The spherical MHD code MagIC

Advanced (1/2)

Thomas Gastine

Institut de Physique du Globe de Paris

5th July 2017

Outline

- MagIC simulates rotating fluid dynamics in a spherical shell
- It solves for the coupled evolution of Navier-Stokes equation, MHD equation, temperature (or entropy) equation and an equation for chemical composition under both the anelastic and the Boussinesq approximations
- A dimensionless formulation of the equations is assumed
- MagIC is a free software (GPL), written in Fortran
- Post-processing relies on python libraries
- **Poloidal/toroidal decomposition is employed**
- MagIC uses spherical harmonic decomposition in the angular directions
- Chebyshev polynomials or finite differences are employed in the radial direction
- MagIC uses a mixed implicit/explicit time stepping scheme
- The code relies on a hybrid parallelisation scheme (MPI/OpenMP)

Poloidal-toroidal decomposition of solenoidal vectors

General characterisation for **solenoidal vector fields**:

$$\boldsymbol{\nabla} \cdot \mathbf{v} = 0 \quad \Leftrightarrow \quad \mathbf{v} = \mathbf{P} + \mathbf{T}$$

$$\mathbf{v} = \boldsymbol{\nabla} \times \boldsymbol{\nabla} \times (W\, \mathbf{e_r}) + \boldsymbol{\nabla} \times (Z\, \mathbf{e_r})$$

W is the **poloidal** potential and Z is the **toroidal potential** (e.g. Chandrasekhar 1961). The radial component of the vector **v** is **purely poloidal**.

Poloidal/Toroidal decomposition

Three unknown field components of a solenoidal vector can be replaced by two scalar fields.

Dimensionless Boussinesq MHD equations

From 9 equations for 8 unknowns...

$$\nabla \cdot \mathbf{u} = 0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{2}{E} \mathbf{e_z} \times \mathbf{u} = -\nabla p' + \frac{Ra}{Pr} g(r)\, T' \mathbf{e_r} + \frac{1}{E\, Pm} (\nabla \times \mathbf{B}) \times \mathbf{B} + \Delta \mathbf{u}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \frac{1}{Pm} \Delta \mathbf{B}$$

$$\frac{\partial T'}{\partial t} + \mathbf{u} \cdot \nabla T' = \frac{1}{Pr} \Delta T'$$

9 equations, 8 unknowns...

Dimensionless Boussinesq MHD equations

To 6 equations for 6 unknowns...

1. Introduce Pol/Tor decomposition for $\tilde{\rho}\mathbf{u}$ and $\mathbf{B}$:

$$\tilde{\rho}\mathbf{u} = \nabla \times \nabla \times (W\,\mathbf{e_r}) + \nabla \times (Z\,\mathbf{e_r})$$
$$\mathbf{B} = \nabla \times \nabla \times (g\,\mathbf{e_r}) + \nabla \times (h\,\mathbf{e_r})$$

2. 6 unknowns: W, Z, g, h, p' and T'
3. Establish poloidal and toroidal Navier-Stokes equations, poloidal and toroidal induction equations, an equation for pressure and heat equation.

Poloidal/Toroidal equations (1/3)

Operators

From vectorial to toroidal and poloidal equations via operators:

$$\mathbf{e_r} \cdot [\tilde{\rho}\mathbf{u}] = -\Delta_H W,$$
$$\mathbf{e_r} \cdot [\boldsymbol{\nabla} \times \tilde{\rho}\mathbf{u}] = -\Delta_H Z,$$

where Δ_H denotes the horizontal part of the Laplacian:

$$\Delta_H = \Delta - \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) = \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\sin\theta\frac{\partial}{\partial\theta} + \frac{1}{r^2\sin\theta}\frac{\partial^2}{\partial^2\phi}$$

N.B. vectors can be expanded as follows:

$$\tilde{\rho}u_r = -\Delta_H W; \quad \tilde{\rho}u_\theta = \frac{1}{r}\frac{\partial^2 W}{\partial r\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial Z}{\partial\phi}; \quad \tilde{\rho}u_\phi = \frac{1}{r\sin\theta}\frac{\partial^2 W}{\partial r\partial\phi} - \frac{1}{r}\frac{\partial Z}{\partial\theta}$$

Poloidal/Toroidal equations (2/3)

- Poloidal potential: take $\mathbf{e_r} \cdot [\cdots]$ of the NS equation:

$$\mathbf{e_r} \cdot \tilde{\rho}\frac{\partial \mathbf{u}}{\partial t} = \frac{\partial}{\partial t}(\mathbf{e_r} \cdot \tilde{\rho}\mathbf{u}) = -\Delta_H \frac{\partial W}{\partial t}$$

- Toroidal potential: take $\mathbf{e_r} \cdot \boldsymbol{\nabla} \times [\cdots]$ of the NS equation:

$$\mathbf{e_r} \cdot \boldsymbol{\nabla} \times \left(\frac{\partial \tilde{\rho}\mathbf{u}}{\partial t}\right) = \frac{\partial}{\partial t}(\mathbf{e_r} \cdot \boldsymbol{\nabla} \times \tilde{\rho}\mathbf{u}) = -\Delta_H \frac{\partial Z}{\partial t}$$

- Pressure: take $\nabla_H \cdot [\cdots]$ of the NS equation:

$$\nabla_H \cdot \left(\tilde{\rho}\frac{\partial u}{\partial t}\right) = \Delta_H \frac{\partial}{\partial t}\left(\frac{\partial W}{\partial r}\right)$$

N.B. Some spherical shell codes get rid of pressure by instead taking $\mathbf{e_r} \cdot \boldsymbol{\nabla} \times \boldsymbol{\nabla} \times [\cdots]$ to derive the equation for the toroidal potential

Poloidal/Toroidal equations (3/3)

One has to proceed the same way for each **linear term**! As an example: Coriolis force that enters the toroidal potential equation:

$$
\begin{aligned}
\mathbf{e_r} \cdot \boldsymbol{\nabla} \times [2\tilde{\rho}\mathbf{u} \times \mathbf{e_z}] &= 2\,\mathbf{e_r} \cdot [(\mathbf{e_z} \cdot \boldsymbol{\nabla})(\tilde{\rho}\mathbf{u})] \\
&= 2\left[\cos\theta \frac{\partial(\tilde{\rho}u_r)}{\partial r} - \frac{\sin\theta}{r}\frac{\partial(\tilde{\rho}u_r)}{\partial\theta} + \frac{\tilde{\rho}u_\theta \sin\theta}{r}\right] \\
&= 2\left[-\cos\theta \frac{\partial}{\partial r}(\Delta_H W) + \right. \\
&\qquad \left. \frac{\sin\theta}{r}\frac{\partial}{\partial\theta}(\Delta_H W) + \frac{\sin\theta}{r^2}\frac{\partial^2 W}{\partial r\partial\theta} + \frac{1}{r^2}\frac{\partial Z}{\partial\phi}\right]
\end{aligned}
$$

...

Outline

- MagIC simulates rotating fluid dynamics in a spherical shell
- It solves for the coupled evolution of Navier-Stokes equation, MHD equation, temperature (or entropy) equation and an equation for chemical composition under both the anelastic and the Boussinesq approximations
- A dimensionless formulation of the equations is assumed
- MagIC is a free software (GPL), written in Fortran
- Post-processing relies on python libraries
- Poloidal/toroidal decomposition is employed
- **MagIC uses spherical harmonic decomposition in the angular directions**
- Chebyshev polynomials or finite differences are employed in the radial direction
- MagIC uses a mixed implicit/explicit time stepping scheme
- The code relies on a hybrid parallelisation scheme (MPI/OpenMP)

Spherical harmonic functions

- Spherical harmonic functions Y_ℓ^m are a natural choice for the horizontal expansion in colatitude θ and longitude ϕ

$$\boxed{Y_\ell^m(\theta,\phi) = P_\ell^m(\cos\theta)\, e^{im\phi}}$$

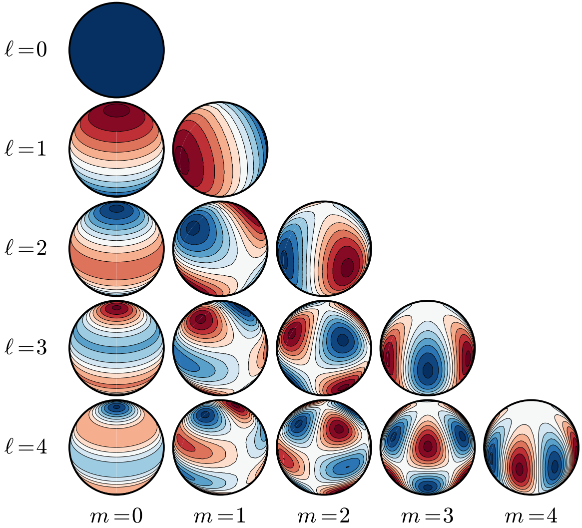
- Degree ℓ and order m
- In MagIC, we adopt a **complete normalisation** of SH:

$$\int_0^{2\pi}\int_0^{\pi} Y_\ell^m(\theta,\phi)\, Y_{\ell'}^{m'*}(\theta,\phi) \sin\theta \,\mathrm{d}\theta\,\mathrm{d}\phi = \delta_{\ell\ell'}\delta^{mm'}$$

- This yields:

$$Y_\ell^m(\theta,\phi) = \left(\frac{(2\ell+1)}{4\pi}\frac{(\ell-|m|)!}{(\ell+|m|)!}\right)^{1/2} P_\ell^m(\cos\theta)\, e^{im\phi}$$

First few spherical harmonics



Some mathematical properties of the spherical harmonics

- Complete and **orthogonal eigenfunctions of** Δ_H:

$$\boxed{\Delta_H Y_\ell^m = -\frac{\ell(\ell+1)}{r^2}\,Y_\ell^m\,.}$$

- Some useful **recursion relations**:

$$\cos\theta\, Y_\ell^m = c_{\ell+1}^m\, Y_{\ell+1}^m + c_\ell^m\, Y_{\ell-1}^m$$

$$\sin\theta \frac{\partial Y_\ell^m}{\partial\theta} = \ell\, c_{\ell+1}^m\, Y_{\ell+1}^m - (\ell+1) c_\ell^m\, Y_{\ell-1}^m$$

$$\text{with}\quad c_{\ell m} = \left[\frac{(\ell+m)(\ell-m)}{(2\ell+1)(2\ell-1)}\right]^{1/2}$$

- Practically this is how θ and ϕ derivatives are computed in MagIC

From spatial to spectral space (1/4)

Inverse spherical harmonic transform

$$\boxed{(r,\theta,\phi) \quad \rightarrow \quad (r,\ell,m)}$$

Suppose we have $Z(r,\theta,\phi,t)$ on a **longitude/latitude representation** (N_θ, N_ϕ). The expansion of the horizontal structure in series of spherical harmonics yields:

$$Z(r,\theta,\phi,t) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} Z_\ell^m(r,t)\, Y_\ell^m(\theta,\phi)$$

Spherical harmonic representation **truncated at degree and order** ℓ_{max}.

From spatial to spectral space (2/4)

Inverse spherical harmonic transform

$$\boxed{(r,\theta,\phi) \quad \rightarrow \quad (r,\ell,m)}$$

One has

$$Z_\ell^m(r,t) = \frac{1}{\pi}\int_0^\pi Z^m(r,\theta,t)\, P_\ell^m(\cos\theta)\sin\theta\,\mathrm{d}\theta$$

with

$$Z^m(r,\theta,t) = \frac{1}{2\pi}\int_0^{2\pi} Z(r,\theta,\phi,t)e^{-im\phi}\mathrm{d}\phi$$

From spatial to spectral space (2/4)

Inverse spherical harmonic transform

$$\boxed{(r, \theta, \phi) \quad \rightarrow \quad (r, \ell, m)}$$

One has

$$Z_\ell^m(r,t) = \frac{1}{\pi}\int_0^\pi Z^m(r,\theta,t)\, P_\ell^m(\cos\theta)\sin\theta\,\mathrm{d}\theta$$

with

$$Z^m(r,\theta,t) = \frac{1}{2\pi}\int_0^{2\pi} Z(r,\theta,\phi,t)e^{-im\phi}\mathrm{d}\phi$$

How do we compute those transformations?

From spatial to spectral space (3/4)

Inverse spherical harmonic transform

$$(r,\theta,\phi) \quad \rightarrow \quad (r,\ell,m)$$

First, we compute an **inverse FFT**:

$$\begin{aligned} Z^m(r,\theta,t) &= \frac{1}{2\pi}\int_0^{2\pi} Z(r,\theta,\phi,t)e^{-im\phi}\mathrm{d}\phi \\ &= \frac{1}{N_\phi}\sum_{j=0}^{N_\phi-1} Z(r,\theta,\phi_j,t)e^{-im\phi_j} \quad \text{with} \quad \phi_j = \frac{2j\pi}{N_\phi} \end{aligned}$$

$\rightarrow$ ϕ_j **needs to be evenly spaced.** N_ϕ **must be "FFT-friendly"** (restrictions in MagIC).

From spatial to spectral space (4/4)

Inverse spherical harmonic transform

$$\boxed{(r, \theta, \phi) \quad \rightarrow \quad (r, \ell, m)}$$

Second, we compute an **inverse Legendre transform**

$$\begin{aligned} Z_\ell^m(r,t) &= \frac{1}{\pi}\int_0^\pi Z^m(r,\theta,t)\, P_\ell^m(\cos\theta)\sin\theta\,\mathrm{d}\theta \\ &= \frac{1}{N_\theta}\sum_{k=0}^{N_\theta-1} w_k\, Z^m(r,\theta_k,t)\, P_\ell^m(\cos\theta_k) \end{aligned}$$

Gaussian quadrature points and Gauss-Legendre weights yield:

$$\theta_k \quad \text{given by} \quad P^0_{N_\theta}(\cos\theta_k) = 0 \quad \text{and} \quad w_k = \frac{2}{(N_\theta+1)^2}\left(\frac{\sin\theta_k}{P^0_{N_\theta+1}(\cos\theta_k)}\right)^2$$

From spectral to spatial space

Inverse spherical harmonic transform

$$\boxed{(r, \ell, m) \quad \rightarrow \quad (r, \theta, \phi)}$$

Simply the opposite procedure

1. **Fourier transform**: $(r, \ell, m) \quad \rightarrow \quad (r, \ell, \phi)$
2. **Legendre transform**: $(r, \ell, \phi) \quad \rightarrow \quad (r, \theta, \phi)$

A bit more on Legendre transforms...

- **No fast Legendre transform** available: $\mathcal{O}(N_\theta^2)$ for one transform!

$$(r, \theta, \phi) \quad \rightarrow \quad (r, \ell, m) \quad \Longrightarrow \quad \mathcal{O}(N_r N_\phi N_\theta^2)$$

- But “savings”: Y_ℓ^m symmetries (only half of the colatitudes required), polar optimisations, ...
- **SHTns** is a high-performance library for SH transforms (https://bitbucket.org/nschaeff/shtns). It can be used in MagIC and provide a significant speed-up for large truncations.
- **Triangular truncation** provides a balanced spatial resolution over the spherical surface $\rightarrow N_\phi = 2N_\theta$

Avoid aliasing problems

Integration of quadratic terms on a discrete grid yields:

$$uv = \sum_{p=-K}^{K} a_p \, e^{ipx} \sum_{q=-K}^{K} a_q \, e^{iqx}$$

$$= \sum_{k=-2K}^{2K} b_k \, e^{ikx}$$

Avoid aliasing problems

Integration of quadratic terms on a discrete grid yields:

$$uv = \sum_{p=-K}^{K} a_p\, e^{ipx} \sum_{q=-K}^{K} a_q\, e^{iqx}$$

$$= \sum_{k=-2K}^{2K} b_k\, e^{ikx}$$

Alias-free SH transform

Orszag's (1971) 2/3 dealiasing rule: *"to obtain an alias-free computation on a grid of N points for a quadratically nonlinear equation, filter the high wavenumbers so as to retain only $(2/3)N$ unfiltered wavenumbers."* (Boyd 2001)

$$N_\theta \geq \frac{3\ell_{\max}+1}{2}$$

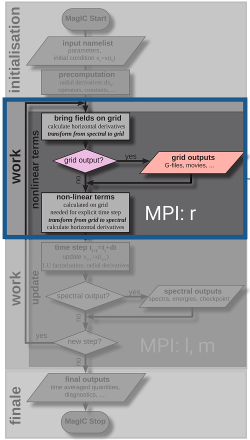
Spherical harmonic transforms: summary

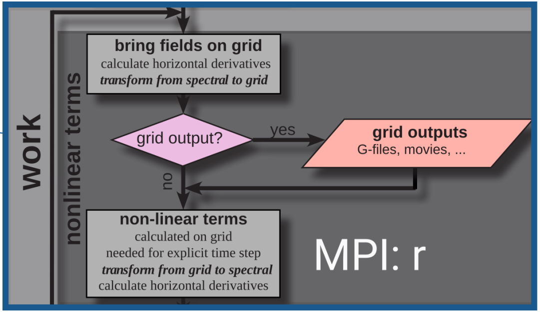
Take-away messages on SH transforms

- **Spectral to spatial** $(r, \ell, m) \rightarrow (r, \theta, \phi)$: Fourier and Legendre transforms
- **Spatial to spectral** $(r, \theta, \phi) \rightarrow (r, \ell, m)$: inverse Fourier and Legendre transforms
- FFT: $\mathcal{O}(\mathbf{N_r N_\theta N_\phi} \log(\mathbf{N_\phi}))$
- Legendre transform represents the most important part of the spherical harmonic transform: $\mathcal{O}(\mathbf{N_r N_\phi N_\theta^2})$
- FFT: prime decomposition of N_ϕ should only contain multiple of 2, 3 and 5 (for built-in FFT)
- Alias-free SH transforms require: $N_\theta \geq \dfrac{3\ell_{max}+1}{2}$

Documentation

MagIC structure





Outline

- MagIC simulates rotating fluid dynamics in a spherical shell
- It solves for the coupled evolution of Navier-Stokes equation, MHD equation, temperature (or entropy) equation and an equation for chemical composition under both the anelastic and the Boussinesq approximations
- A dimensionless formulation of the equations is assumed
- MagIC is a free software (GPL), written in Fortran
- Post-processing relies on python libraries
- Poloidal/toroidal decomposition is employed
- MagIC uses spherical harmonic decomposition in the angular directions
- **Chebyshev polynomials or finite differences are employed in the radial direction**
- MagIC uses a mixed implicit/explicit time stepping scheme
- The code relies on a hybrid parallelisation scheme (MPI/OpenMP)

Radial representation in spherical shell codes

Different approaches have been employed to represent the radial variation of the unknowns:

- calypso, Parody, xshells, ... : **finite differences** (usually 2nd order)
- ASH, Rayleigh, ... : expansion in **Chebyshev polynomials**.
- MagIC: since version 5.6: both FD and Chebyshev polynomials are supported.

Special focus on spectral method here...

Some mathematical properties of Chebyshev polynomials

- The Chebyshev polynomial of degree n is defined by:

$$\mathcal{C}_n(x) = \cos[n \arccos(x)], \quad -1 < x < 1$$

- Recursion relation:

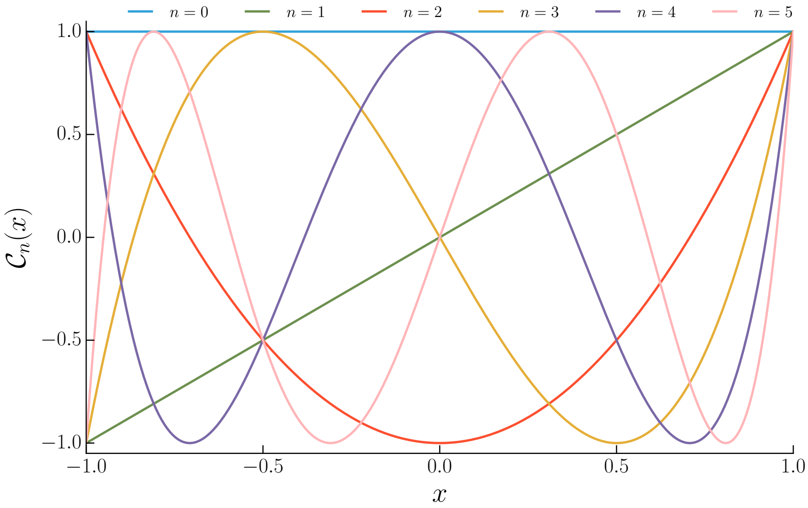
$$\mathcal{C}_{n+1}(x) = 2\, x\, \mathcal{C}_n(x) - \mathcal{C}_{n-1}(x)$$

- Derivatives

$$\frac{\mathrm{d}\mathcal{C}_{n+1}}{\mathrm{d}x} = 2\, \mathcal{C}_n + 2\, x\, \frac{\mathrm{d}\mathcal{C}_n}{\mathrm{d}x} - \frac{\mathrm{d}\mathcal{C}_{n-1}}{\mathrm{d}x}$$

$$\frac{\mathrm{d}^2\mathcal{C}_{n+1}}{\mathrm{d}x^2} = 4\frac{\mathrm{d}\mathcal{C}_n}{\mathrm{d}x} + 2\, x\, \frac{\mathrm{d}^2\mathcal{C}_n}{\mathrm{d}x^2} - \frac{\mathrm{d}^2\mathcal{C}_{n-1}}{\mathrm{d}x^2}$$

First Chebyshev polynomials



Gauss-Lobatto grid points (suitable for boundary layers and “FFT-friendly”):

$$x_k = \cos\left(\frac{k\pi}{N}\right), \quad k = 0, 2, \cdots, N$$

This yields

$$\mathcal{C}_n(x_k) = \cos\left(\frac{n\,k\,\pi}{N}\right)$$

The Gauss-Lobatto grid points are **linearly mapped** on a $[r_i, r_o]$ grid:

$$r_k = r_i + \frac{r_o - r_i}{2}\left(1 + \cos\left[\frac{k\pi}{N}\right]\right)$$

N.B Additional nonlinear mappings can be used to modify the grid-point density

Radial representation (1/2)

Truncating the radial expansion of the toroidal flow potential at degree N reads:

$$Z_\ell^m(r_k, t) = \sum_{n=0}^{N} Z_{\ell n}^m(t)\,\mathcal{C}_n(r_k)$$

with

$$Z_{\ell n}^m(t) = \frac{2 - \delta_{n0} - \delta_{nN}}{\pi} \int_{-1}^{1} \frac{Z_\ell^m(r(x), t)\,\mathcal{C}_n(x)\,\mathrm{d}x}{\sqrt{1 - x^2}}$$

Radial representation (1/2)

Truncating the radial expansion of the toroidal flow potential at degree N reads:

$$Z_\ell^m(r_k, t) = \sum_{n=0}^{N} Z_{\ell n}^m(t)\, \mathcal{C}_n(r_k)$$

with

$$Z_{\ell n}^m(t) = \frac{2 - \delta_{n0} - \delta_{nN}}{\pi} \int_{-1}^{1} \frac{Z_\ell^m(r(x), t)\, \mathcal{C}_n(x)\, \mathrm{d}x}{\sqrt{1 - x^2}}$$

At this stage, we make use of the Gaussian quadrature rule:

$$\int_{-1}^{1} f(x) w(x)\, \mathrm{d}x = \sum_{n=0}^{N} w_n\, f(x_n)$$

Radial representation (2/2)

Using the Gauss-Lobatto grid with $x_n = \cos(n\pi/N)$ gives (e.g. Abramowitz & Stegun)

$$w_j = \begin{cases} \dfrac{\pi}{N} & i = 1, 2, \cdots, N-1 \\ \dfrac{\pi}{2N} & i = 0, N \end{cases}$$

Radial representation (2/2)

Using the Gauss-Lobatto grid with $x_n = \cos(n\pi/N)$ gives (e.g. Abramowitz & Stegun)

$$w_j = \begin{cases} \dfrac{\pi}{N} & i = 1, 2, \cdots, N-1 \\ \dfrac{\pi}{2N} & i = 0, N \end{cases}$$

This finally yields

From real to Chebyshev space

$$Z_{\ell n}^m(t) = \frac{1}{2N}\left[Z_\ell^m(r_0, t) + Z_\ell^m(r_N, t) + 2\sum_{n=1}^{N-1} Z_\ell^m(r_n, t)\,\cos\left(\frac{n\,k\,\pi}{N}\right)\right]$$

This is a **fast discrete cosine transform**: this forces us to use some "FFT-friendly" number of radial grid points.

Chebyshev polynomials: summary

Take-away messages on Chebyshev polynomials

- **Gauss-Lobatto grid**: boundary layer refinement and "FFT-friendly"
- **Chebyshev space to grid** $n \rightarrow r$: discrete cosine transform
- **Grid to Chebyshev space** $r \rightarrow n$: discrete cosine transform
- DCT: $\mathcal{O}(\mathbf{N_r} \log(\mathbf{N_r}))$
- DCT: prime decomposition of $N_r - 1$ should only contain multiple of 2, 3 and 5

Documentation

Outline

Spectral poloidal dynamo equation (1/4)

- All the necessary tools to derive the spectral equations have been introduced
- As an example, I focus here on the derivation of the equation for the **poloidal magnetic field potential**:

$$\frac{\partial \mathbf{B}}{\partial t} = \boldsymbol{\nabla} \left(\mathbf{u} \times \mathbf{B} \right) + \frac{1}{Pm} \boldsymbol{\Delta} \mathbf{B}$$

To derive the equation for $g_{\ell n}^m$, take the **radial component** of the induction equation

Spectral poloidal dynamo equation (2/4)

Time derivative

Time derivative:

$$\mathbf{e_r} \cdot \frac{\partial \mathbf{B}}{\partial t} = \frac{\partial B_r}{\partial t}$$

Spectral poloidal dynamo equation (2/4)

Time derivative

Time derivative:

$$\mathbf{e_r} \cdot \frac{\partial \mathbf{B}}{\partial t} = \frac{\partial B_r}{\partial t}$$

We have

$$B_r(r,\theta,\phi,t) = -\Delta_H\, g = \sum_{\ell,m} \frac{\ell(\ell+1)}{r^2} g_\ell^m(r,t) Y_\ell^m(\theta,\phi)$$

Spectral poloidal dynamo equation (2/4)

Time derivative

Time derivative:

$$\mathbf{e_r} \cdot \frac{\partial \mathbf{B}}{\partial t} = \frac{\partial B_r}{\partial t}$$

We have

$$B_r(r,\theta,\phi,t) = -\Delta_H\, g = \sum_{\ell,m} \frac{\ell(\ell+1)}{r^2} g_\ell^m(r,t) Y_\ell^m(\theta,\phi)$$

Hence

$$\boxed{\mathbf{e_r} \cdot \frac{\partial \mathbf{B}}{\partial t} = \sum_{\ell,m} \frac{\ell(\ell+1)}{r^2} \frac{\partial g_\ell^m}{\partial t} Y_\ell^m}$$

Spectral poloidal dynamo equation (3/4)

Diffusion term

Same procedure:

$$\begin{aligned}\mathbf{e_r}\cdot\left(\frac{1}{Pm}\boldsymbol{\Delta}\mathbf{B}\right) &= \frac{1}{Pm}\left(\Delta B_r - \frac{2}{r^2}B_r - \frac{2}{r}\boldsymbol{\nabla}_H\cdot\mathbf{B}\right)\\ &= \frac{1}{Pm}\left(\Delta B_r - \frac{2}{r^2}B_r - \underbrace{\boldsymbol{\nabla}\cdot\mathbf{B}}_{=0} + \frac{2}{r^3}\frac{\partial}{\partial r}(r^2 B_r)\right)\\ &= \frac{1}{Pm}\left(\frac{1}{r^2}\frac{\partial^2(r^2 B_r)}{\partial r^2} + \Delta_H B_r\right)\end{aligned}$$

Spectral poloidal dynamo equation (3/4)

Diffusion term

Same procedure:

$$\mathbf{e_r}\cdot\left(\frac{1}{Pm}\mathbf{\Delta B}\right) = \frac{1}{Pm}\left(\Delta B_r - \frac{2}{r^2}B_r - \frac{2}{r}\boldsymbol{\nabla}_H\cdot\mathbf{B}\right)$$

$$= \frac{1}{Pm}\left(\Delta B_r - \frac{2}{r^2}B_r - \underbrace{\boldsymbol{\nabla}\cdot\mathbf{B}}_{=0} + \frac{2}{r^3}\frac{\partial}{\partial r}(r^2 B_r)\right)$$

$$= \frac{1}{Pm}\left(\frac{1}{r^2}\frac{\partial^2(r^2 B_r)}{\partial r^2} + \Delta_H B_r\right)$$

Hence

$$\boxed{\mathbf{e_r}\cdot\left(\frac{1}{Pm}\mathbf{\Delta B}\right) = \frac{1}{Pm}\sum_{\ell,m}\frac{\ell(\ell+1)}{r^2}\left(\frac{\partial^2 g_\ell^m}{\partial r^2} - \frac{\ell(\ell+1)}{r^2}g_\ell^m\right)Y_\ell^m}$$

Spectral poloidal dynamo equation (4/4)

Now mulitply by Y_ℓ^{m*} and expand in Chebyshev polynomials:

$$\frac{\ell(\ell+1)}{r^2}\left[\left(\frac{\partial}{\partial t}+\frac{1}{Pm}\frac{\ell(\ell+1)}{r^2}\right)\mathcal{C}_n-\frac{1}{Pm}\mathcal{C}_n''\right]g_{\ell n}^m=\int(\mathbf{e_r}\cdot\boldsymbol{\mathcal{D}})\,Y_\ell^{m*}\,\mathrm{d}\Omega$$

where $\boldsymbol{\mathcal{D}}$ is the nonlinear induction term expressed by

$$\boldsymbol{\mathcal{D}}=\boldsymbol{\nabla}\times(\mathbf{u}\times\mathbf{B})$$

Spectral poloidal dynamo equation (4/4)

Now mulitply by Y_ℓ^{m*} and expand in Chebyshev polynomials:

$$\frac{\ell(\ell+1)}{r^2}\left[\left(\frac{\partial}{\partial t}+\frac{1}{Pm}\frac{\ell(\ell+1)}{r^2}\right)\mathcal{C}_n-\frac{1}{Pm}\mathcal{C}''_n\right]g_{\ell n}^m=\int(\mathbf{e_r}\cdot\boldsymbol{\mathcal{D}})\,Y_\ell^{m*}\,\mathrm{d}\Omega$$

where $\boldsymbol{\mathcal{D}}$ is the nonlinear induction term expressed by

$$\boldsymbol{\mathcal{D}}=\boldsymbol{\nabla}\times(\mathbf{u}\times\mathbf{B})$$

How do we treat this remaining term?

Solving the nonlinear terms

1. Calculate the horizontal component of EMF $\boldsymbol{\mathcal{F}} = \mathbf{u} \times \mathbf{B}$ **on physical grid**

$$\mathcal{F}_\theta = u_\phi B_r - u_r B_\phi; \quad \mathcal{F}_\phi = u_r B_\theta - u_\theta B_r$$

such that

$$\mathcal{N}_g = \mathbf{e_r} \cdot \boldsymbol{\mathcal{D}} = \frac{1}{r \sin\theta} \left[\frac{\partial(\sin\theta\, \mathcal{F}_\phi)}{\partial\theta} - \frac{\partial \mathcal{F}_\theta}{\partial\phi} \right]$$

Solving the nonlinear terms

1. Calculate the horizontal component of EMF $\boldsymbol{\mathcal{F}} = \mathbf{u} \times \mathbf{B}$ **on physical grid**

$$\mathcal{F}_\theta = u_\phi B_r - u_r B_\phi; \quad \mathcal{F}_\phi = u_r B_\theta - u_\theta B_r$$

such that

$$\mathcal{N}_g = \mathbf{e_r} \cdot \boldsymbol{\mathcal{D}} = \frac{1}{r \sin\theta} \left[\frac{\partial(\sin\theta\, \mathcal{F}_\phi)}{\partial\theta} - \frac{\partial \mathcal{F}_\theta}{\partial\phi} \right]$$

2. **Transform to spectral space**:

$$\mathcal{F}_\theta(\theta,\phi) \xrightarrow{\text{FFT, Leg.}} \hat{\mathcal{F}}_{\theta\ell}^{\,m}; \quad \mathcal{F}_\phi(\theta,\phi) \xrightarrow{\text{FFT, Leg.}} \hat{\mathcal{F}}_{\phi\ell}^{\,m}$$

Solving the nonlinear terms

1. Calculate the horizontal component of EMF $\boldsymbol{\mathcal{F}} = \mathbf{u} \times \mathbf{B}$ **on physical grid**

$$\mathcal{F}_\theta = u_\phi B_r - u_r B_\phi; \quad \mathcal{F}_\phi = u_r B_\theta - u_\theta B_r$$

such that

$$\mathcal{N}_g = \mathbf{e_r} \cdot \boldsymbol{\mathcal{D}} = \frac{1}{r \sin\theta} \left[\frac{\partial(\sin\theta\, \mathcal{F}_\phi)}{\partial\theta} - \frac{\partial \mathcal{F}_\theta}{\partial\phi} \right]$$

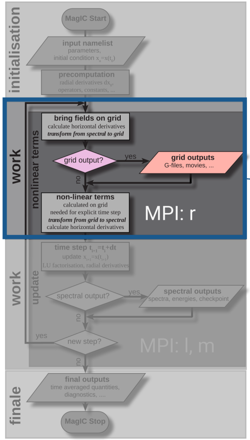
2. **Transform to spectral space**:

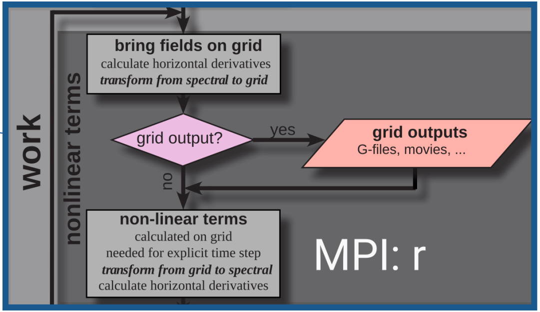
$$\mathcal{F}_\theta(\theta,\phi) \xrightarrow{\text{FFT, Leg.}} \hat{\mathcal{F}}_{\theta\ell}^{\,m}; \quad \mathcal{F}_\phi(\theta,\phi) \xrightarrow{\text{FFT, Leg.}} \hat{\mathcal{F}}_{\phi\ell}^{\,m}$$

3. **Calculate θ and ϕ derivatives using recurrence relations**:

$$\boxed{\mathcal{N}_\ell^m = (\ell+1) c_\ell^m \hat{\mathcal{F}}_{\phi\ell-1}^{\,m} - \ell c_{\ell+1}^m \hat{\mathcal{F}}_{\phi\ell+1}^{\,m} - im\, \hat{\mathcal{F}}_{\theta\ell}^{\,m}}$$

MagIC structure





Spectral poloidal dynamo equation

Equation for each spherical harmonic degree and order

$$\frac{\ell(\ell+1)}{r^2}\left[\left(\frac{\partial}{\partial t}+\frac{1}{Pm}\frac{\ell(\ell+1)}{r^2}\right)\mathcal{C}_n-\frac{1}{Pm}\mathcal{C}''_n\right]g^m_{\ell n}=\mathcal{N}^m_\ell$$

▸ Documentation

Spectral poloidal dynamo equation

Equation for each spherical harmonic degree and order

$$\frac{\ell(\ell+1)}{r^2}\left[\left(\frac{\partial}{\partial t}+\frac{1}{Pm}\frac{\ell(\ell+1)}{r^2}\right)\mathcal{C}_n-\frac{1}{Pm}\mathcal{C}''_n\right]g_{\ell n}^m=\mathcal{N}_\ell^m$$

General recipe

- We proceed the same way to derive the other equations for W_ℓ^m, Z_ℓ^m, s_ℓ^m, h_ℓ^m, p_ℓ^m
- Nonlinear terms are treated on the grid, linear terms in the spectral space (except Coriolis force, see after)
- **Each equation couples** $N+1$ **Chebyshev coefficients for a given spherical harmonic mode** (ℓ, m)

▸ Documentation

Mechanical boundary conditions

- **Impermeable boundaries** = zero radial flow on the boundaries:

$$u_r = 0 \quad \rightarrow \quad \mathcal{C}_n(r) W_{\ell n}^m = 0 \text{ at } r = r_i, r_o$$

- **Rigid boundaries** = no-slip boundary condition (velocity cancels out):

$$u_\theta = u_\phi = 0 \quad \rightarrow \quad \mathcal{C}'_n(r) W_{\ell n}^m = \mathcal{C}_n(r) Z_{\ell n}^m = 0 \text{ at } r = r_i, r_o$$

- Or **stress-free** boundary conditions:

$$\begin{matrix} \dfrac{\partial}{\partial r}\left(\dfrac{u_\theta}{r}\right) = 0 \\ \dfrac{\partial}{\partial r}\left(\dfrac{u_\phi}{r}\right) = 0 \end{matrix} \rightarrow \left\{ \begin{aligned} \left[\mathcal{C}''_n(r) - \left(\frac{2}{r} + \mathcal{L}_\rho\right)\mathcal{C}'_n(r)\right] W_{\ell n}^m = & \quad 0 \\ \left[\mathcal{C}'_n(r) - \left(\frac{2}{r} + \mathcal{L}_\rho\right)\mathcal{C}_n(r)\right] Z_{\ell n}^m = & \quad 0 \end{aligned} \right. \text{ at } r = r_i, r_o$$

$$\text{with } \mathcal{L}_\rho \equiv \frac{\mathrm{d} \ln \tilde{\rho}}{\mathrm{d} r}$$

Magnetic boundary conditions

- **Insulating (vacuum)** boundary condition = toroidal field cannot enter an insulator (no current):

$$\mathbf{B} = -\nabla\Phi \quad \rightarrow \quad \mathcal{C}_n(r)\, h_{\ell n}^m = 0 \text{ at } r = r_i, r_o$$

- Matching condition for the poloidal field:

$$\mathbf{B} = -\nabla\Phi \rightarrow \begin{cases} \left[\mathcal{C}_n'(r) + \dfrac{\ell+1}{r}\mathcal{C}_n(r)\right] g_{\ell n}^m = & 0 \text{ at } r = r_i \\ \left[\mathcal{C}_n'(r) + \dfrac{\ell}{r}\mathcal{C}_n(r)\right] g_{\ell n}^m = & 0 \text{ at } r = r_o \end{cases}$$

- Other possible boundary conditions: **pseudo-vacuum, conducting inner core**, ...

Thermal boundary conditions

- **Constant entropy** (or constant temperature):

$$s' \text{ (or } T') = 0 \quad \rightarrow \quad \mathcal{C}_n(r)\, s'^{m}_{\ell n} = 0 \text{ at } r = r_i, r_o$$

- **Constant entropy flux** (or constant temperature flux):

$$\frac{\partial s'}{\partial r} \left(\text{or } \frac{\partial T'}{\partial r} \right) = 0 \quad \rightarrow \quad \mathcal{C}'_n(r)\, s'^{m}_{\ell n} = 0 \text{ at } r = r_i, r_o$$

- On top of that, **heterogeneous thermal boundary conditions** can be produced by imposing a suitable combination of (ℓ, m) modes...

Documentation