other circumstances, however given the conditions of the present simulation, this effect is small. This is not entirely surprising, since the shear current effect is a higher order term.

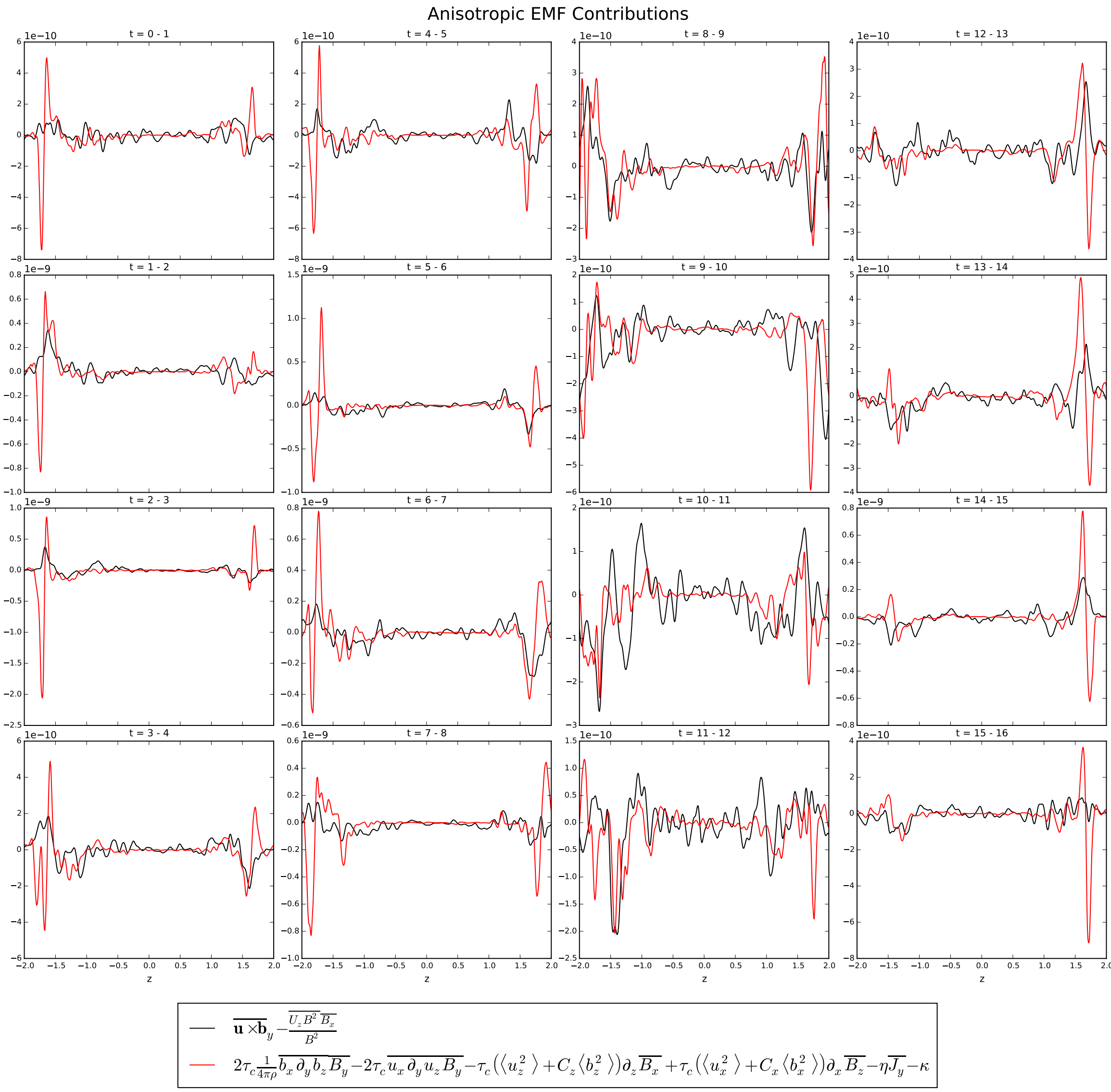


Figure 4.11: Calculated EMF with the model calculation, including buoyancy and shear current effects. We see that the inclusion of buoyancy and the shear current effect has marginally increased the correllation. We can conclude that during this stage of the dynamo it is safe to exclude the effects of magnetic buoyancy and the shear current effect.

The averaging process applied to this calculation is in Figure 4.12.

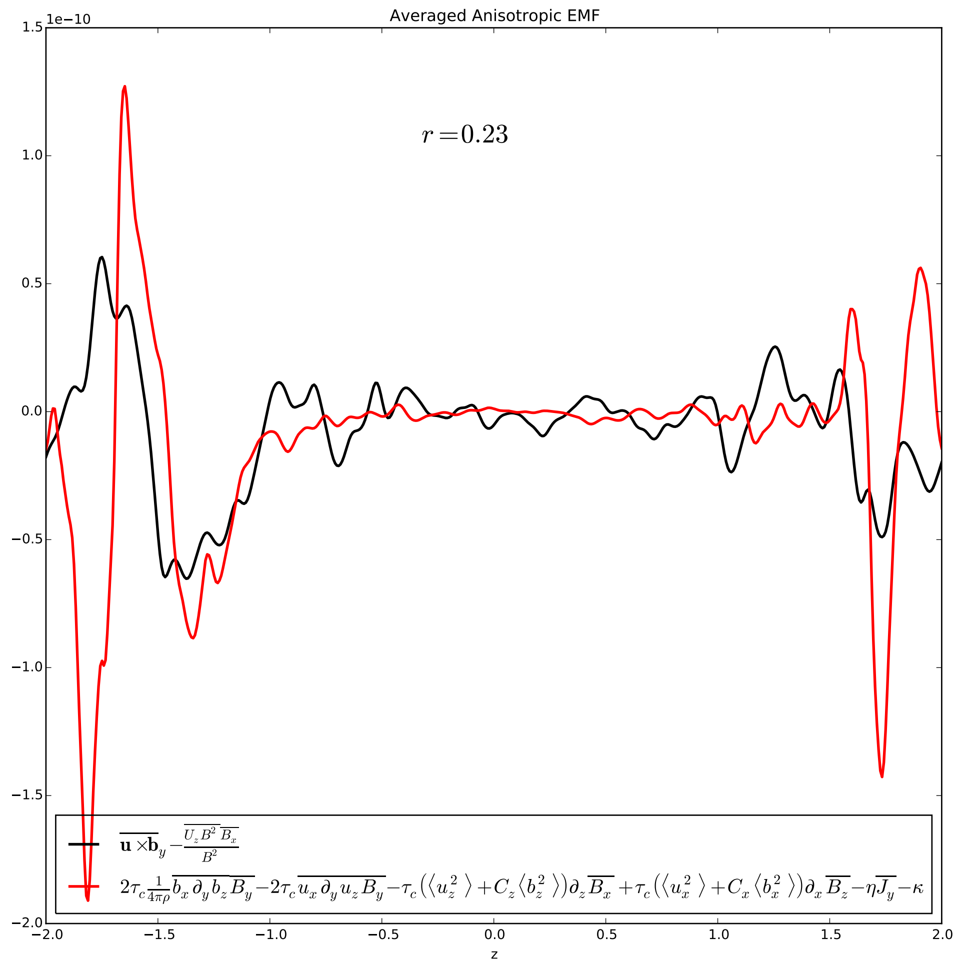


Figure 4.12: Average of the calculated EMF with the model values, again we see a marginal improvement in the correlation, however the value of the correlation is consistent with no effect.

4.2.4 Turbulent Correlation Timescale

In our discussion of the EMF we have neglected to mention anything about the turbulent correlation timescale, τ_c. We could leave it as a free parameter and then fit it based on the correlations, but to get a sense of its value we can get this information from the simulation itself. There are two methods we can use to calculate it, and there are several values for where it may apply.

In general, we approximate

$$\overline{\int_{-\infty}^{t} \boldsymbol{u}(\boldsymbol{x},t) \cdot \boldsymbol{u}(\boldsymbol{x},t')\,\mathrm{d}t'} = \overline{u^2}\tau, \tag{4.13}$$

and so we expect that the correlation between any quantity at times t and t' should decrease roughly exponentially as the time interval. The time constant of this exponential is described by τ and given functionally by $e^{-t/\tau}$.

The first method then, is to calculate the correlations for each quantity in the same location as a function of time, then fit an exponential. The second method calculates the correlations as a function of time, and then simply numerically integrate to get τ.

There are many quantities for which we could calculate this τ for, and each of them correspond to different processes. For instance, we expect there to be a difference between the correlation time calculated from either the small scale magnetic field or velocity and the large scale velocity. This difference should correspond to the turbulence driven by the MRI (small scale) and that of the dynamo (large scale). Additionally, there is no reason why the times should be exactly the same for each direction or for $\boldsymbol{b}$ and $\boldsymbol{u}$. We therefore calculate the turbulent correlation time for each scale in each direction using both methods. A summary of these calculations is found in Table 4.1. For reference, an example of the calculated correlations versus time is presented in Figure 4.13. The plots for the rest of the values can be found in Appendix A.1. A superimposed sinusoidal feature can be seen in the data, this could be a high order breathing mode like those discussed in Blaes et al. (2011) or related to the epicyclic frequency. In either case, they are unlikely to contribute to the dynamo since they are antisymmetric, which was tested by taking partial derivatives

	$\overline{\boldsymbol{U}}$	$\overline{U_x}$	$\overline{U_y}$	$\overline{U_z}$	$\overline{\boldsymbol{B}}$	$\overline{B_x}$	$\overline{B_y}$	$\overline{B_z}$
Fit: $\tau =$	620	771	664	372	*	680	*	480
Integration: $\tau =$	727	688	904	473	*	1039	*	532
	$\boldsymbol{u}$	u_x	u_y	u_z	$\boldsymbol{b}$	b_x	b_y	b_z
Fit: $\tau =$	200	210	190	200	190	170	210	130
Integration: $\tau =$	181	192	182	164	271	247	297	158

Table 4.1: Presented here are the correlation time scales calculated through fitting of an exponential and direct integration. The correlation times are given in units of $t\Omega^{-1}$ where t is defined from the first output as t_0 and incremented by the timestep used by the simulation. The correlations were calculated as follows: for each time slice we calculate $\frac{\langle \boldsymbol{X}(t_0)\cdot\boldsymbol{X}(t')\rangle}{\langle \boldsymbol{X}(t_0)^2\rangle}$ where the anlgle brackes represent a volume average and X is whatever quantity we are looking for, then either fit this quantitiy as a function of time or integrate it. The missing values for $\overline{\boldsymbol{B}}$ and B_y are due to the large scale dynamo cycles causing strong correlations and so no calculation was possible for these values.

and redoing the correlation procedure. The plots demonstrating this can be found in Appendix A.1.

In the analysis of the correlation times for the large scale magnetic field, the presence of dynamo cycles leads to correlations in the azimuthal and total magnetic fields. For this reason, we are unable to use this method to get correlation timescales for $\overline{B_y}$ and $\overline{\boldsymbol{B}}$. The other quantities are sufficiently decoupled from this cycle and so we get a good fit for the exponentials.

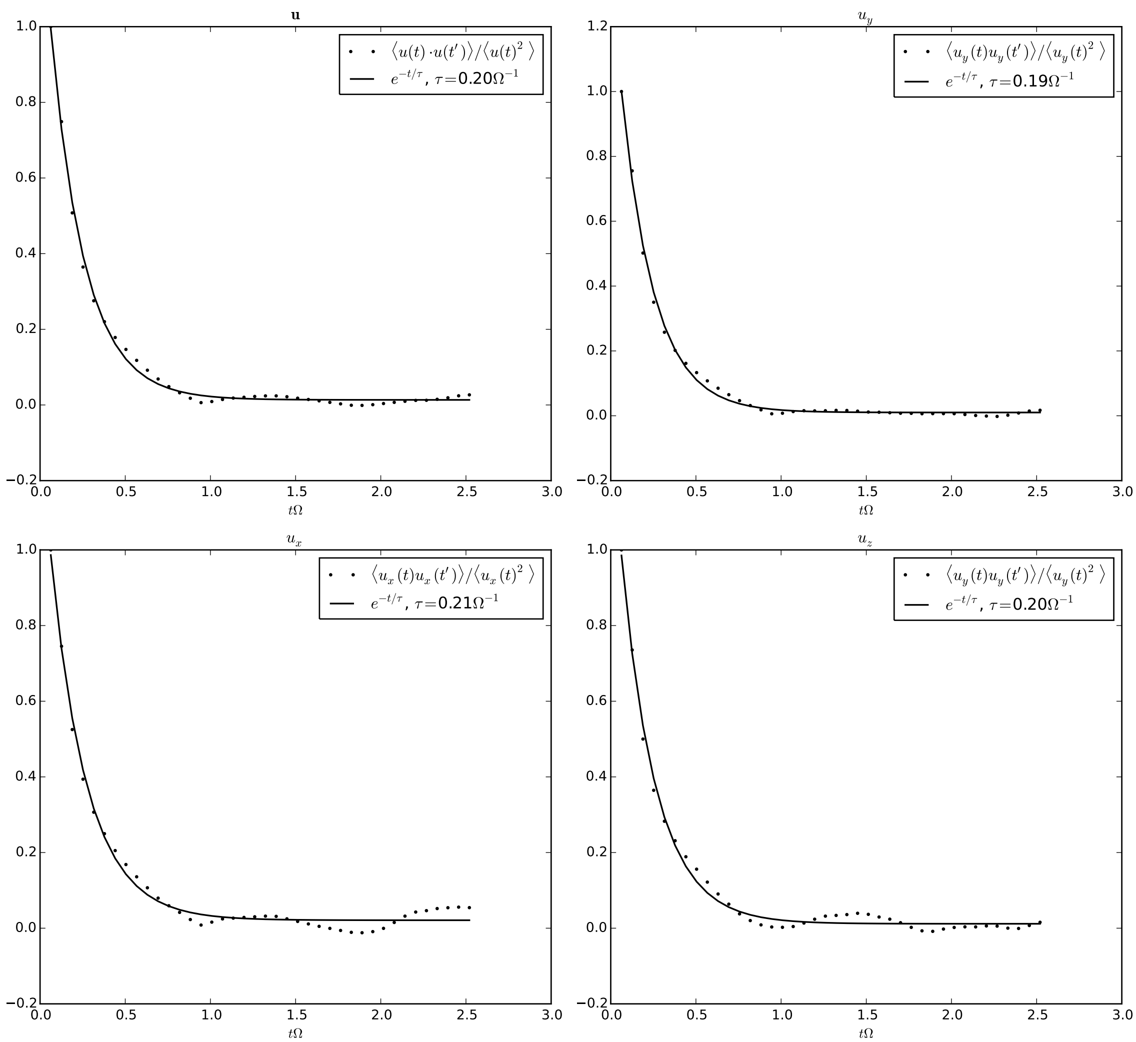


Figure 4.13: The correlation data for the turbulent velocity. The data are fit by an exponential of the form $Ae^{t/\tau} + C$, in all cases the values of A are close to 1 and the values of C are close to 0 as expected.

The correlation times found through this method are estimates of the Lagrangian turbulent correlation time. The τ quoted in the mean field models is the Eulerian timescale, so we should take these estimates as lower bounds to the actual correlation times. For this reason, we allow the value of τ to be a free parameter (within bounds of a factor of 2) while fitting the model predictions.

In many simulations of the MRI, the azimuthal field undergoes cycles. For reference, see Davis et al. (2010) where these cycles can be seen in their "butterfly" or space-time diagrams. The butterfly diagrams are characterised by correlated behaviour of the azimuthal field over long periods of time and so we expect to see this correlated behaviour while analysing the turbulent correlation timescale. Indeed, Figure 4.14 shows the correlation of the azimuthal field as a function of time. Our analysis suggests a period of approximately 9.5 orbits. To fully analyse the effects this cycle might have on our model, a full cycle should be analysed in the same way but due to computational constraints was not performed in this work.

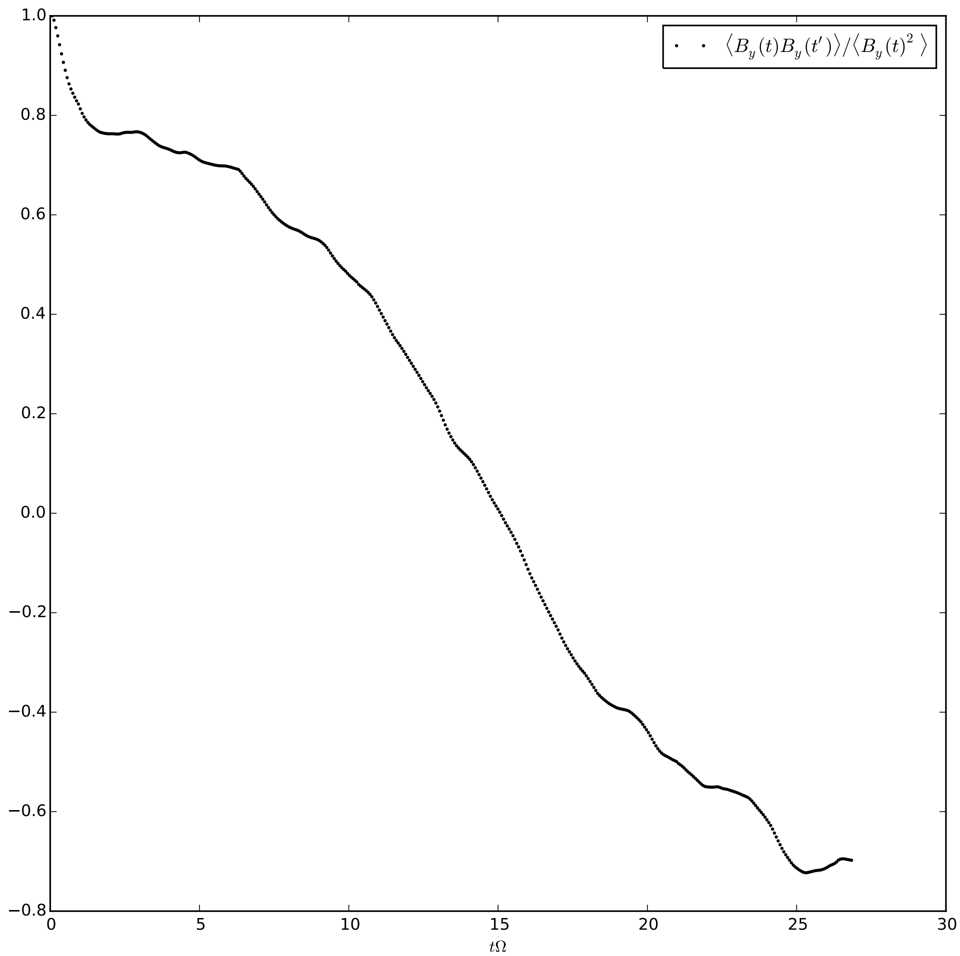


Figure 4.14: The correlation of $\overline{B_y}$ as a function of time. On short time scales, the exponential decay can be seen which is likely an artifact of small scale structures mixing into the filter. Afterwards, the correlation proceeds nearly linearly towards a complete anti-correlation (with a superimposed sinusoidal feature). We would expect to see this correlation turn around and go back to $\sim +1$ if we calculated the correlation for a whole dynamo cycle.

4.3 Code Property Tests

4.3.1 Magnetic Helicity Conservation

The flux of magnetic helicity plays a pivotal role in this model, but it relies heavily on the magnetic helicity being conserved (or approximately conserved). And so, we test this conservation here using the continuity equation,

$$\partial_t H + \nabla \cdot \boldsymbol{J_H} = -2\eta \boldsymbol{J} \cdot \boldsymbol{B} \tag{4.14}$$

We show the results of this test in Figure 4.15.

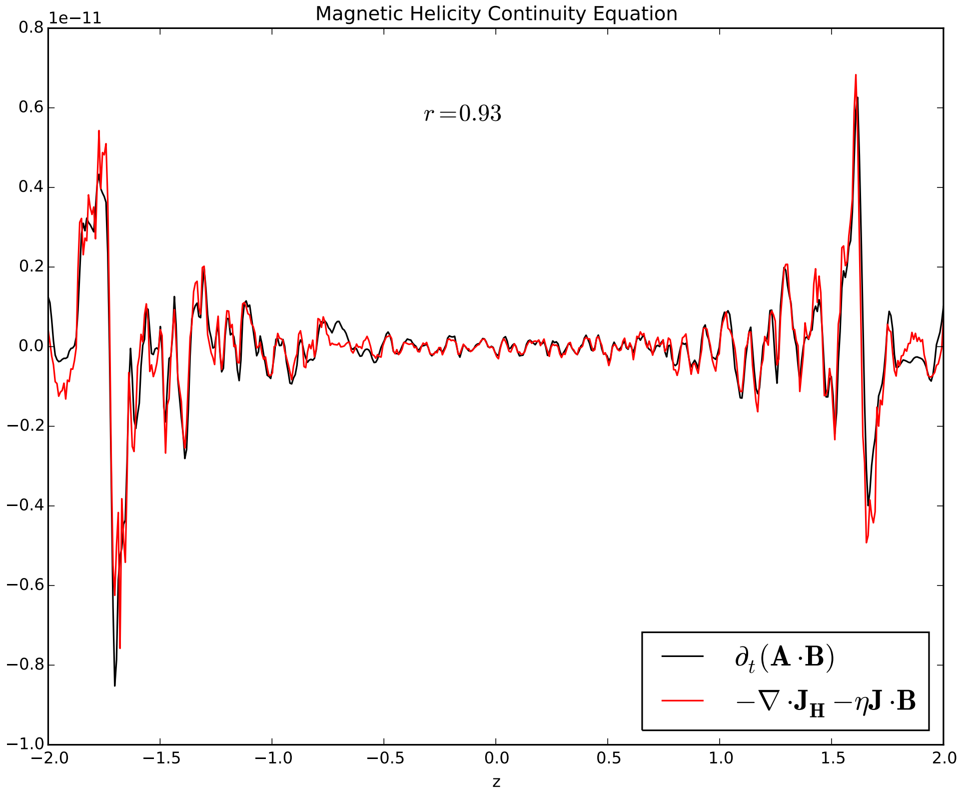


Figure 4.15: Shown here is the time derivative of the magnetic helicity, calculated numerically using finite difference. The red curve is the magnetic helicity flux and resistive term which were calculated from the fields and Equation (2.43). We see a good correlation which suggests the code is conserving magnetic helcity at an instantaneous point.

We can also test the conservation through use of the integral form of the magnetic helicity as a function of time. A plot of the value of the magnetic helicity as a function of time is given in Figure 4.16. There are some interesting features associated with this test, namely that we see that the largest amounts of magnetic helicity appearing about midway through our sample. This corresponds to a null point in the dynamo cycle, where the large scale azimuthal magnetic field is flipping signs. The reasoning for why we might see a accumulation of magnetic helicity during this time is due to the

MRI. The length scale where the MRI is operating has a minimum when the dynamo is at one of these null points. This places the minimum length of the MRI very close to the dissipation scale in k-space which will lead to a maximum of magnetic helicity non-conservation.

Additionally, this test shows a periodic signal in the production of magnetic helicity. This could be a result of a numerical artefact from the code being used, or it could be physical and related to some timescale in the simulation. A more detailed analysis of the magnetic helicity conservation is required to determine the precise cause of these cycles.

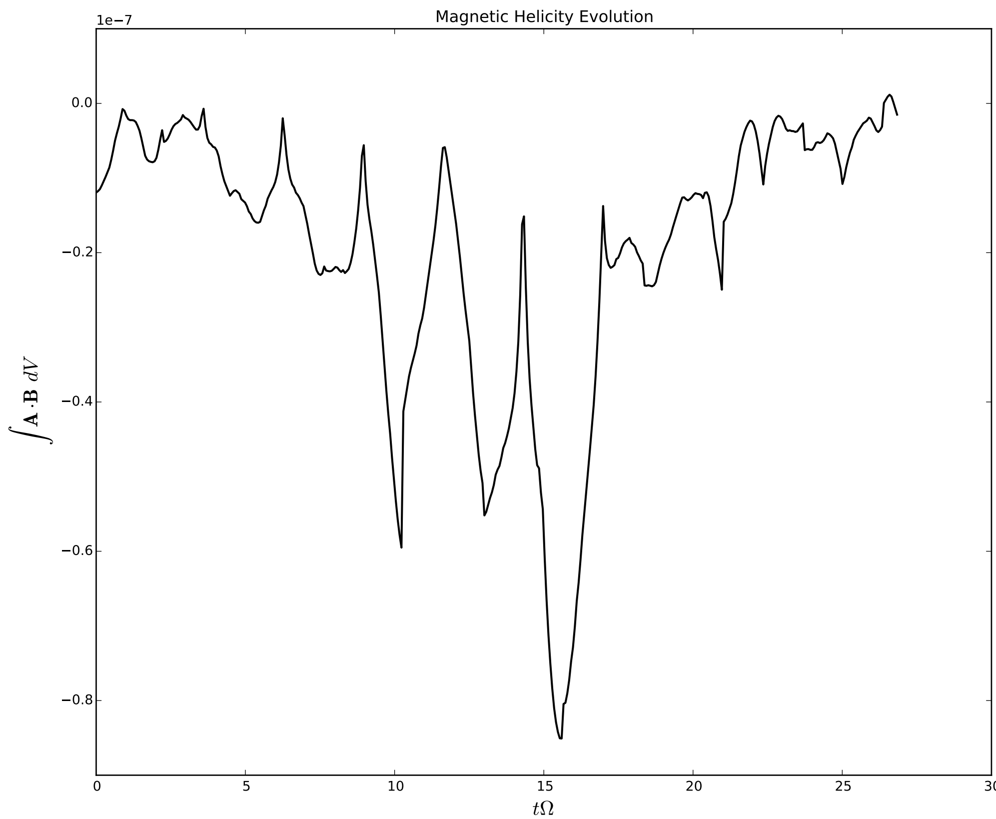


Figure 4.16: In this plot, we show the integrated magnetic helicity as a function of time (in units of $t\Omega$). We see that overall the magnetic helicity in the box stays roughly constant, which is consistent with this quantity being conserved. For reference, the RMS value of the magnetic helicity is $\sim 10^{-10}$ which means this production of magnetic helicity is not simply due to noise.

4.3.2 Current Helicity

To link the magnetic α effect with the magnetic helicity, we stated that $\overline{\boldsymbol{j} \cdot \boldsymbol{b}} \approx k_f^2 \overline{\boldsymbol{a} \cdot \boldsymbol{b}}$. We test this assumption here and attempt to measure the forcing scale, k_f. This result is shown in Figure 4.17. The strong correlation between the current and magnetic helicity is essential to be able to use the magnetic helicity flux to describe the dynamo.

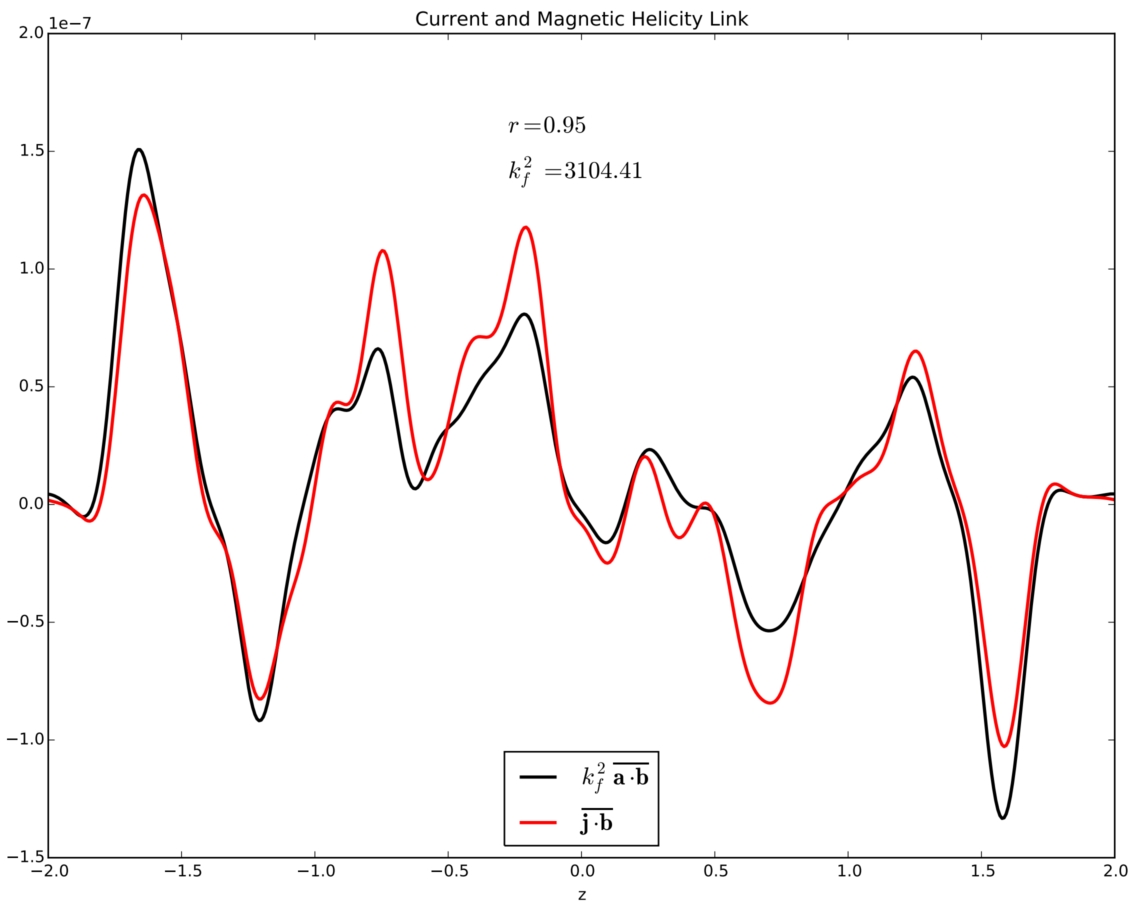


Figure 4.17: We overplot the current and magnetic helicities and fit a value for the force carrying scale. We see a strong correlation which justifies the link between the current helicity in the magnetic α term and the magnetic helicity flux. We can also estimate the force carrying scale by fitting the curves with k_f as a free parameter. Doing this yields a value of $k_f = 55$. Comparing this to the power spectrum in Figure 3.2 for the magnetic field, we find that this value of k_f is in good agreement with location of the peak due to the MRI.

4.4 Magnetic Helicity Flux

The connection between the current and magnetic helicities links the mean field alpha model with the dynamo being driven by the inverse cascade of magnetic helicity flux. The next step then, is to calculate the flux of magnetic helicity to establish its role in the dynamo process. The magnetic helicity flux is separated into large and small scale parts using the mean field approach. We have already seen the continuity equation for the total magnetic helicity flux, for completeness the same expression for the large scale components is given in Figure 4.18 and small scale components in Figure 4.19. It should be no surprise that these expressions match well, but it does show that there is not a complete agreement owing to the short term variations in the numerical time derivative, especially in the small scale.

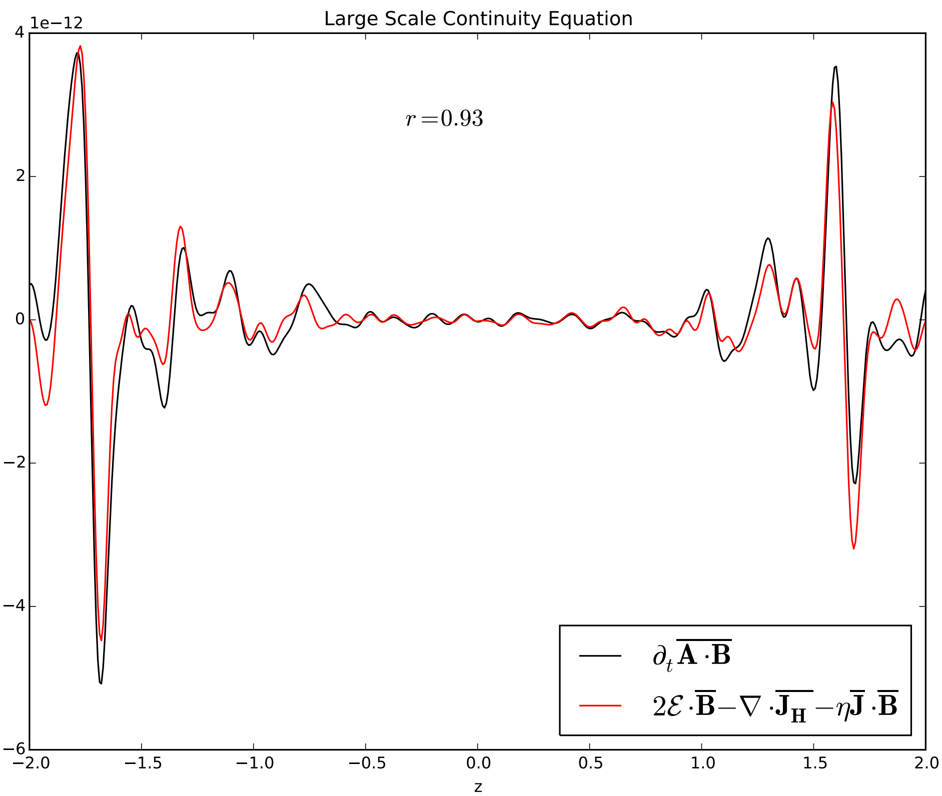


Figure 4.18: For the large scale magnetic helicity continuity equation, we see good agreement. It is noteworthy that this expression includes the term describing the transfer of magnetic helicity between scales, $\overline{\boldsymbol{u}\times\boldsymbol{b}}\cdot\overline{\boldsymbol{B}}$.

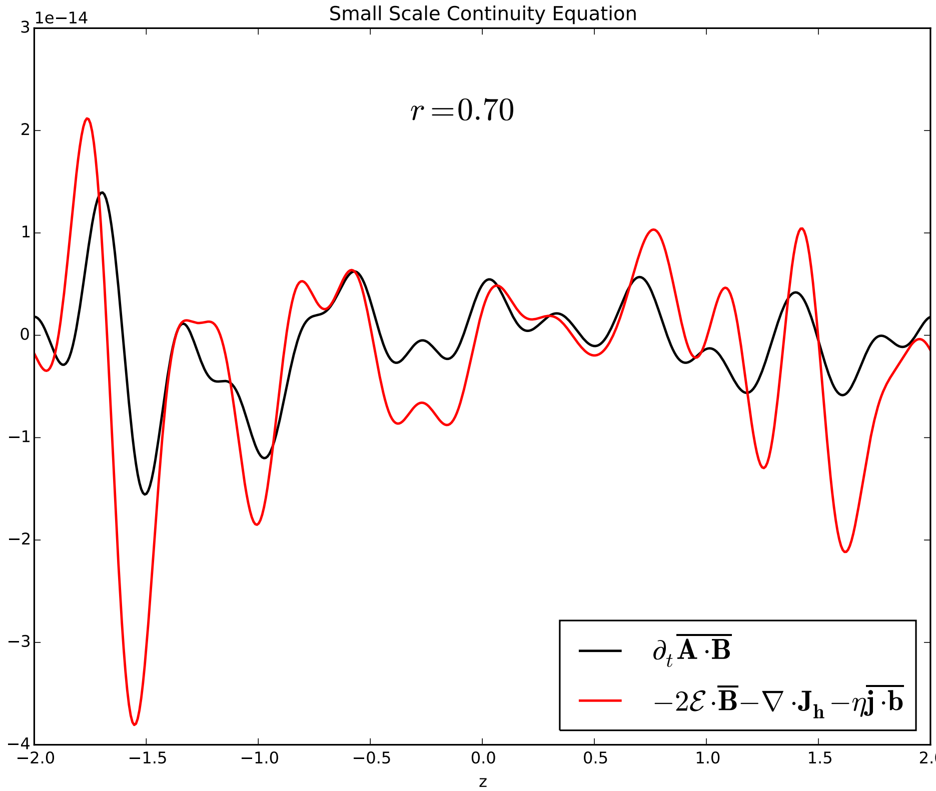


Figure 4.19: The small scale continuity equation contains more noise than the total and large scale components, though we still see a reasonable correlation here.

To make the link between the magnetic helicity flux and the dynamo, the transfer term, $-2\mathcal{E} \cdot \overline{\boldsymbol{B}}$ should be a good estimator of divergence of the small scale magnetic helicity flux. Figure 4.20 shows a moderate correlation where the errors can be attributed to the conservation of magnetic helicity not being perfect and a small amount of resistivity. This is a crucial step as it connects the discussion of the mean field α model with that of the magnetic helicity flux transport as a means of driving the dynamo.

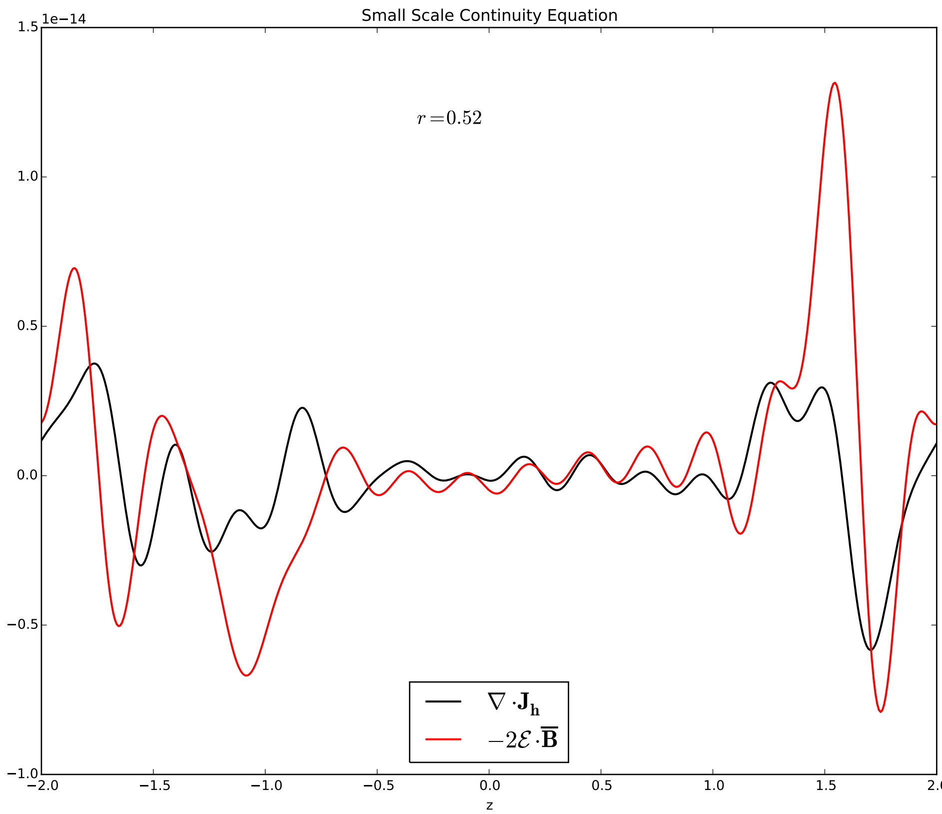


Figure 4.20: Here we link the component of the EMF parallel to the large scale magnetic field to the divergence of the small scale magnetic helicity flux. We see a moderate correlation ($r = 0.52$).

Following the same reasoning regarding the anisotropic nature of the turbulence, we can look at the vertical contribution to the divergence of the magnetic helicity flux and compare it to the azimuthal component of the EMF transfer term, this is plotted in Figure 4.21. Again we see a moderate correlation which helps confirm that anisotropy does not play a large role here and that the assumption of weak anisotropic turbulence is acceptable.

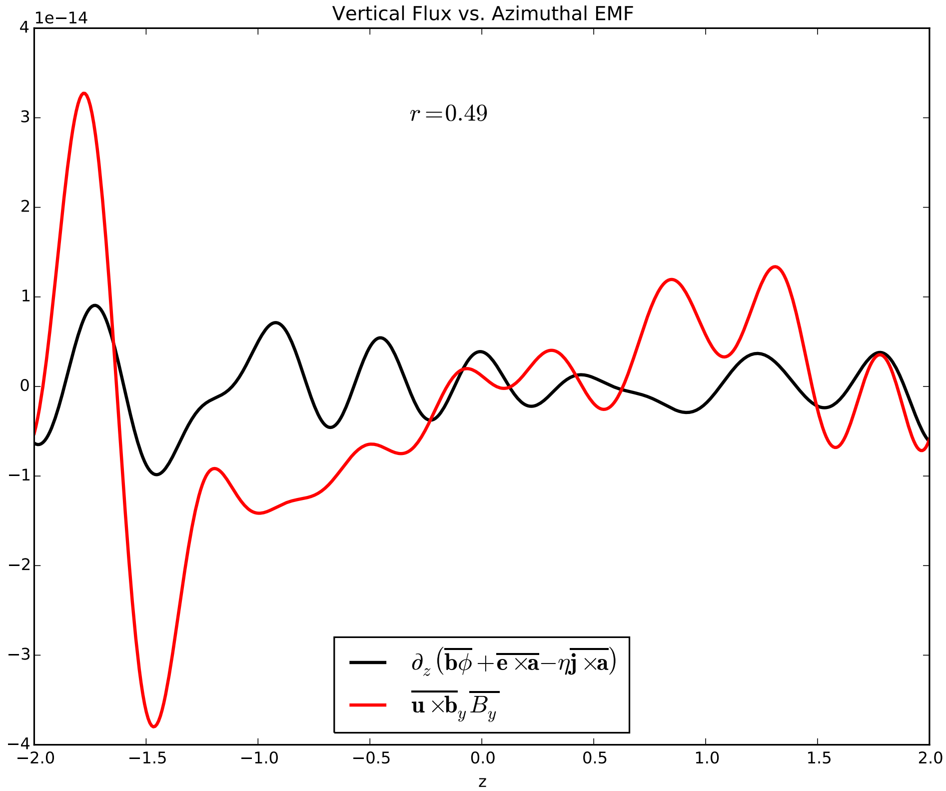


Figure 4.21: Here we show the link between the vertical transfer of magnetic helicity and the azimuthal component of the EMF times the large scale magnetic field.

Finally, to close off this chapter we will compare some of the results to those found in Vishniac & Shapovalov (2014). There, the authors performed a simulation in a shearing box geometry but with forced turbulence rather than turbulence driven by the MRI as in this simulation. An important result found was in equation (12) from Vishniac & Shapovalov (2014),

$$b_z\left(\phi_2 + \overline{\boldsymbol{U}} \cdot \boldsymbol{a}\right), \tag{4.15}$$

in relation to how much this term contributes to the divergence of the magnetic helicity flux. In Figure 4.22 we show a decrease of approximately 0.15 when correlating the expression for the divergence term with and without the term in Equation (4.15).

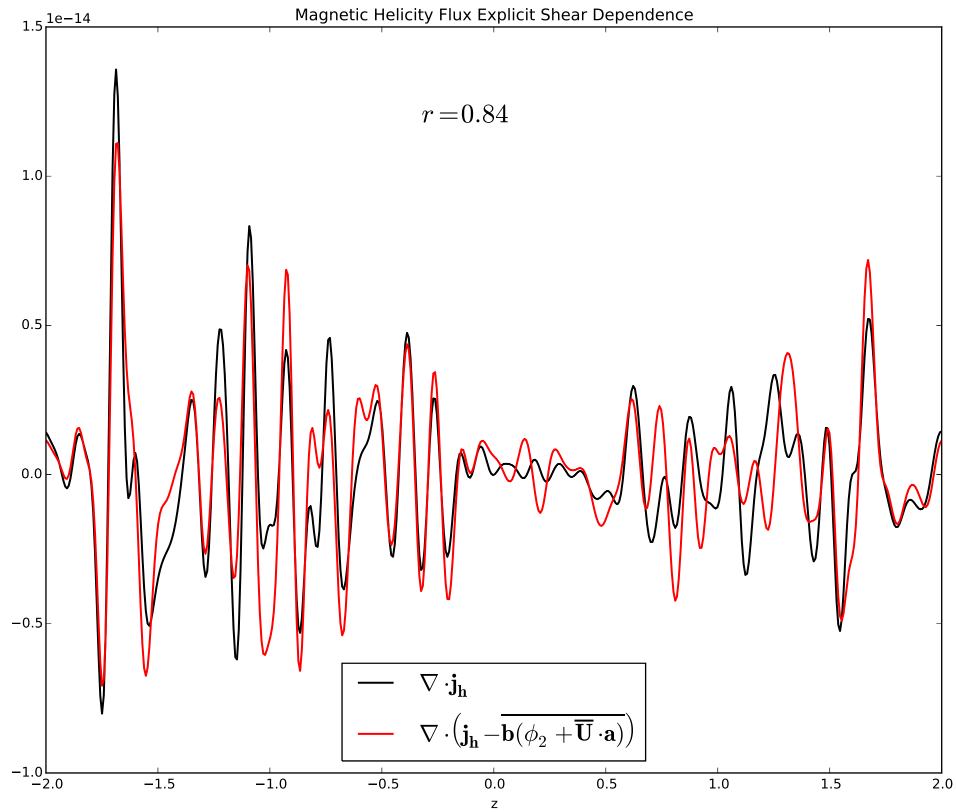


Figure 4.22: Here we show the divergence of the small scale magnetic helicity flux with and without the term which depends on the shear. We see that this term is approximately 15% of the total amplitude, consistent with the value seen in Vishniac & Shapovalov (2014).

Chapter 5

Conclusions

To finish this discussion we will present a summary of the results found during this work. First and foremost, we have established a link between the divergence of the magnetic helicity flux and the component of the EMF parallel to the large scale magnetic field in a local simulation driven by the MRI. In this way we have demonstrated that the transport of magnetic helicity flux between scales can drive a large scale dynamo in a realistic local simulation. Although this work is concerned only with a local simulation of an accretion disk dynamo, it provides the most detailed example to date of a magnetic helicity flux driven dynamo, a phenomena which we suspect forms the basis of large scale astrophysical dynamos everywhere. We note in particular that this approach removes the problem of alpha suppression, in which the small scale magnetic helicity poisons the large scale dynamo. Instead, the inverse cascade of the small scale magnetic helicity forms the basis of a successful dynamo.

In the process of demonstrating this connection, we have also verified the tools of mean field dynamo theory and the dynamically quenched α dynamo. In addition, we have provided a method in which to estimate the Lagrangian turbulent correlation timescale for the large and small scale magnetic and velocity fields. Included in this is a more detailed estimate of the correlation timescale by doing the same calculation as a function of the vertical coordinate.

The simulation used in this work was analysed at a state where the dynamo was saturated. Recently Vishniac & Shapovalov (2014) has seen in simulations using forced turbulence that the mean field α prediction and the transfer of magnetic helicity flux is most correlated when the dynamo is first growing from the initial seed field to saturation. During the saturated stage we correspondingly expect that the correlations will be weaker. Further investigation into the pre-equipartition stage of the dynamo is needed to test this hypothesis, however.

The mean field approach relies on the separation of the problem into two scales and it is always possible to separate scales for a k less than the dimensions of the box. In our case we separate scales and project onto the vertical direction, which is to say we do a horizontal average. If the separation between scales is small compared to the box size, then this projection will be noisy. Additionally, the time series will also show this noise, Brandenburg & Subramanian (2005). Our choice of cut-off frequency, and thus the quality of the scale separation are set by the driving scale of the turbulence (in this case the MRI) and the resolution of the simulation. The forcing scale is more or less fixed, however further work could involve a higher resolution simulation to allow more small scale structure. Additionally, the tool to separate scales uses Fourier filtering which introduces some of the problems discussed in Section 3.0.7. A more modern approach is to use a multi-scale decomposition using wavelets. The main advantage would be the ability to localise in space as well as scale, as opposed to just scale.

Our results of the previous chapter tend to show large amplitudes near the vertical extents of the box which can skew some of the conclusions. A possible explanation might be due to stratification, since the density in these regions is very low ($< 10\%$

of the midplane value), and the presence of large gradients in the magnetic and velocity fields. The gravitational potential smoothly flipping sign in this region may be important as well, though this aspect is unclear.

The validation of the mean field approach along with the importance of the transfer of magnetic helicity flux allow us to estimate the value of the α parameter from the large scale, or mean fields. The mean fields are the values that we can observe in a physical system, or the elements corresponding to the smallest resolution in a global accretion disk simulation. We can then estimate a value of α at any point in the disk without needing to resolve the turbulent scale.

With a better understanding of the accretion process, the MHD turbulence driven by the MRI, and the dynamo mechanisms we stand a good chance of being able to describe some of the more mysterious phenomenon such as outflows, twinkling and jets.

Bibliography

Balbus, S. A. & Hawley, J. F. 1991, ApJ, 376, 214

Blackman, E. G. & Field, G. B. 1999, ApJ, 521, 597

Blaes, O., Krolik, J. H., Hirose, S., & Shabaltas, N. 2011, ApJ, 733, 110

Blandford, R. D. & Payne, D. G. 1982, MNRAS, 199, 883

Bonnor, W. B. 1956, MNRAS, 116, 351

Brandenburg, A., Nordlund, A., Stein, R. F., & Torkelsson, U. 1995, ApJ, 446, 741

Brandenburg, A. & Subramanian, K. 2005, Phys. Rep., 417, 1

Cattaneo, F. & Hughes, D. W. 1996, Physical Review E, 54, 4532

Chandrasekhar, S. 1960, Proceedings of the National Academy of Science, 46, 253

Cooley, J. W. & Tukey, J. W. 1965, Math. Comput., 19, 297

Davis, S. W., Stone, J. M., & Pessah, M. E. 2010, ApJ, 713, 52

Deardorff, J. W. 1970, Journal of Fluid Mechanics, 41, 453

Donati, J.-F., Paletou, F., Bouvier, J., & Ferreira, J. 2005, Nature, 438, 466

Ebert, R. 1955, ZAp, 37, 217

Emden, R. 1907, in Gaskugeln - Anwendungen der Mechan. Warmtheorie. (Druck und Verlag Von B. G. Teubner, Leipzig), 129

Fromang, S. & Papaloizou, J. 2007, A&A, 476, 1113

Gardiner, T. A. & Stone, J. M. 2005, Journal of Computational Physics, 205, 509

Gruzinov, A. V. & Diamond, P. H. 1994, Physical Review Letters, 72, 1651

Hawley, J. F., Gammie, C. F., & Balbus, S. A. 1995, ApJ, 440, 742

Horne, K. 1995, A&A, 297, 273

Käpylä, P. J. & Korpi, M. J. 2011, MNRAS, 413, 901

Krause, F. & Raedler, K.-H. 1980, Mean-field magnetohydrodynamics and dynamo theory

Kulsrud, R. M. & Anderson, S. W. 1992, ApJ, 396, 606

Lane, J. H. 1870, The American Journal of Science and Arts, 2, 57

Lesur, G. & Longaretti, P.-Y. 2007, MNRAS, 378, 1471

Mitra, D., Candelaresi, S., Chatterjee, P., Tavakol, R., & Brandenburg, A. 2010, Astronomische Nachrichten, 331, 130

Moffatt, H. K. 1978, Magnetic field generation in electrically conducting fluids, ed. Moffatt, H. K.

Monin, A. S. & I'Aglom, A. M. 1971, Statistical fluid mechanics; mechanics of turbulence

Parker, E. N. 1979, Cosmical magnetic fields: Their origin and their activity

Pessah, M. E., Chan, C., & Psaltis, D. 2007, ApJ, 668, L51

Pouquet, A., Frisch, U., & Leorat, J. 1976, Journal of Fluid Mechanics, 77, 321

Reynolds, O. 1895, Philosophical Transactions of the Royal Society of London, 186, 123

Rogachevskii, I. & Kleeorin, N. 2003, Physical Review E, 68, 036301

Shakura, N. I. & Sunyaev, R. A. 1973, A&A, 24, 337

Shi, J., Krolik, J. H., & Hirose, S. 2010, ApJ, 708, 1716

Spitzer, L. 1962, Physics of Fully Ionized Gases

Stone, J. M. & Gardiner, T. A. 2010, ApJS, 189, 142

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, ApJS, 178, 137

Stone, J. M., Hawley, J. F., Gammie, C. F., & Balbus, S. A. 1996, ApJ, 463, 656

Vainshtein, S. I. & Cattaneo, F. 1992, ApJ, 393, 165

Velikhov, E. P. 1959, Journal of Theoretical and Experimental Physics (U.S.S.R), 36, 1398

Vishniac, E. T. 2009, ApJ, 696, 1021

Vishniac, E. T. & Cho, J. 2001, ApJ, 550, 752

Vishniac, E. T. & Shapovalov, D. 2014, ApJ, 780, 144

Appendix A

Appendix A

A.1 Turbulent Correlation Times

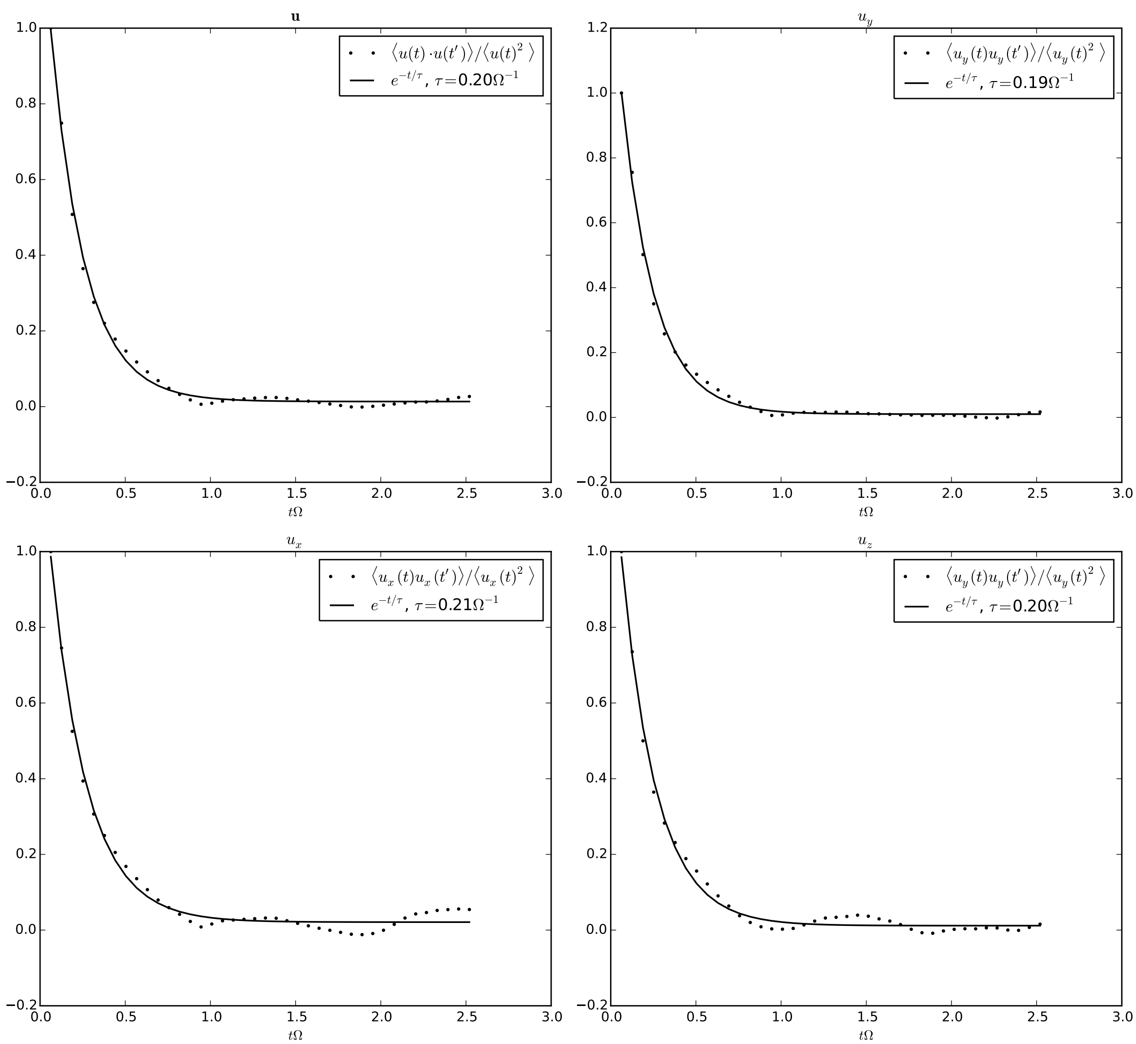

u
$\langle u(t) \cdot u(t')\rangle / \langle u(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.20Ω^{-1}
u_y
$\langle u_y(t) u_y(t')\rangle / \langle u_y(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.19Ω^{-1}
u_x
$\langle u_x(t) u_x(t')\rangle / \langle u_x(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.21Ω^{-1}
u_z
$\langle u_y(t) u_y(t')\rangle / \langle u_y(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.20Ω^{-1}
$t\Omega$

Figure A.1

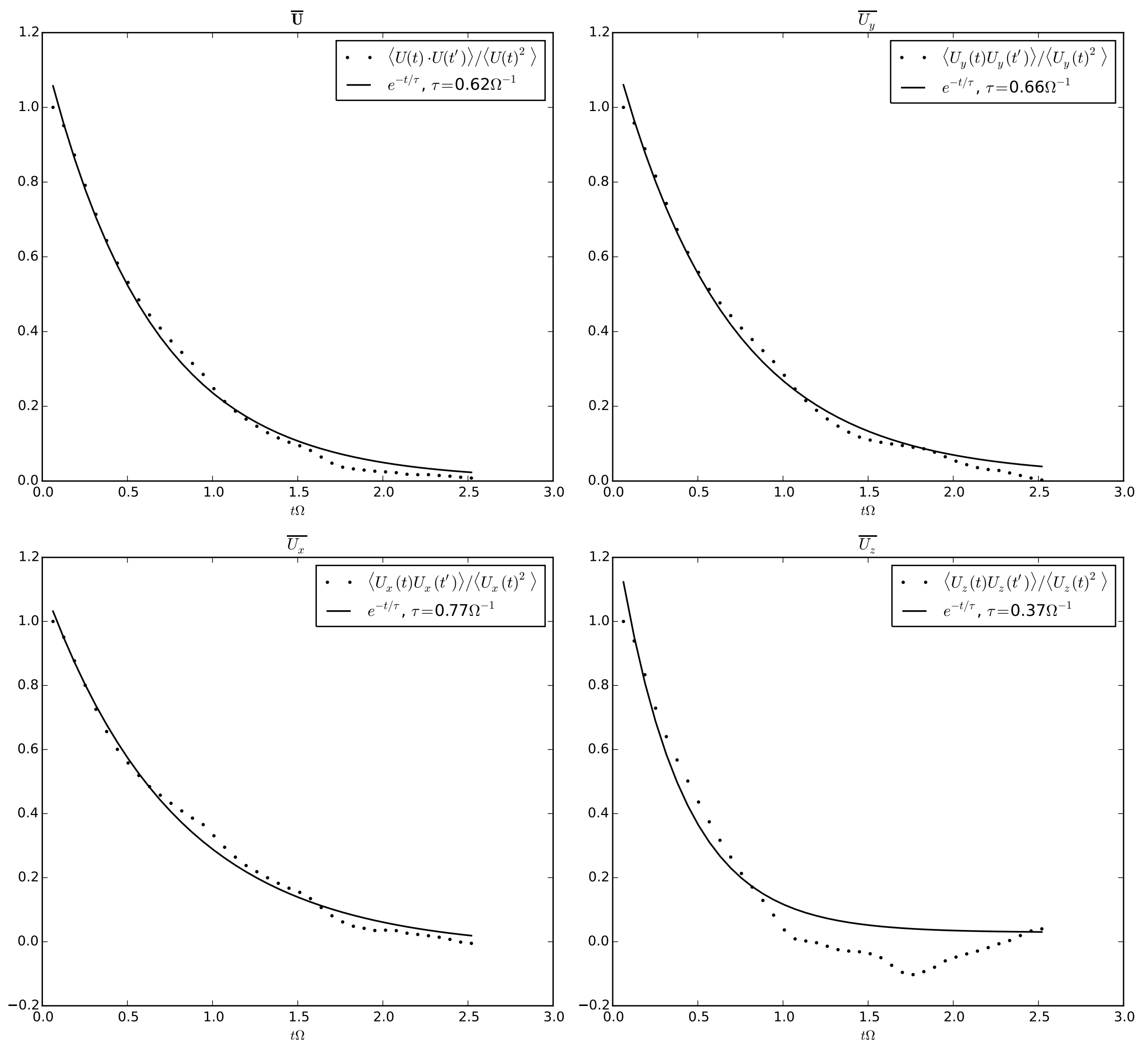
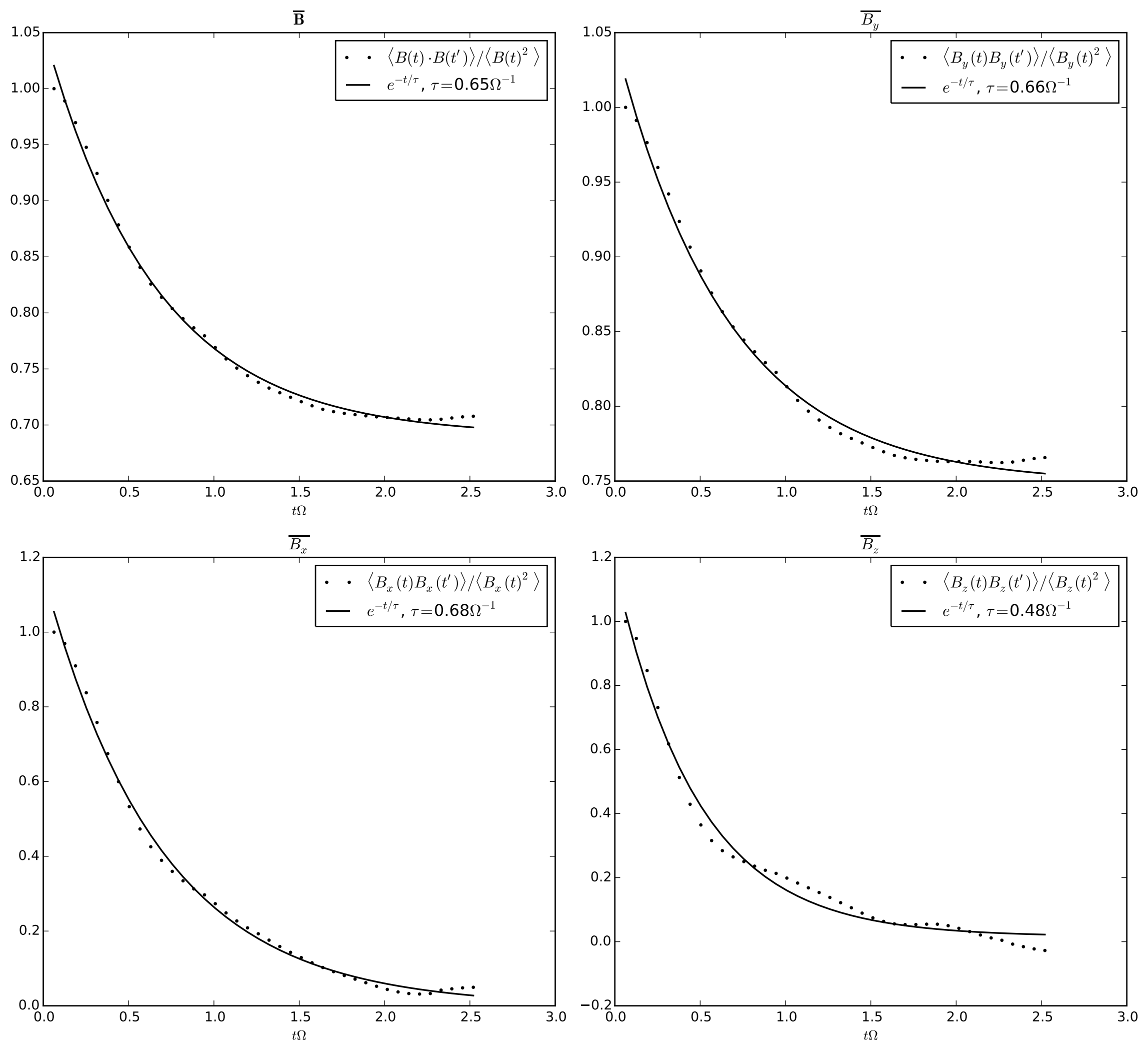

$\overline{\mathbf{U}}$
$\langle U(t) \cdot U(t') \rangle / \langle U(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.62Ω^{-1}
$\overline{U_y}$
$\langle U_y(t) U_y(t') \rangle / \langle U_y(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.66Ω^{-1}
$\overline{U_x}$
$\langle U_x(t) U_x(t') \rangle / \langle U_x(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.77Ω^{-1}
$\overline{U_z}$
$\langle U_z(t) U_z(t') \rangle / \langle U_z(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.37Ω^{-1}
$t\Omega$

Figure A.2



$\overline{\mathbf{B}}$
$\langle B(t) \cdot B(t') \rangle / \langle B(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.65Ω^{-1}
$\overline{B_y}$
$\langle B_y(t) B_y(t') \rangle / \langle B_y(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.66Ω^{-1}
$\overline{B_x}$
$\langle B_x(t) B_x(t') \rangle / \langle B_x(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.68Ω^{-1}
$\overline{B_z}$
$\langle B_z(t) B_z(t') \rangle / \langle B_z(t)^2 \rangle$
$e^{-t/\tau}$, τ=0.48Ω^{-1}
$t\Omega$

Figure A.3

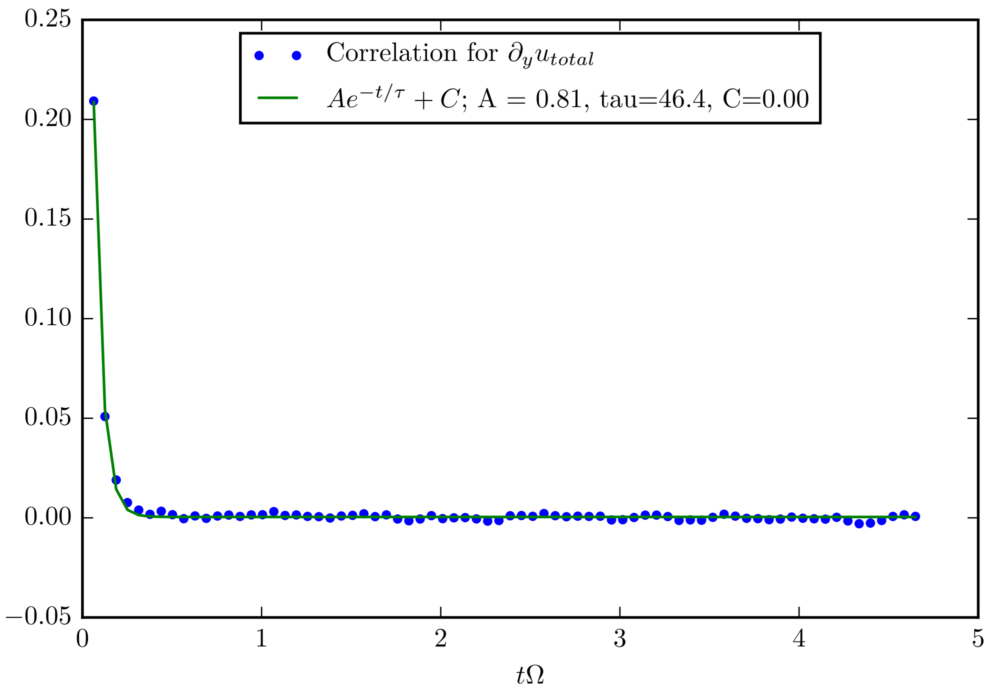

Correlation for $\partial_y u_{total}$
$Ae^{-t/\tau}+C$; A = 0.81, tau=46.4, C=0.00
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0
1
2
3
4
5
$t\Omega$

Figure A.4

A.2 Analysis Code

```
#Define some constants:
cs = 5.e-7
eta = 5.8e-8

#Generation Functions:

def Generate_Time_Derivatives(fs):
    """
    Generates time derivatives for:
    U, B, u, b, EMF, H, h

    f: is the path to the vtk files
    """
    dudt = zeros((len(files),3,N[-1]),dtype=float32)
    dbdt = zeros_like(dudt)
    dUdt = zeros_like(dudt)
    dBdt = zeros_like(dudt)
    dEdt = zeros_like(dudt)

    uav = zeros_like(dudt)
    bav = zeros_like(dudt)
    Uav = zeros_like(dudt)
    Bav = zeros_like(dudt)
    Eav = zeros_like(dudt)

    Ht = zeros((len(files),N[-1]),dtype=float32)
    H = zeros_like(Ht)
    h = zeros_like(Ht)
    dHt = zeros_like(H)
    dH = zeros_like(H)
    dh = zeros_like(H)
    for i in range(len(fs)):
        N,D,B,U,t = read_data(files[i],time_out=True)
        b = highpass(B)
        u = highpass(U)
        tm[i] = t
        #make use of the fact that the average of the
           derivative is the same as the derivative of the
           average
        uav[i] = av(u)
        bav[i] = av(b)
```

```
        Uav[i] = av(U)
        Bav[i] = av(B)

        Eav[i] = av(LP(cross(u,b)))

        #Current Density
        J = curl(B)
        #Vector Potential
        A = fft(-J)/k2
        A[:,0,0,0] = 0.
        A = ifft(A)
        a = highpass(A)
        Ht[i] = av(dot(A,B))
        H[i] = av(dot(LP(A),LP(B)))
        h[i] = av(LP(dot(a,b)))

        if(i > 0):
            #Take the numerical derivatives
            dt = tm[i]-tm[i-1]
            dudt[i] = (uav[i]-uav[i-1])/dt
            dbdt[i] = (bav[i]-bav[i-1])/dt
            dUdt[i] = (Uav[i]-Uav[i-1])/dt
            dBdt[i] = (Bav[i]-Bav[i-1])/dt
            dEdt[i] = (Eav[i]-Eav[i-1])/dt
            dHt[i] = (Ht[i]-Ht[i-1])/dt
            dH[i] = (H[i] - H[i-1])/dt
            dh[i] = (h[i]-h[i-1])/dt
    return dudt, dbdt, dUdt, dBdt, dEdt, dHt, dH, dh

def Generate_MFT_EMF(f):
    """
    Generates the standard MFT EMF terms
    f is the path to the vtk file
    """
    N,D,B,U,rho,t = read_data(f,dens_inc=True,time_out=True)
    b = highpass(B)
    u = highpass(U)

    #EMF
    EMF = av(LP(cross(u,b)))

    #alpha_M term, 1/(12 pi rho) <j dot b> <B>
    alpha_M = av(LP(dot(curl(b),b))*LP(B)/3./rho)

    #alpha_K term, 1/3 <w dot u> <B>
```

```
    alpha_K = av(LP(dot(curl(u),u))*LP(B)/3.)

    #diffusion term for u, 1/3 <u^2> <J>
    diff_u = av(LP(dot(u,u))*LP(curl(B))/3.)

    #diffusion term for b, 1/3 <b^2> <J>
    diff_b = av(LP(dot(b,b))*LP(curl(B))/3./rho)

    #resistive term, eta <J dot B>
    res =  av(eta*LP(curl(B)))

    return EMF, alpha_M, alpha_K, diff_u, diff_b, res

def anisotropic_EMF(f):
    """Takes path to vtk file as input, outputs the average
       anisotropic mean field emf terms
    """
    N,D,B,U,rho,t = read_data(f,dens_inc=True,time_out=True)
    b = delta(B)
    u = delta(U)

    #EMF, y-component
    emf = av(LP(cross(u,b)))[1]

    #alpha_M_yy term, 2 <b_x b_z,y> <B_y> / rho
    alpha_M_yy = 2.*av(LP(b[0]*ifft(1.j*k[1]*fft(b[2])))*LP(B
       [1])/rho)

    #alpha_K term, 2 <u_x u_z,y> <B_y>
    alpha_K_yy = 2.*av(LP(u[0]*ifft(1.j*k[1]*fft(u[2])))*LP(B
       [1]))

    #using the gaussian lowpass since it reduces the overshoot
    #and a smoother shape function
    bx2 = lowpass(b[0]*b[0], fc=2./128., alpha=0.1, wind='gauss
       ')
    by2 = lowpass(b[1]*b[1], fc=2./128., alpha=0.1, wind='gauss
       ')
    bz2 = lowpass(b[2]*b[2], fc=2./128., alpha=0.1, wind='gauss
       ')
    ux2 = lowpass(u[0]*u[0], fc=2./128., alpha=0.1, wind='gauss
       ')
    uy2 = lowpass(u[1]*u[1], fc=2./128., alpha=0.1, wind='gauss
       ')
```

```
    uz2 = lowpass(u[2]*u[2], fc=2./128., alpha=0.1, wind='gauss
       ')

    #Shape function for (1 - k_z^2 / k^2) <b_z^2>
    Cx = by2*bz2 / (bz2*by2 + 2*bz2*bx2 + 2*bx2*by2)
    Cz = by2*bx2 / (bx2*by2 + 2*bx2*bz2 + 2*bz2*by2)
    S = bx2*by2 / (2.*bz2*by2 + 2.*bz2*bx2 + bx2*by2)

    #diffusion term for u, <u_x^2> <B_z,x>
    diff_u_x = av(ux2*ifft(1.j*k[0]*LP(fft(B[2]))))

    #diffusion term for u, <u_z^2> <B_x,z>
    diff_u_z = av(uz2*ifft(1.j*k[2]*LP(fft(B[0]))))

    #diffusion term for b, <b_x^2> <B_z,x>
    diff_b_x = av((1.-4.*Cz)*bx2*ifft(1.j*k[0]*LP(fft(B[2])))/
       rho)

    #diffusion term for b, <b_z^2> <B_x,z>
    diff_b_z = av((1.-4.*Cx)*bz2*ifft(1.j*k[2]*LP(fft(B[0])))/
       rho)

    #resistive term, eta <J_y>
    res =  av(eta*curl(LP(B))[1])

    #shape function for the KR term
    #KR term, Omega tau^2 (s1+s2) <b_z^2> B_y,z
    KR = av(0.001*LP(S*bz2)*ifft(1.j*k[2]*LP(fft(B[1])))/rho)

    #buoyancy term, <U_z B^2> <B_x> / <B^2>
    E_buoy = av(LP(u[2] * sqrt(dot(B,B)))*LP(B[0]) / LP(sqrt(
       dot(B,B))))
    return emf, E_buoy, alpha_M_yy, alpha_K_yy, diff_u_x,
       diff_u_z, diff_b_x, diff_b_z, res, KR

def Generate_Magnetic_Helicity_Flux(f):
    N,D,B,U,t = read_data(files[i],time_out=True)
    b = highpass(B)
    u = highpass(U)

    J = curl(B)
    jc = highpass(J)

    #Vector potential, \nabla x A = B, \nabla^2 A = -J => A = J
       /k^2
```

```
    A = fft(J)/k2
    A[:,0,0,0] = 0.
    A = ifft(A)
    a = highpass(A)

    EMF = LP(cross(u,b))

    #Total flux = B Phi + A x (U x B) - eta (A x J)
    #E = -\nabla Phi, \nabla^2 Phi = -\nabla \cdot E => Phi = k
        \cdot E / k^2
    phi = ifft(dot(1.j*k, -fft(cross(U,B)))/k2)
    JHt = av(B*phi + cross(cross(U,B),A) - eta*cross(J,A))

    #Large Scale flux = <B> <Phi> + <A> x (<U> x <B> + EMF) -
       eta (<A> x <J>)
    phi = ifft(dot(1.j*k, -fft(cross(U-u,B-b) + EMF))/k2)
    JH = av(LP(B)*phi + cross(cross(LP(U),LP(B))+EMF, LP(A)) -
       eta*cross(LP(J), LP(A)))

    #Small scale flux = <b phi> + <a x e> - eta <a x j>
    e = cross(LP(U),b) + cross(u,LP(B)) + cross(u,b) - LP(cross
       (u,b))
    phi = dot(1.j*k, fft(-e))/k2
    phi[0,0,0]
    phi = ifft(phi)
    jh = LP( b*phi) + LP(cross(a, e)) - LP(eta*cross(a,jc) )

    #small scale resistive term
    res = 2.*av(eta*LP(dot(jc,b)))

    #Large scale resistive term
    Res = 2.*av(eta*dot(LP(J),LP(B)))

    #Total resistive term
    Rest = 2.*av(eta*dot(J,B))

    #Scale transfer term
    EdB = 2.*av(dot(EMF,LP(B)))

    return JHt, JH, jh, Rest, Res, res, EdB

#Correlation times:
def Generate_tau(fs):
    """Generates the correlations time for <U>, <B>, u, b
    """
```

```
corr_U = zeros(len(fs),dtype=float32)
corr_u = zeros(len(fs),dtype=float32)
corr_U_x = zeros(len(fs),dtype=float32)
corr_U_y = zeros(len(fs),dtype=float32)
corr_U_z = zeros(len(fs),dtype=float32)
corr_u_x = zeros(len(fs),dtype=float32)
corr_u_y = zeros(len(fs),dtype=float32)
corr_u_z = zeros(len(fs),dtype=float32)

corr_B = zeros(len(fs),dtype=float32)
corr_b = zeros(len(fs),dtype=float32)
corr_B_x = zeros(len(fs),dtype=float32)
corr_B_y = zeros(len(fs),dtype=float32)
corr_B_z = zeros(len(fs),dtype=float32)
corr_b_x = zeros(len(fs),dtype=float32)
corr_b_y = zeros(len(fs),dtype=float32)
corr_b_z = zeros(len(fs),dtype=float32)

#Generate the reference locations
N,D,Bt,Ut,t = read_data(fs[0],time_out=True)
t0 = t*1.
B0 = LP(Bt)
U0 = LP(Ut)
b0 = highpass(B0)
u0 = highpass(U0)

U2 = mean(dot(U0,U0))
B2 = mean(dot(B0,B0))

u2 = mean(dot(u0,u0))
b2 = mean(dot(b0,b0))

U2x = mean(U0[0]*U0[0])
U2y = mean(U0[1]*U0[1])
U2z = mean(U0[2]*U0[2])

B2x = mean(B0[0]*B0[0])
B2y = mean(B0[1]*B0[1])
B2z = mean(B0[2]*B0[2])

u2x = mean(u0[0]*u0[0])
u2y = mean(u0[1]*u0[1])
u2z = mean(u0[2]*u0[2])

b2x = mean(b0[0]*b0[0])
```

```
    b2y = mean(b0[1]*b0[1])
    b2z = mean(b0[2]*b0[2])

    for i in range(len(fs)):
        N,D,Bt,Ut,t = read_data(fs[i],time_out=True)
        B = LP(Bt)
        U = LP(Ut)
        b = highpass(B)
        u = highpass(U)
        #Correlation for large scale fields
        corr_U[i] = mean(dot(U0,U))/U2
        corr_U_x[i] = mean(U0[0]*U[0])/U2x
        corr_U_y[i] = mean(U0[1]*U[1])/U2y
        corr_U_z[i] = mean(U0[2]*U[2])/U2z

        corr_B[i] = mean(dot(B0,B))/B2
        corr_B_x[i] = mean(B0[0]*B[0])/B2x
        corr_B_y[i] = mean(B0[1]*B[1])/B2y
        corr_B_z[i] = mean(B0[2]*B[2])/B2z

        #Correlation for small scale fields
        corr_u[i] = mean(dot(u0,u))/u2
        corr_u_x[i] = mean(u0[0]*u[0])/u2x
        corr_u_y[i] = mean(u0[1]*u[1])/u2y
        corr_u_z[i] = mean(u0[2]*u[2])/u2z

        corr_b[i] = mean(dot(b0,b))/b2
        corr_b_x[i] = mean(b0[0]*b[0])/b2x
        corr_b_y[i] = mean(b0[1]*b[1])/b2y
        corr_b_z[i] = mean(b0[2]*b[2])/b2z


def Generate_tauz_corr(fs):
    """Generates the correlations as a function of z for each
       file from 0 to num
    """
    N,D,Bt,Ut,t = read_data(fs[0],time_out=True)
    t0 = t*1.
    u0 = highpass(Ut)
    b0 = highpass(Bt)
    u2 = av(dot(u0,u0))
    b2 = av(dot(b0,b0))
    del Ut, Bt

    c_u = zeros((num, N[-1]), dtype=float32)
```

```
    c_b = zeros((num, N[-1]), dtype=float32)

    for i in range(len(fs)):
        N,D,B,U,t = read_data(fs[i],time_out=True)
        u = highpass(U)
        b = highpass(B)
        c_u[i] = av(dot(u0,u))/u2
        c_b[i] = av(dot(b0,b))/b2
    return c_u, c_b

#Utility Functions:
def read_data(file_in,dens_inc=False, time_out = False):
    """
    Reads in data from file_in.vtk into N, D, b, u, (rho), (
       time)

    file_in:  Path to the location of the vtk file
    dens_inc: Flag to output the density, default = False
    time_out: Flag to output the time of the simulation (from
       ATHENA), default = False

    Variables are stored with vectors in the first column, e.g.
        B[0], B[1], B[2]
    The values at the (x,y,z) coordinates of the box are the
       next 3 columns, e.g. B[:, x, y, z]
    """
    import vtk as vtk
    from vtk.util.numpy_support import vtk_to_numpy
    from numpy import rollaxis, reshape, array, float32, zeros,
        linspace

    #Some vtk bookkeeping and object stuff
    reader = vtk.vtkStructuredPointsReader()
    reader.SetFileName(file_in)
    reader.ReadAllVectorsOn()
    reader.ReadAllScalarsOn()
    reader.Update()
    #Find where 'time=' is in the list, and take the next
       element and cut off the comma
    time = reader.GetHeader().split()
    try:
        time = float(time[time.index('time=')+1][:-1])
    except:
        time = float(time[time.index('=')+1])
    data = reader.GetOutput()
```

```
    # Dimensions of the box, store as a tuple
    dim = data.GetDimensions()
    nx = dim[0] - 1
    ny = dim[1] - 1
    nz = dim[2] - 1
    D = data.GetSpacing()
    N = (nx, ny, nz)

    ## #Read the raw data from the vtk file
    #Data comes out with vector in the transposed form
    try:
        u = vtk_to_numpy(data.GetCellData().GetArray('velocity'
           )).reshape(nz,ny,nx,3)
    except:
        u = vtk_to_numpy(data.GetCellData().GetArray('momentum'
           )).reshape(nz,ny,nx,3)

    b = vtk_to_numpy(data.GetCellData().GetArray('
       cell_centered_B')).reshape(nz, ny, nx,3)

    if(dens_inc):
        rho = vtk_to_numpy(data.GetCellData().GetArray('density
           ')).reshape(nz,ny,nx)
        if time_out:
            return (N,D,b.T,u.T,rho.T,time)
        else:
            return (N,D,b.T,u.T,rho.T)
    if(time_out):
        if dens_inc:
            rho = vtk_to_numpy(data.GetCellData().GetArray('
               density')).reshape(nz,ny,nx)
            return (N,D,b.T,u.T,rho.T,time)
        else:
            return (N,D,b.T,u.T,time)
    v_k = zeros((3,N[0],N[1],N[2]), dtype=float32)
    x=linspace(-.5,.5,N[0])
    for i in range(N[0]):
        v_k[1,i,:,:] += -1.5*0.001*x[i]

    return (N, D, b.T, u.T+v_k)

def define_k(N, D):
    """
    Creates the 3d k arrays from N and D
    To save space, each k vector is only length N
```

```
    K has dimensions of 3, with k[0] having shape (N[0],1,1)
       etc.
    """
    from numpy.fft import fftfreq
    from numpy import array, pi, float32
    nx, ny, nz = N
    dx, dy, dz = D
    #Use a matlab like ndgrid to create the wave-vectors with
       the proper shape to take advantage of pythons
       vectorization
    kx = fftfreq(nx,dx)*2.*pi
    ky = fftfreq(ny,dy)*2.*pi
    kz = fftfreq(nz,dz)*2.*pi

    #Output a single vector
    return array([kx[:,None,None], ky[None,:,None], kz[None,
       None,:]])

def fftvec(vec):
    """
    performs a fft on a vector with 3 components in the first
       index position
    This is really just a wrapper for fft, fftn and their
       inverses

    The code will properly detect scalar vs vector quantities
       and FFT appropriately. (As defined in read_data)
    """
    #Use the annfft library if possible for speed (Needs to be
       installed)
    try:
        from anfft import fft, fftn
        fft_type = 1
    except:
        #Use numpy with mkl instead, this is required,
           otherwise this should be modified to use plain numpy
            instead
        import mkl
        mkl.set_num_threads(8)
        from numpy.fft import fft, fftn
        fft_type = 0

    from numpy import complex64, shape, array, empty
    if vec.ndim > 2:
        if vec.shape[0] == 3:
```

```
            # "Vector": first index has size 3 so fft the other
                columns
            if fft_type==1:
                return array([fftn(i,measure=True) for i in vec
                   ]).astype(complex64)
            elif fft_type==0:
                return fftn(vec, axes=range(1,vec.ndim)).astype
                   (complex64)
            elif fft_type==2:
                result = empty(vec.shape, dtype=complex64)
                result[0] = gpu_fft(vec[0].copy())
                result[1] = gpu_fft(vec[1].copy())
                result[2] = gpu_fft(vec[2].copy())
                return result
        else: # "Scalar", fft the whole thing
            if fft_type==1:
                return fftn(vec,measure=True).astype(complex64)
            elif fft_type==0:
                return fftn(vec).astype(complex64)
            elif fft_type==2:
                return gpu_fft(vec.copy())
    elif vec.ndim == 1: #Not a vector, so use fft
        if fft_type==1:
            return fft(vec,measure = True).astype(complex64)
        elif fft_type==0:
            return fft(vec).astype(complex64)
        elif fft_type==2:
            return gpu_fft(vec.astype(complex64))
    else:
        #0th index is 3, so its a vector
        return array([fft(i) for i in vec])

def ifftvec(vec):
    """
    performs a fft on a vector with 3 components in the last
       index position
    This is a wrapper for ifft and ifftn, see fftvec
    """
    try:
        from anfft import ifft, ifftn
        fft_type = 1
    except:
        #Use numpy compiled agains mkl
        import mkl
        mkl.set_num_threads(8)
```

```
        from numpy.fft import ifft, ifftn
        fft_type = 0
    from numpy import float32, real, array, empty, complex64
    if vec.ndim > 2:
        if vec.shape[0] == 3:
            # "Vector": first index has size 3 so fft the other
                columns
            if fft_type==1:
                return array([ifftn(i,measure=True) for i in
                   vec]).astype(float32)
            elif fft_type==0:
                return ifftn(vec, axes=range(1,vec.ndim)).
                   astype(float32)
            elif fft_type==2:
                result = empty(vec.shape, dtype=float32)
                result[0] = gpu_ifft(vec[0].copy()).astype(
                   float32)
                result[1] = gpu_ifft(vec[1].copy()).astype(
                   float32)
                result[2] = gpu_ifft(vec[2].copy()).astype(
                   float32)
                return result
        else: # "Scalar", fft the whole thing
            if fft_type==1:
                return ifftn(vec,measure=True).astype(float32)
            elif fft_type==0:
                return ifftn(vec).astype(float32)
            elif fft_type==2:
                return gpu_ifft(vec.copy()).astype(float32)
    elif vec.ndim == 1: #Not a vector, so use fft
        if fft_type==1:
            return ifft(vec,measure = True).astype(float32)
        elif fft_type==0:
            return ifft(vec).astype(float32)
        elif fft_type==2:
            return gpu_ifft(vec).astype(float32)
    else:
        #0th index is 3, so its a vector
        return array([ifft(i) for i in vec]).astype(float32)

def av(a):
    """
    Takes a 3D array and computes the average over the x-y
       plane
```

```
    If the first index has size 3, its a vector so output will
       be a vector
    """
    from numpy import mean
    if a.shape[0] == 3:
        return mean(mean(a, -2), -2)
    else:
        return mean(mean(a, 0), 0)

def lowpass(a, wind='rect', alpha=0.1, fc = 1./64.):
    """
    Function which takes the num_ks smallest k-value positions
       in each dimension and filters the rest.
    The input is expected to be of the form a[3,...] for a
       vector.
    Alpha defines the sharpness of the filter window if hamming
        is chosen
    fc is the cutoff frequency relative to the nyquist
       frequency
    """
    from numpy import shape,ones,zeros,float32
    assert( wind=='rect' or wind=='hamming' or wind=='gauss' or
        wind=='shell')
    #Does the filtering in 1 line
    #    -FFT
    #    -Window the data
    #    -IFFT
    return ifftvec(create_filter(a.shape, fc, alpha, wind)*
       fftvec(a))

def tukey_filter(width, alpha):
    """
    Creates a window with length width and sharpness alpha
    1 is a cosine (Hamming)
    0 is a rectangle/boxcar/brickwall (passing 0 will actually
       give a divide by zero, so don't do that)
    """
    from numpy import arange, ones, cos, pi, float32
    #create the x values to pass to the function
    x = arange(width).astype(float32)
    #Do some fancy slicing with numpy arrays to create a
       piecewise function
    p1 = slice(None, int(alpha*(width-1)/2))
    p2 = slice(int(alpha*(width-1)/2), int((width-1)*(1-alpha
       /2)))
```

```
    p3 = slice( int((width-1)*(1-alpha/2)), int(width-1) )
    #Set default values to 1
    result = ones(width)
    #Create the piecewise function
    result[p1] = 0.5*(1.+cos(pi*(2.*x[p1]/alpha/(width-1)-1.)))
    result[p2] = 1.
    result[p3] = 0.5*(1.+cos(pi*(2.*x[p3]/alpha/(width-1)-alpha
       /2.+1.)))
    if width%2 == 0:
        result[:-width/2:-1] = result[:width/2-1] #mirror the
           window to work in k-space
    else:
        result[:-width/2:-1] = result[:width/2]
    return result

def gaussian_filter(length, cutoff):
    """
    Create a 1D gaussian filter with length and frequency
       cuttoff
    """
    from numpy import roll, exp, arange, insert
    #return roll(exp(-(arange(length) - length/2)**2/2./cutoff/
       cutoff),length/2)
    #create first half of filter
    filt = exp(-arange(length)**2 / 2. / cutoff / cutoff )
    #mirror the array
    filt[:-length/2+1:-1] = filt[1:length/2-1]

    return filt

def shell_filter(dims, fcs):
    """
    Create a spherical shell low pass filter

    Want all the positions with (kx**2 + ky*2 + kz**2) > fc**2
       to be = 0.
    """
    #Because I'm lazy, going to put in a hard code the spacing
    # the spacing is the same for the 128,512,128 runs
    # as it is for the 128,512,512 runs
    from numpy.fft import fftfreq
    from numpy import ones, max, min
    dx = 0.0078125
    kx,ky,kz = define_k(dims,[dx,dx,dx])
    shell = ones(dims)
```

```
    shell[(kx**2 + ky**2 + kz**2)>kx[min(fcs)]**2] = 0.
    return shell

def create_filter(axis_dimensions, fc, alpha, wind = 'rect'):
    """
    Creates the 3D filter window to be multiplied by the signal
    """
    from numpy import append, insert, ones, zeros, hstack,
       float32

    #If its a vector, only look at 1: indeces
    if axis_dimensions[0] == 3:
        ad = axis_dimensions[1:]
    else:
        ad = axis_dimensions

    #Define the cutoff frequency relative to the nyquist
    fcx = int(ad[0]*fc)
    fcy = int(ad[1]*fc)
    fcz = int(ad[2]*fc)

    #Rectangular, boxcar window
    if(wind == 'rect'):
        fcx = int(ad[0]*fc)
        fcy = int(ad[1]*fc)
        fcz = int(ad[2]*fc)
        f1 = zeros(ad[0])
        f2 = zeros(ad[1])
        f3 = zeros(ad[2])
        f1[:fcx+1]=1.
        f2[:fcy+1]=1.
        f3[:fcz+1]=1.
        f1[-fcx:]=1.
        f2[-fcy:]=1.
        f3[-fcz:]=1.
    #hamming window -- actually the tukey window
    elif(wind == 'hamming'):
        #Create the 1D filters in each dimension
        f1 = 1.-tukey_filter(ad[0]-fcx, alpha)
        f2 = 1.-tukey_filter(ad[1]-fcy, alpha)
        f3 = 1.-tukey_filter(ad[2]-fcz, alpha)
        #Pad the ones on the ends up to the cutoff desired
        f1 = hstack((ones((ad[0]-f1.size)/2),f1,ones((ad[0]-f1.
           size)/2)))
```

```
        f2 = hstack((ones((ad[1]-f2.size)/2),f2,ones((ad[1]-f2.
           size)/2)))
        f3 = hstack((ones((ad[2]-f3.size)/2),f3,ones((ad[2]-f3.
           size)/2)))
    #Gaussian window
    elif(wind == 'gauss'):
        f1 = gaussian_filter(ad[0], fcx*1.)
        f2 = gaussian_filter(ad[1], fcy*1.)
        f3 = gaussian_filter(ad[2], fcz*1.)
    #Shell type window
    elif(wind == 'shell'):
        return shell_filter(ad, [fcx, fcy, fcz])

    #This is needed if the array sizes get slightly borked
    if f1.size != ad[0]:
        f1 = insert(f1, ad[0]/2, 0.)
    if f2.size != ad[1]:
        f2 = insert(f2, ad[1]/2, 0.)
    if f3.size != ad[2]:
        f3 = insert(f3, ad[2]/2, 0.)

    return f1[:,None,None]*f2[None,:,None]*f3[None,None,:]

#Define some vector calc functions
k = define_k(N,D)
k2 = dot(k,k)
k2[k2==0] = 1.

def dot(a, b):
    """
    Warning: This shares a name with the numpy function
    Custom dot product function for (3, nx, ny, nz) sized
       arrays
    DOES NOT COMPLEX CONJUGATE
    *** Beware *** No checking is done about the size of the
       input arrays, you've been warned
    This should actually work for any array with 3 as the first
        index deonoting vector components.
    So long as they are the same shape (or can broadcast)
    """
    #Because of the way python-numpy arrays work, if the input
       are arrays, this is the same as:
    # b[0,...]*a[0,...] etc.
    return sum(i*j for i,j in zip(a,b))
```

```
def cross(a, b):
    """
    Warning: This shares a name with the numpy function
    Custom cross product routine for (3, nx, ny, nz) sized
       arrays
    The same warnings as the dot_p function apply.
       Interestingly, I could also make this a wrapper for
    numpy.cross(a, b, axis=0) as it does the same thing
    """
    from numpy import array
    return array([a[1]*b[2]-a[2]*b[1], a[2]*b[0]-a[0]*b[2], a
       [0]*b[1]-a[1]*b[0]])

def curl(y):
    return ifftvec(cross(1.j*k,fftvec(y)))

def div(y):
    return ifftvec(dot(1.j*k,fftvec(y)))

def curl(a):
    return ifftvec(cross(1.j*k, fftvec(a)))

def grad(a):
    a_ = fftvec(a)
    return ifftvec(array([1.j*k[0]*a_, 1.j*k[1]*a_, 1.j*k[2]*a_
       ]))

def a_dot_grad_b(a,b):
    return array([dot(a,grad(b[0])), dot(a,grad(b[1])), dot(a,
       grad(b[2]))])

## Code to do the corrections to the shearing box
   transformations before FFT's are applied
def unwrap2(v, N, D, time):
        """
        Attempt to vectorize the original function, unwrap
        This version has a significant speed up over unwrap 1 (
           factor of about 50)
        """
        from numpy import float32, zeros, linspace, shape,
           array
        Omega = 0.001
        q = 1.5
        dx,dy,dz = D
        nx,ny,nz = N
```

```
        Lx = nx*dx
        Ly = ny*dy
        Lz = nz*dz
        Ox = 0.5
        #This apparently finds the nearest periodic point
        dt = my_mod(time, Ly/(q * Omega * Lx))
        x = array([i*dx - Ox for i in range(nx)])
        dy_tot = -q * Omega * x * dt
        if v.shape[0] == 3:
                return array([yshift(v[0], dy_tot/dy),yshift(v
                   [1], dy_tot/dy), yshift(v[2], dy_tot/dy)])
        else:
                return array(yshift(v,dy_tot/dy))

def rewrap2(v, N, D, time):
        """
        This version has a significant speed up over rewrap1 (
           factor of about 50)
        """
        from numpy import float32, zeros, linspace, shape,
           array, arange
        Omega = 0.001
        q = 1.5
        dx,dy,dz = D
        nx,ny,nz = N
        Lx = nx*dx
        Ly = ny*dy
        Lz = nz*dz
        #This apparently finds the nearest periodic point
        dt = my_mod(time, Ly/(q * Omega * Lx))
        ky = arange(ny*1.)
        ky[ky>ny/2] -= 1.*ny
        ky /= Ly
        dkx_tot = q * Omega * ky * dt
        if(v.shape[0] == 3):
                return array([kxshift(v[0],dkx_tot*Lx),kxshift(
                   v[1],dkx_tot*Lx),kxshift(v[2],dkx_tot*Lx)])
        else:
                return array(kxshift(v,-dkx_tot*Lx))

def kxshift(y1, dn):
        from numpy import zeros, array, shape, empty_like
        nx,ny,nz = y1.shape
        inx = array([int(i) for i in dn])
        frac = dn - inx
```

```
        inx[frac<0] -= 1
        frac[frac<0] += 1.
        ip = []
        ipp = []
        #The trick to speeding this up was to use array slices
           instead of a mask
        for i in range(ny):
                ip.append(((range(nx) + inx[i])%nx, i,slice(
                   None)))
                ipp.append(((range(nx) + inx[i] + 1)%nx, i,
                   slice(None)))
        y2 = empty_like(y1)
        #This creation of the list is the slowest part
        #possibly the list comprehension is not efficient
        for i in range(ny):
                y2[:,i,:] = (1.-frac[i])*y1[ip[i]] + frac[i]*y1
                   [ipp[i]]
        return y2

def yshift(y1, dn):
        from numpy import zeros, array, shape, empty_like
        nx,ny,nz = y1.shape
        inx = array([int(i) for i in dn])
        frac = dn - inx
        inx[frac<0] -= 1
        frac[frac<0] += 1.
        ip = []
        ipp = []
        #The trick to speeding this up was to use array slices
           instead of a mask
        for i in range(nx):
                ip.append((i,(range(ny) + inx[i])%ny,slice(None
                   )))
                ipp.append((i,(range(ny) + inx[i] + 1)%ny,
                   slice(None)))
        y2 = empty_like(y1)
        #for i in range(nx):
        #       y2[i,:,:] = (1.-frac[i])*y1[ip[i]] + frac[i]*y1
           [ipp[i]]
        #return y2
        return [(1.-frac[i])*y1[ip[i]] + frac[i]*y1[ipp[i]] for
            i in range(nx)]

def mapped_fft2(a, N, D, t):
  from Functions import fftvec as fft
```

```
  return rewrap2(fft(unwrap2(a,N,D,t)),N,D,t)

def mapped_ifft2(a, N, D, t):
        from Functions import ifftvec as ifft
        return unwrap2(ifft(rewrap2(a,N,D,-t)),N,D,-t)

def highpass_mapped2(a):
    return mapped_ifft2((1.-create_filter(a.shape, alpha=alpha,
       fc=fc,wind=window))*mapped_fft2(a,N,D,t),N,D,t)

def lowpass_mapped2(a, N, D, t, alpha, fc, wind):
  #from Functions import fftvec as fft
  #from Functions import ifftvec as ifft
  fft = mapped_fft2
  ifft = mapped_ifft2
  from Functions import create_filter
  from numpy import shape,ones,zeros,float32
  assert( wind=='rect' or wind=='hamming' or wind=='gauss' or
     wind=='shell')
  #Does the filtering in 1 line
  #    -FFT
  #    -Window the data
  #    -IFFT
  #Create a rectangular window
  return ifft(create_filter(a.shape, fc, alpha, wind)*fft(a,N,D
     ,t),N,D,t)


def ndgrid(*args, **kwargs):
    """
    n-dimensional gridding like Matlab's NDGRID

    The input *args are an arbitrary number of numerical
       sequences,
    e.g. lists, arrays, or tuples.
    The i-th dimension of the i-th output argument
    has copies of the i-th input argument.

    Optional keyword argument:
    same_dtype : If False (default), the result is an ndarray.
                         If True, the result is a lists of
                            ndarrays, possibly with
                         different dtype. This can save
                            space if some *args
                         have a smaller dtype than others.
```

```
Typical usage:
>>> x, y, z = [0, 1], [2, 3, 4], [5, 6, 7, 8]
>>> X, Y, Z = ndgrid(x, y, z) # unpacking the returned
   ndarray into X, Y, Z

Each of X, Y, Z has shape [len(v) for v in x, y, z].
>>> X.shape == Y.shape == Z.shape == (2, 3, 4)
True
>>> X
array([[[0, 0, 0, 0],
                [0, 0, 0, 0],
                [0, 0, 0, 0]],
             [[1, 1, 1, 1],
                [1, 1, 1, 1],
                [1, 1, 1, 1]]])
>>> Y
array([[[2, 2, 2, 2],
                [3, 3, 3, 3],
                [4, 4, 4, 4]],
             [[2, 2, 2, 2],
                [3, 3, 3, 3],
                [4, 4, 4, 4]]])
>>> Z
array([[[5, 6, 7, 8],
                [5, 6, 7, 8],
                [5, 6, 7, 8]],
             [[5, 6, 7, 8],
                [5, 6, 7, 8],
                [5, 6, 7, 8]]])

With an unpacked argument list:
>>> V = [[0, 1], [2, 3, 4]]
>>> ndgrid(*V) # an array of two arrays with shape (2, 3)
array([[[0, 0, 0],
                [1, 1, 1]],
             [[2, 3, 4],
                [2, 3, 4]]])

For input vectors of different data types, same_dtype=False
    makes ndgrid()
return a list of arrays with the respective dtype.
>>> ndgrid([0, 1], [1.0, 1.1, 1.2], same_dtype=False)
[array([[0, 0, 0], [1, 1, 1]]),
 array([[ 1. ,  1.1,  1.2], [ 1. ,  1.1,  1.2]])]
```

```
    Default is to return a single array.
    >>> ndgrid([0, 1], [1.0, 1.1, 1.2])
    array([[[ 0. ,  0. ,  0. ], [ 1. ,  1. ,  1. ]],
                 [[ 1. ,  1.1,  1.2], [ 1. ,  1.1,  1.2]]])
    """
    from numpy import array, zeros, ones_like, append, shape
    same_dtype = kwargs.get("same_dtype", True)
    V = [array(v) for v in args] # ensure all input vectors are
        arrays
    shape = [len(v) for v in args] # common shape of the
       outputs
    result = []
    for i, v in enumerate(V):
            # reshape v so it can broadcast to the common shape
            # http://docs.scipy.org/doc/numpy/user/basics.
               broadcasting.html
            zero = zeros(shape, dtype=v.dtype)
            thisshape = ones_like(shape)
            thisshape[i] = shape[i]
            result.append(zero + v.reshape(thisshape))
    if same_dtype:
            return array(result) # converts to a common dtype
    else:
            return result # keeps separate dtype for each
               output
```