SLAC-TN-12-019

Investigating the Magnetorotational Instability with Dedalus, an Open-Source Hydrodynamics Code

Keaton J. Burns

Office of Science, Science Undergraduate Laboratory Internship (SULI)

University of California Berkeley

SLAC National Accelerator Laboratory

Menlo Park, CA 94025

August 17, 2011

Prepared in partial fulfillment of the requirements of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Jeffrey S. Oishi with the Kavli Institute for Particle Astrophysics and Cosmology.

Participant: _____

Signature

Research Advisor: _____

Signature

Work supported by US Department of Energy contract DE-AC02-76SF00515.

TABLE OF CONTENTS

ABSTRACT

Investigating the Magnetorotational Instability with Dedalus, an Open-Source Hydrodynamics Code. KEATON J. BURNS (University of California Berkeley, Berkeley, CA 94720) JEFFREY S. OISHI (Kavli Institute for Particle Astrophysics and Cosmology, Menlo Park, CA 94025)

The magnetorotational instability is a fluid instability that causes the onset of turbulence in discs with poloidal magnetic fields. It is believed to be an important mechanism in the physics of accretion discs, namely in its ability to transport angular momentum outward. A similar instability arising in systems with a helical magnetic field may be easier to produce in laboratory experiments using liquid sodium, but the applicability of this phenomenon to astrophysical discs is unclear. To explore and compare the properties of these standard and helical magnetorotational instabilities (MRI and HRMI, respectively), magnetohydrodynamic (MHD) capabilities were added to Dedalus, an open-source hydrodynamics simulator. Dedalus is a Python-based pseudospectral code that uses external libraries and parallelization with the goal of achieving speeds competitive with codes implemented in lower-level languages. This paper will outline the MHD equations as implemented in Dedalus, the steps taken to improve the performance of the code, and the status of MRI investigations using Dedalus.

INTRODUCTION

Astrophysical accretion discs are discs of material orbiting around a star or black hole. For decades, an outstanding theoretical problem in the physics of accretion discs was the identification of an efficient mechanism for transporting angular momentum outward, since viscous transport was unable to account for the observed rates of infalling material [1]. In 1991, Balbus and Hawley discovered a linear instability that occurs in discs with a poloidal magnetic field [2]. This magnetorotational instability (MRI) is generally accepted as the mechanism responsible for angular momentum transport and turbulence in accretion discs.

Laboratory experiments seeking to produce and study the MRI face a key difficulty: the toroidal field perturbations required by the MRI must be produced by induction if an entirely poloidal external magnetic field is applied. However, liquid metal flows with sufficiently high magnetic Reynolds numbers (Rm) to produce these induction effects tend to have extremely high hydrodynamic Reynolds numbers (Re) ($\frac{Rm}{Re} \approx 10^{-5}$ for these materials), and hence tend to be turbulent even without contributions from the MRI [3]. One solution to this issue is to apply a helical magnetic field in the laboratory, exciting a helical magnetorotational instability (HMRI). Simulations of the MRI and HMRI into their non-linear phases may provide information pertaining to the applicability of experimental HRMI results to the processes occurring in astrophysical discs.

METHODOLOGY: DEDALUS DEVELOPMENT

Dedalus is an open-source hydrodynamics code, written in Python 2.7. It is a pseudospectral code, meaning it uses fast Fourier transforms (FFTs) to compute spatial derivatives. It was designed to be flexible and easy to use, with the FFTs handled by external libraries and/or parallelization to abate the performance penalties of using a high-level language. The code makes extensive use of object-oriented programming, facilitating the modular

implementation of different domain representations and physics. Dedalus is currently hosted on a public Bitbucket repository[1].

MHD Equations

Incompressible MHD is primarily governed by four equations[2] [3]: the Navier-Stokes equation with the Lorentz force and viscosity, the induction equation with magnetic diffusivity, the mass continuity equation, and Gauss's law for magnetism:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho_0} + \frac{\mathbf{F_L}}{\rho_0} + \nu \nabla^2 \mathbf{u}, \tag{1}$$

$$\partial_t \mathbf{B} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}, \tag{2}$$

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \xrightarrow{\text{incomp.}} \nabla \cdot \mathbf{u} = 0, \tag{3}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{4}$$

Expanding the Lorentz force as

$$\mathbf{F_L} = \frac{(\nabla \times \mathbf{B}) \times \mathbf{B}}{4\pi} = \frac{\mathbf{B} \cdot \nabla \mathbf{B}}{4\pi} - \frac{\nabla B^2}{8\pi}, \tag{5}$$

Eq. (1) becomes

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla P_{tot}}{\rho_0} + \frac{\mathbf{B} \cdot \nabla \mathbf{B}}{4\pi \rho_0} + \nu \nabla^2 \mathbf{u}, \tag{6}$$

$$P_{tot} = p + \frac{B^2}{8\pi}. \tag{7}$$

[1] http://bitbucket.org/jsoishi/dedalus

[2] All presented here in Gaussian units, and with ∂_t indicating $\frac{\partial}{\partial t}$.

[3] With $\mathbf{u}$ the velocity field, $\mathbf{B}$ the magnetic field, p the pressure, ρ the density, ν the kinematic viscosity, and η the magnetic diffusivity.

Using the identity $\nabla \times (\mathbf{A} \times \mathbf{B}) = \mathbf{A}(\nabla \cdot \mathbf{B}) - \mathbf{B}(\nabla \cdot \mathbf{A}) + (\mathbf{B} \cdot \nabla)\mathbf{A} - (\mathbf{A} \cdot \nabla)\mathbf{B}$ along with the constraining equations (Eqs. (3) and (4)), Eq. (2) becomes

$$\partial_t \mathbf{B} = \mathbf{B} \cdot \nabla \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{B} + \eta \nabla^2 \mathbf{B}. \tag{8}$$

Spectral Implementation

In a periodic domain, any sufficiently smooth field variable can be represented by its discrete Fourier decomposition on the grid. That is, for some function f,

$$f(\mathbf{x}, t) = \frac{1}{\sqrt{N}} \sum_{\mathbf{k}} \hat{f}(\mathbf{k}, t) e^{i x_i k_i}, \tag{9}$$

$$\hat{f}(\mathbf{k}, t) = \frac{1}{\sqrt{N}} \sum_{\mathbf{x}} f(\mathbf{x}, t) e^{-i x_i k_i}. \tag{10}$$

Pseudospectral codes, such as Dedalus, use FFTs to evolve partial differential equations in Fourier space, where they become systems of ordinary differential equations. This is because the spatial derivatives in the governing equations become inexpensive multiplications under the Fourier transform: $\nabla \xrightarrow{\text{FT}} i\mathbf{k}$. The spectral implementation follows Maron and Goldreich [4].

Taking the Fourier transforms of Eqs. (6), (8), (3), and (4) yields

$$\partial_t \hat{\mathbf{u}} = -\widehat{\mathbf{u} \cdot \nabla \mathbf{u}} - \frac{i\mathbf{k}\hat{P_{tot}}}{\rho_0} + \frac{\widehat{\mathbf{B} \cdot \nabla \mathbf{B}}}{4\pi\rho_0} - \nu k^2 \hat{\mathbf{u}}, \tag{11}$$

$$\partial_t \hat{\mathbf{B}} = \widehat{\mathbf{B} \cdot \nabla \mathbf{u}} - \widehat{\mathbf{u} \cdot \nabla \mathbf{B}} - \eta k^2 \mathbf{B}, \tag{12}$$

$$i\mathbf{k} \cdot \hat{\mathbf{u}} = 0, \tag{13}$$

$$i\mathbf{k} \cdot \hat{\mathbf{B}} = 0, \tag{14}$$

the Fourier space evolution and constraining equation pairs, respectively. Taking the scalar product of $i\mathbf{k}$ with Eq. (11) leads to an expression for the total pressure:

$$\frac{\hat{P_{tot}}}{\rho_0} = \frac{i\mathbf{k} \cdot \widehat{\mathbf{u} \cdot \nabla \mathbf{u}}}{k^2} - \frac{i\mathbf{k} \cdot \widehat{\mathbf{B} \cdot \nabla \mathbf{B}}}{4\pi\rho_0 k^2}. \tag{15}$$

The nonlinear terms ($\widehat{\mathbf{u} \cdot \nabla \mathbf{u}}$, $\widehat{\mathbf{u} \cdot \nabla \mathbf{B}}$, $\widehat{\mathbf{B} \cdot \nabla \mathbf{u}}$, and $\widehat{\mathbf{B} \cdot \nabla \mathbf{B}}$) in Eqs. (11), (12), and (15) are computed in real space. To eliminate aliasing effects, the 2/3 rule is utilized, zeroing any mode with a k component greater than or equal to 2/3 of the Nyquist wavenumber in that direction. This zeroing is done prior to every reverse Fourier transform, after every forward Fourier transform, and at each temporal evolution.

Temporal Integration

The Fourier space ODEs, along with the initial conditions specified at the start of the simulation, form an initial value problem that is integrated using explicit Runge-Kutta methods, specifically the second-order midpoint method. For simulations with viscosity and/or magnetic diffusivity, an integrating factor is used to evaluate the normally linear steps used to construct the Runge-Kutta stages. Consider Eq. (11) for a specified mode $\mathbf{k}$, with the non-viscous terms considered to be constant during an integration step:

$$\partial_t \hat{\mathbf{u}}(t) + \nu k^2 \hat{\mathbf{u}}(t) = \mathbf{RHS}. \tag{16}$$

This is an equation of the form $y'(x) + P(x)y = Q(x)$, which has the exact solution

$$y(x) = \frac{\int Q(x)M(x)dx}{M(x)}, \tag{17}$$

where $M(x) = e^{\int P(x)dx}$ is called the integrating factor. Hence the solution of Eq. (16) at time $t + dt$ is found to be

$$\hat{\mathbf{u}}(t + dt) = \left[\hat{\mathbf{u}}(t) + \frac{\mathbf{RHS}}{\nu k^2}(e^{\nu k^2 dt} - 1)\right] e^{-\nu k^2 dt}. \tag{18}$$

Shearing Box

To study the effects of the MRI, local simulations are performed in which the computational domain represents a small part of an astrophysical disc. The domain is taken to be a co-rotating box, whose left edge is a distance r_0 from the axis of rotation, and whose length in each dimension is much less than this fiducial radius. In the co-rotating frame, the unit vector $\mathbf{e_x}$ is in the outward radial direction, and the unit vector $\mathbf{e_z}$ is along the axis of rotation. The box is rotating with an angular velocity $\mathbf{\Omega_0} = \Omega_0 \mathbf{e_z} = \Omega(r_0)\mathbf{e_z}$.

The radial dependence of angular velocity in a Keplerian disc, $\Omega(r) = \sqrt{GM} r^{-3/2}$, gives rise to a linear shear flow in this domain: with the domain moving at the angular velocity of the left (inner) edge, the fluid in the box will shear in the x direction with a velocity of $-\frac{3}{2}\Omega_0 x \mathbf{e_y}$, as shown in Fig. 1. This shear motivates the construction of a domain representation and a corresponding physics implementation to handle MHD in a box with an arbitrary local linear shear.

Consider an arbitrary power-law shearing profile, $\Omega(r) = Cr^S$ (which arises from an attractive force of magnitude $\rho_0 C^2 r^{2S+1}$ and gives rise to a linear background shear in the local frame with velocity $S\Omega_0 x \mathbf{e_y}$. Hence $C = \sqrt{GM}$ and $S = -3/2$ for Keplerian rotation). In this case, the centrifugal ($-\mathbf{\Omega_0} \times (\mathbf{\Omega_0} \times \mathbf{r}) = {\Omega_0}^2 r \mathbf{e_x}$) and attractive ($-C^2 r^{2S+1} \mathbf{e_x}$) accelerations partially cancel (via approximations utilizing $x, y, z \ll r_0$):

$$\mathbf{a_{rad}} = ({\Omega_0}^2 - C^2 r^{2S}) r \mathbf{e_x} \approx -2S{\Omega_0}^2 x \mathbf{e_x}. \tag{19}$$

With this radial acceleration and the Coriolis acceleration ($-2\mathbf{\Omega_0}\times\mathbf{v}$), Eq. (1) for the velocity field in the rotating frame, $\mathbf{v}$, becomes

$$\partial_t\mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v} = -\frac{\nabla p}{\rho_0} + \frac{\mathbf{F_L}}{\rho_0} + \nu\nabla^2\mathbf{v} - 2S{\Omega_0}^2 x\mathbf{e_x} - 2\mathbf{\Omega_0}\times\mathbf{v}. \tag{20}$$

Decomposing $\mathbf{v}$ into the background shear flow and velocity perturbations ($\mathbf{v} = S\Omega_0 x\mathbf{e_y} + \mathbf{u}$), Eq. (20) becomes

$$\partial_t\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{\nabla p}{\rho_0} + \frac{\mathbf{F_L}}{\rho_0} + \nu\nabla^2\mathbf{u} + 2\Omega_0 u_y\mathbf{e_x} - (2+S)\Omega_0 u_x\mathbf{e_y} - S\Omega_0 x\partial_y\mathbf{u}. \tag{21}$$

To account for the background shear in the evolution of the perturbations, the remesh-free approach of Brucker et al [5] was implemented. Due to the shear in the local frame, the forward and reverse Fourier transforms must be modified to maintain periodicity along the shearing direction:

$$\hat{f}(\mathbf{k},t) = \sum_{\mathbf{x}} f(\mathbf{x},t)e^{-i(x_i k_i - S\Omega_0 x t k_y)}, \tag{22}$$

$$f(\mathbf{x},t) = \sum_{\mathbf{k}} \hat{f}(\mathbf{k},t)e^{i(x_i k_i - S\Omega_0 x t k_y)}. \tag{23}$$

Hence, the fixed-grid wavevector $\mathbf{K}$ becomes a function of the Lagrangian (shearing) wavevector $\mathbf{k}$ and time, $\mathbf{K}(\mathbf{k},t) = (k_x - S\Omega_0 t k_y, k_y, k_z)$, and when transforming to Fourier space, the derivative operators become

$$\partial_x \rightarrow i(k_x - S\Omega_0 t k_y) = iK_x, \tag{24}$$

$$\partial_y \rightarrow ik_y = iK_y, \tag{25}$$

$$\partial_z \rightarrow ik_z = iK_z, \tag{26}$$

$$\partial_t \rightarrow \partial_t - iS\Omega_0 x k_y. \tag{27}$$

The shearing box implementations of Eqs. (11) and (15) are then

$$\partial_t \hat{\mathbf{u}} = -\widehat{\mathbf{u}\cdot\nabla\mathbf{u}} - \frac{i\mathbf{K}\hat{P_{tot}}}{\rho_0} + \frac{\widehat{\mathbf{B}\cdot\nabla\mathbf{B}}}{4\pi\rho_0} - \nu K^2\hat{\mathbf{u}} + 2\Omega_0\hat{u_y}\mathbf{e}_{\hat{\mathbf{x}}} - (2+S)\Omega_0\hat{u_x}\mathbf{e}_{\hat{\mathbf{y}}}, \tag{28}$$

$$\frac{\hat{P_{tot}}}{\rho_0} = \frac{i\mathbf{K}\cdot\hat{\mathbf{M}}}{K^2} - \frac{i\mathbf{K}\cdot\hat{\mathbf{L}}}{4\pi\rho_0 K^2} - \frac{2i\Omega_0\hat{u_y}K_x}{K^2} + \frac{(1+S)2i\Omega_0\hat{u_x}K_y}{K^2}. \tag{29}$$

Eq. (8) becomes, in velocity perturbations,

$$\partial_t\mathbf{B} = \mathbf{B}\cdot\nabla\mathbf{u} - \mathbf{u}\cdot\nabla\mathbf{B} + \eta\nabla^2\mathbf{B} + S\Omega_0 B_x\mathbf{e}_{\mathbf{y}} - S\Omega_0 x\partial_y\mathbf{B}, \tag{30}$$

and the shearing box implementation of Eq. (12) becomes

$$\partial_t\hat{\mathbf{B}} = \widehat{\mathbf{B}\cdot\nabla\mathbf{u}} - \widehat{\mathbf{u}\cdot\nabla\mathbf{B}} - \eta K^2\mathbf{B} + S\Omega_0\hat{B_x}\mathbf{e}_{\hat{\mathbf{y}}}. \tag{31}$$

FFTs and Parallelization

Although Dedalus is written in Python, the FFTs dominate the computational cost of a simulation. Optimizing the FFTs largely negates the performance penalties of using a high-level language, while maintaining the ease of use and speed of development of Python.

Outsourcing the FFTs from Python to FFTW, a C-based library that optimizes FFT routines based on local hardware, results in a substantial speed improvement over Python's (i.e. NumPy's) built-in FFT algorithms. Much greater gains can be made on a graphics processing unit (GPU) using Nvidia's CUDA architecture to compute the FFTs and other calculations.

Finally, MPI-based parallelization has been implemented, allowing a single simulation to simultaneously run as N separate tasks. To achieve this, the computational domain is evenly divided among the N tasks along the k_z direction in Fourier space. The reverse Fourier transform is then accomplished in a series of steps, as depicted in Fig. 2. First, each

task performs a 1D IFFT in the k_x direction on its dataset, and the shearing phase shift is applied as in Eq. (23), if necessary. Second, an MPI All-To-All call is issued, in which each task evenly divides its data N times in the x direction, and sends the Nth slab to the Nth task. Each task then stacks the N slabs it has received, and performs a 2D IFFT in the k_y and k_z directions. The resulting datasets have gone through a full 3D IFFT, and the data is evenly divided among the tasks along the x direction in real space. The parallelized forward transform is the reverse of this process.

RESULTS

Code Verification and Testing

Multiple tests were performed to verify the MHD implementation in Dedalus. First, the motion of a shear-Alfvén wave, an eigenmode of the linearized MHD equations which travels along magnetic field lines, was tested. A constant background magnetic field $\mathbf{B_0}$ was setup across the domain, along with the velocity and magnetic field perturbations corresponding to a shear-Alfvén eigenmode. Shear-Alfvén waves were found to propagate with the correct phase velocities based on their direction of travel and the magnetic field strength: $v_A \propto B_0 \cos\theta$, where v_A is the speed of a shear-Alfvén wave and θ is the angle between the magnetic field and the wavevector.

This test was repeated with nonzero viscosity and magnetic diffusion. Fig. 3 shows the resulting amplitude and phase velocity plots for several seeded modes, which behave as expected. The velocity and magnetic field amplitudes decay exponentially, with rates proportional to wavevector magnitude, as expected from Eq. (18), and the phase velocities are constant in time and of the expected magnitude (here $\mathbf{B_0}$ is directed along $(1, 0, 0)$).

The advection of a low amplitude magnetic field was also tested. A small ($|\mathbf{B}| \approx 10^{-3}$), random magnetic field was imposed on a constant fluid velocity of order unity. The magnetic

field forced velocity perturbations of the same order, and both fields were advected with the background fluid velocity. This behavior was expected, since the non-linear terms are negligible with such small field perturbations.

The shearing box implementation was tested using the swinging wave test of Lithwick [6]. The wavefronts of the vorticity waves are seen to turn with the local domain's shear, as expected. Snapshots of the wave at several times are shown in Fig. 4.

Science Results

At the time of this report, the parallelization techniques described above are being finalized and tested. While they are critical for high-resolution simulations quantitatively studying the MRI, linear and early non-linear behavior can be investigated at lower resolution. These MRI simulations begin with a vertical magnetic field and small ($\approx 10^{-6}$) random velocity perturbations. Channel modes, precursors to turbulence in the MRI, are seen to form and grow exponentially. The mode formation is seen in Fig. 5, which depicts $\mathbf{u_x}$ in one xz plane at several times. The modes lie primarily in xy planes and are nearly axisymmetric, as expected. Higher resolution simulations following these modes further into the non-linear domain will be performed utilizing the parallel architecture of Dedalus.

DISCUSSION AND CONCLUSIONS

Dedalus is among the first entirely open-source spectral MHD codes. The Python development environment facilitates ease of use and code development, and the project's emphasis on object-oriented techniques have helped make Dedalus a very modular code which can be easily adapted to study a variety of problems. The parallelized FFT algorithms, the use of external libraries, and the incorporation of GPU-based calculations using CUDA all contribute substantially to the performance of the code. With shearing-box simulations

underway, the MHD capabilities of Dedalus may help identify points of comparison and departure between current lab-based HMRI experiments and the standard MRI operating in astrophysical accretion discs.

ACKNOWLEDGMENTS

Dedalus is currently being developed by Jeffrey S. Oishi, Oliver Hahn, Keaton J. Burns, and Greg Peairs. This project was made possibly by the U.S. Department of Energy's SULI program, and the KIPAC group at SLAC National Accelerator Laboratory and Stanford University.

REFERENCES

[1] G. Lesur and P.-Y. Longaretti, "Impact of dimensionless numbers on the efficiency of magnetorotational instability induced turbulent transport," *Monthly Notices of the Royal Astronomical Society*, vol. 378, no. 4, pp. 1471–1480, Jul. 2007. [Online]. Available: http://doi.wiley.com/10.1111/j.1365-2966.2007.11888.x

[2] S. A. Balbus and J. F. Hawley, "A powerful local shear instability in weakly magnetized disks. I - Linear analysis. II - Nonlinear evolution," *The Astrophysical Journal*, vol. 376, pp. 214–233, 1991. [Online]. Available: http://adsabs.harvard.edu/abs/1991ApJ...376..214B

[3] O. N. Kirillov and F. Stefani, "ON THE RELATION OF STANDARD AND HELICAL MAGNETOROTATIONAL INSTABILITY," *The Astrophysical Journal*, vol. 712, no. 1, pp. 52–68, Mar. 2010. [Online]. Available: http://arxiv.org/abs/0911.0067

[4] J. Maron and P. Goldreich, "Simulations of Incompressible Magnetohydrodynamic Turbulence," *The Astrophysical Journal*, vol. 554, no. 2, pp. 1175–1196, Jun. 2001.

[Online]. Available: http://arxiv.org/abs/astro-ph/0012491http://adsabs.harvard.edu/abs/2001ApJ...554.1175M

[5] K. A. Brucker, J. C. Isaza, T. Vaithianathan, and L. R. Collins, "Efficient algorithm for simulating homogeneous turbulent shear flow without remeshing," *Journal of Computational Physics*, vol. 225, no. 1, pp. 20–32, Jul. 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0021999106004827

[6] Y. Lithwick, "Nonlinear evolution of hydrodynamical shear flows in two dimensions," *The Astrophysical Journal*, vol. 670, pp. 789–804, 2007.

FIGURES

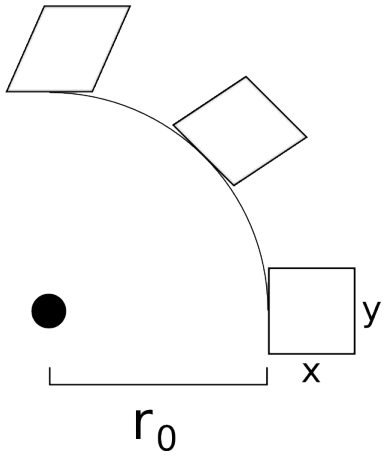

Figure 1: The motion of fluid tracers initially aligned with the edges of the co-rotating local domain, as viewed from an inertial frame. Lower angular velocity at larger radii results in a local linear shear across the x direction of the local domain. The size of the box relative to r_0 is exaggerated for clarity.

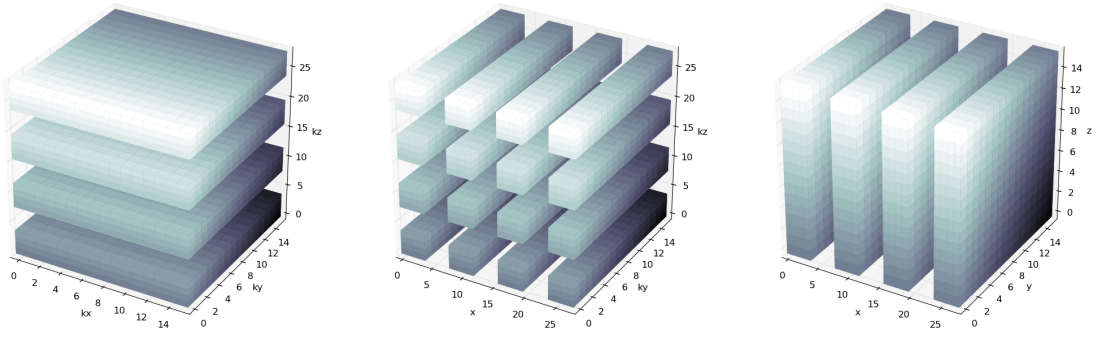

Figure 2: Data distribution using MPI parallelization: task cuts along k_z in Fourier space, data passing during MPI All-To-All call, and task cuts along x in real space.

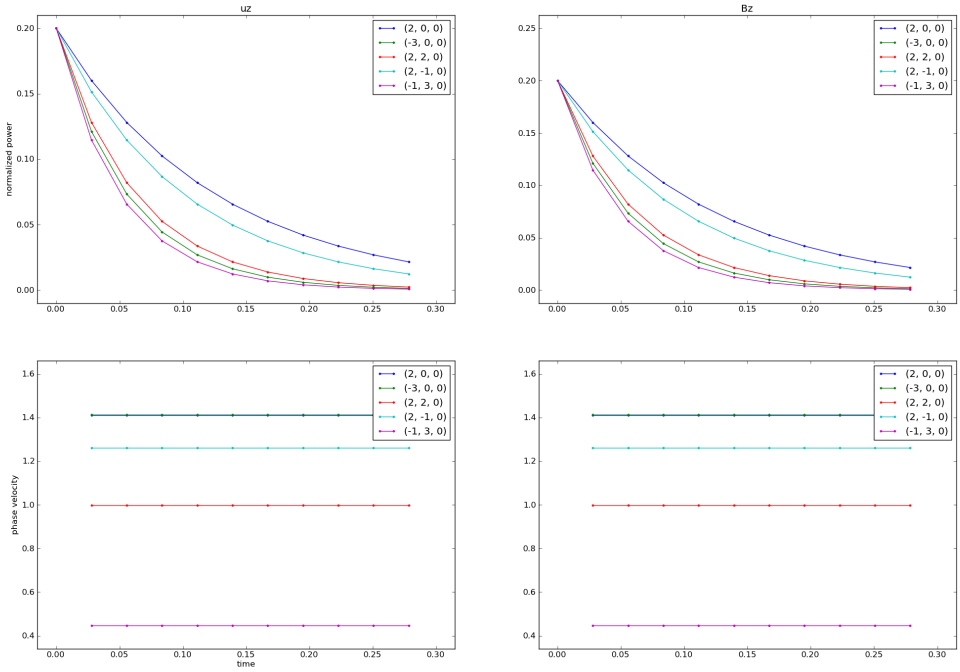

Figure 3: Amplitude and phase velocity evolution of several shear-Alfvén waves, with viscosity and magnetic diffusivity.

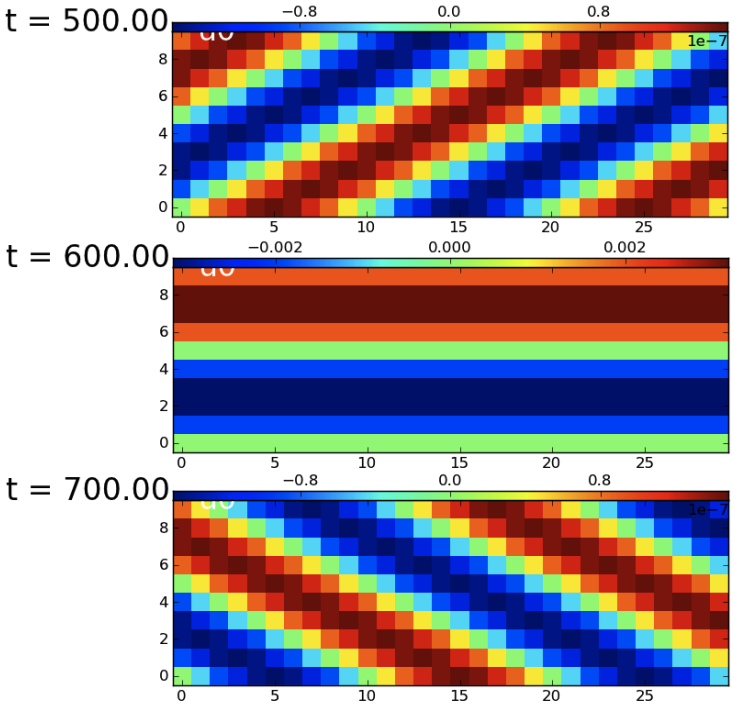

Figure 4: Swinging wave demonstrating local linear shear ($\mathbf{u_x}$ depicted).

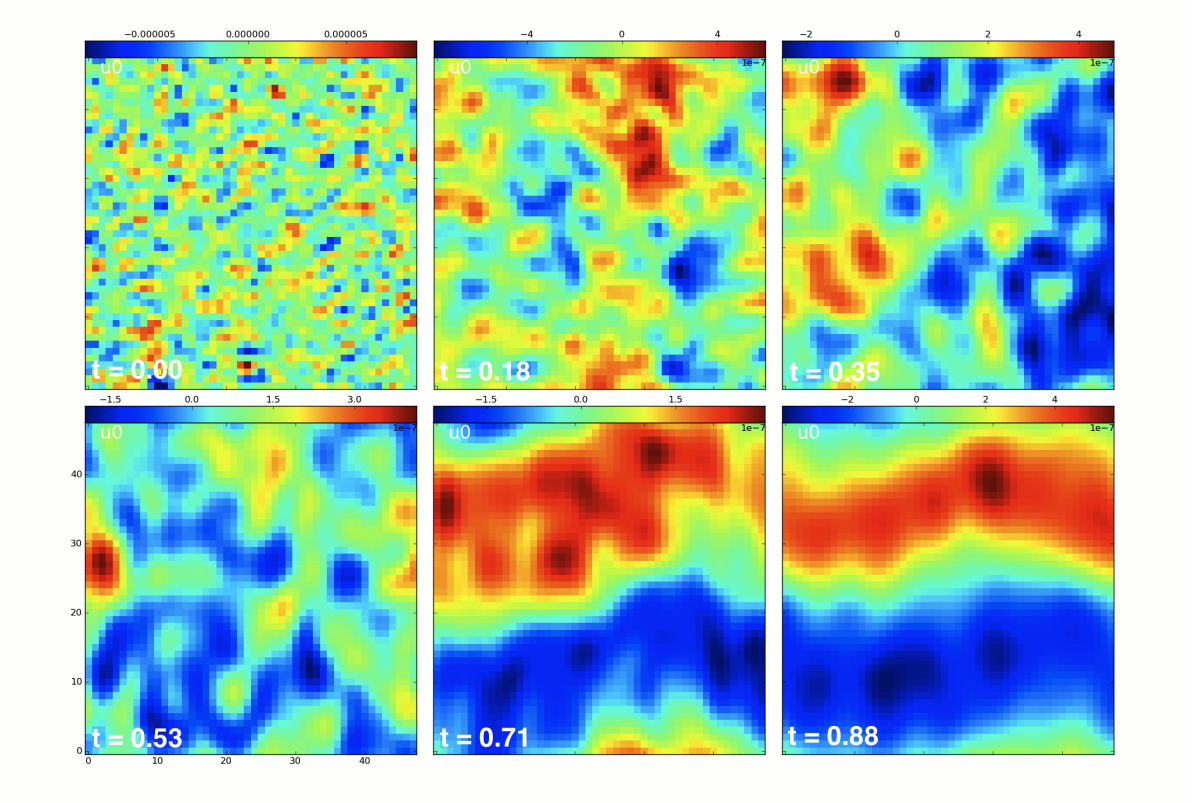

Figure 5: Formation of channel modes ($\mathbf{u_x}$ depicted in an xz plane).