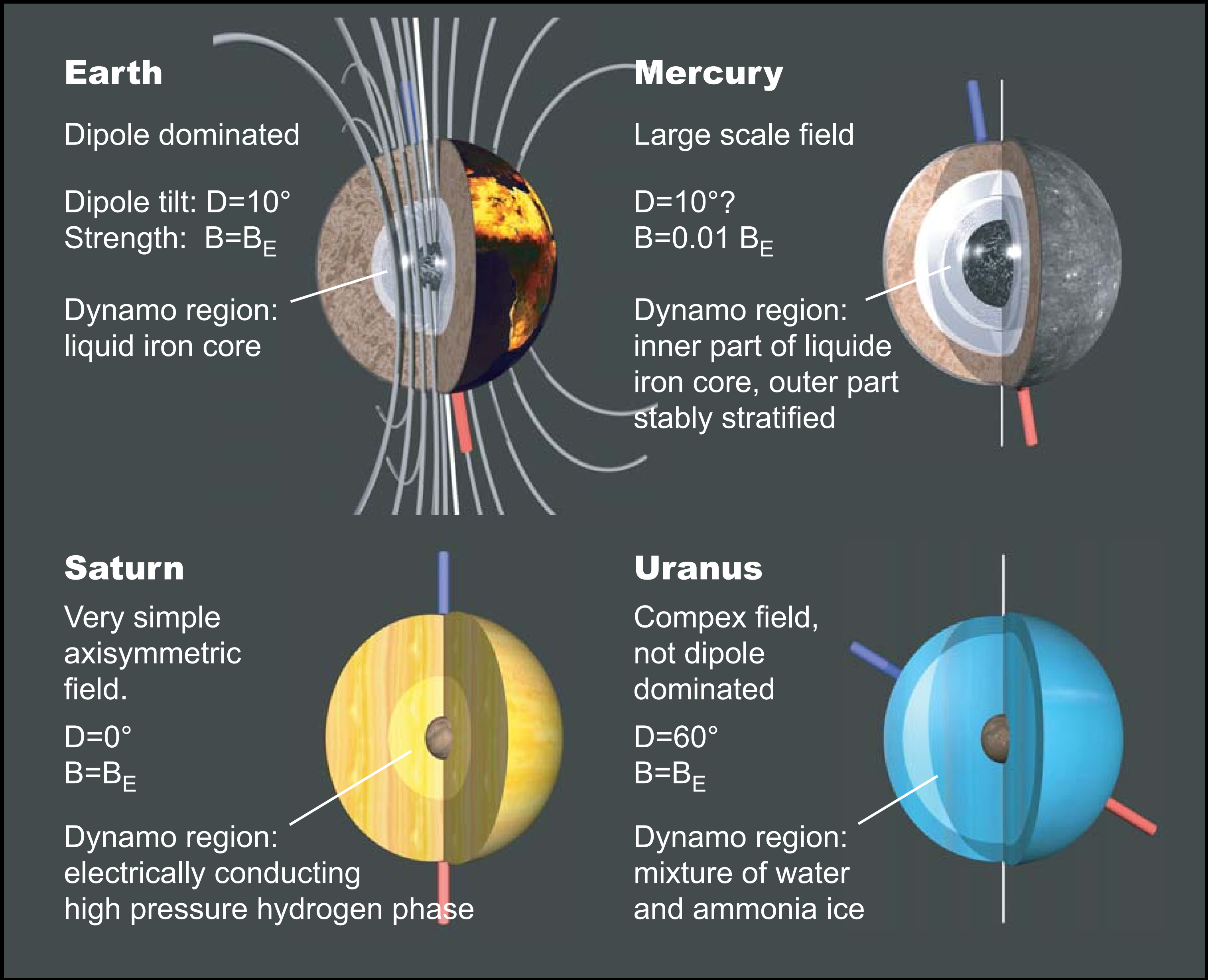
The Spherical MHD Code MagIC, Fundamentals

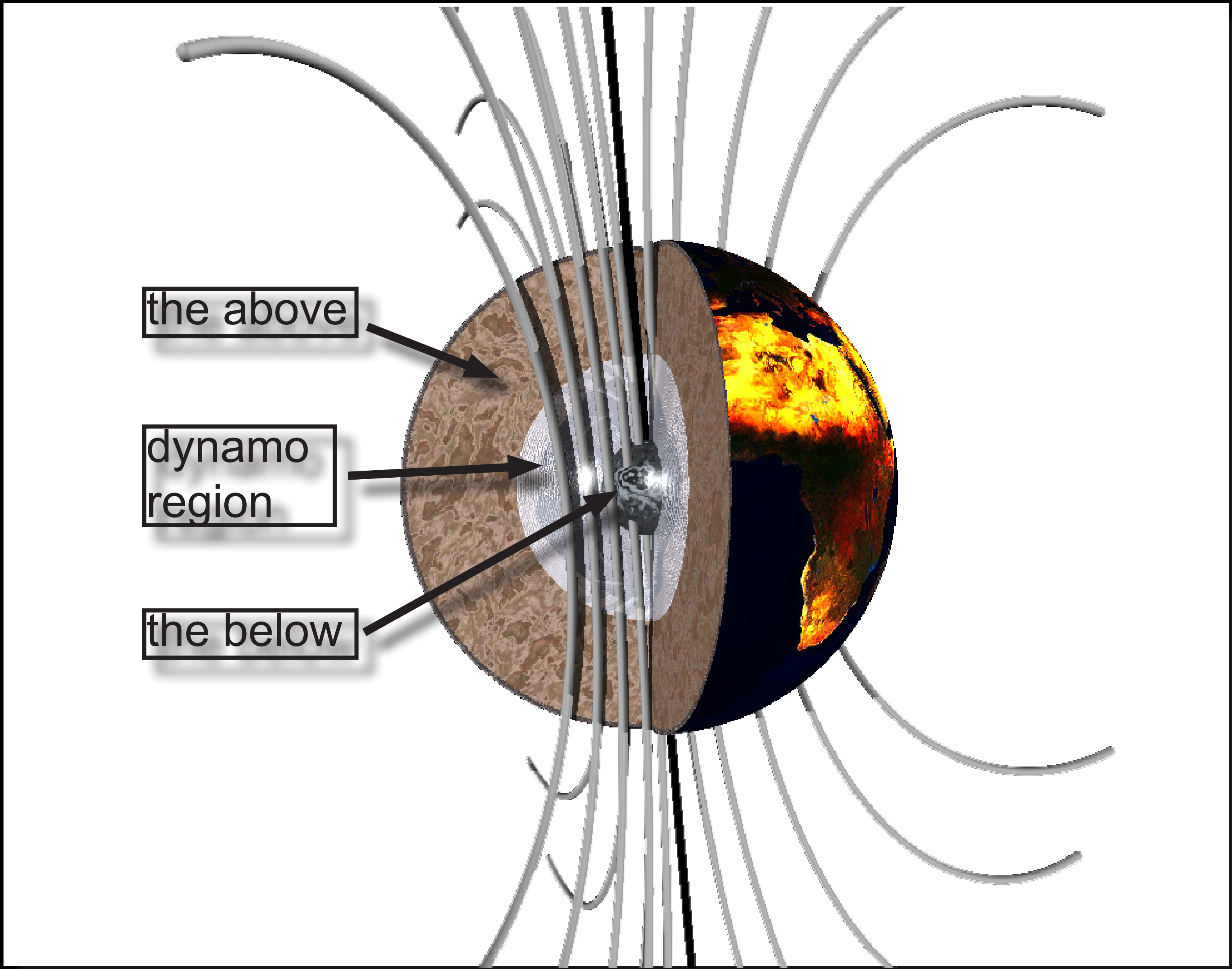
Johannes Wicht, Thomas Gasting, Ankit Barik
MPS Katlenburg-Lindau

The Goal

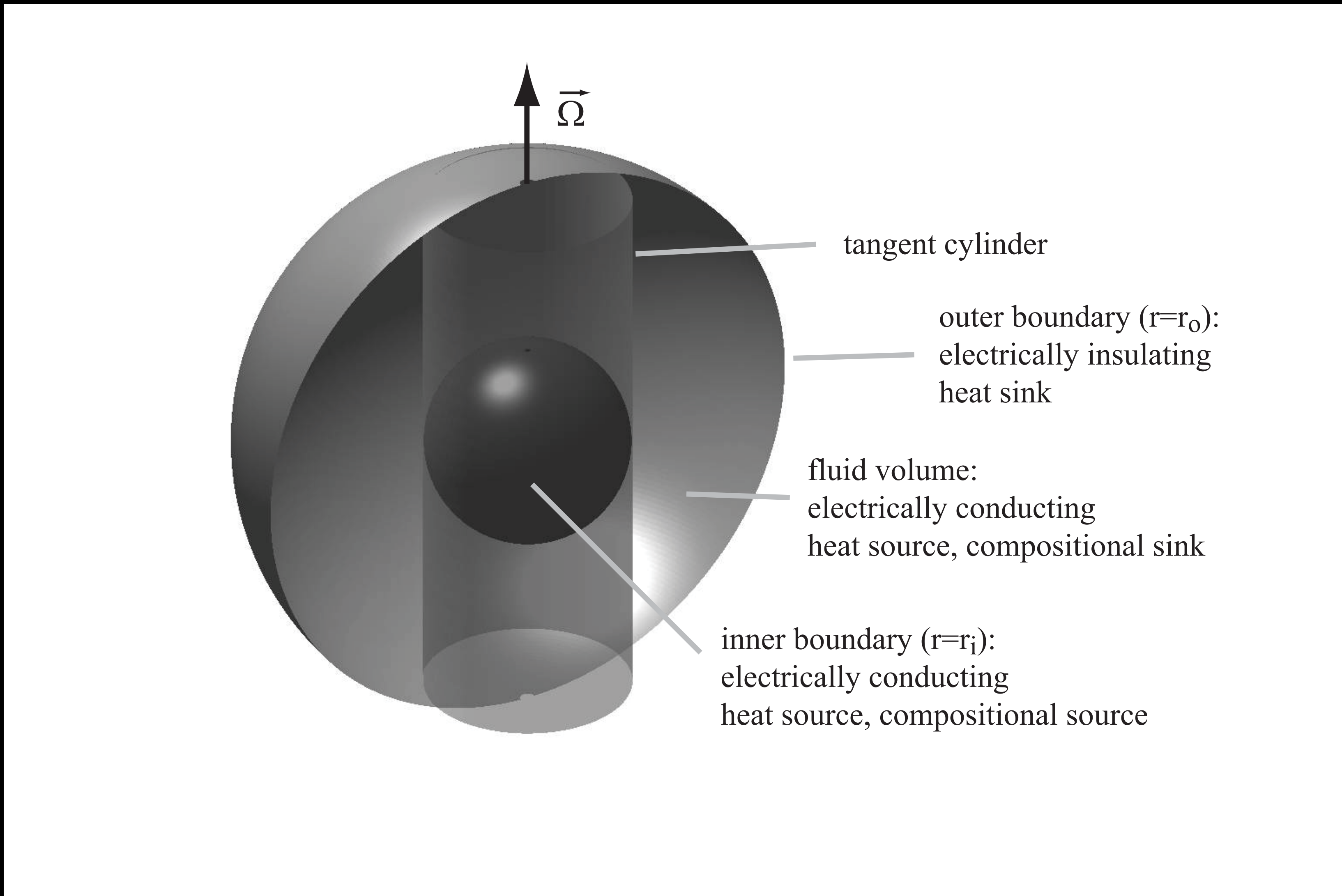


Earth
Dipole dominated
Dipole tilt: D=10°
Strength: $B=B_E$
Dynamo region:
liquid iron core
Mercury
Large scale field
D=10°?
$B=0.01\ B_E$
Dynamo region:
inner part of liquide
iron core, outer part
stably stratified
Saturn
Very simple
axisymmetric
field.
D=0°
$B=B_E$
Dynamo region:
electrically conducting
high pressure hydrogen phase
Uranus
Compex field,
not dipole
dominated
D=60°
$B=B_E$
Dynamo region:
mixture of water
and ammonia ice

The Setup

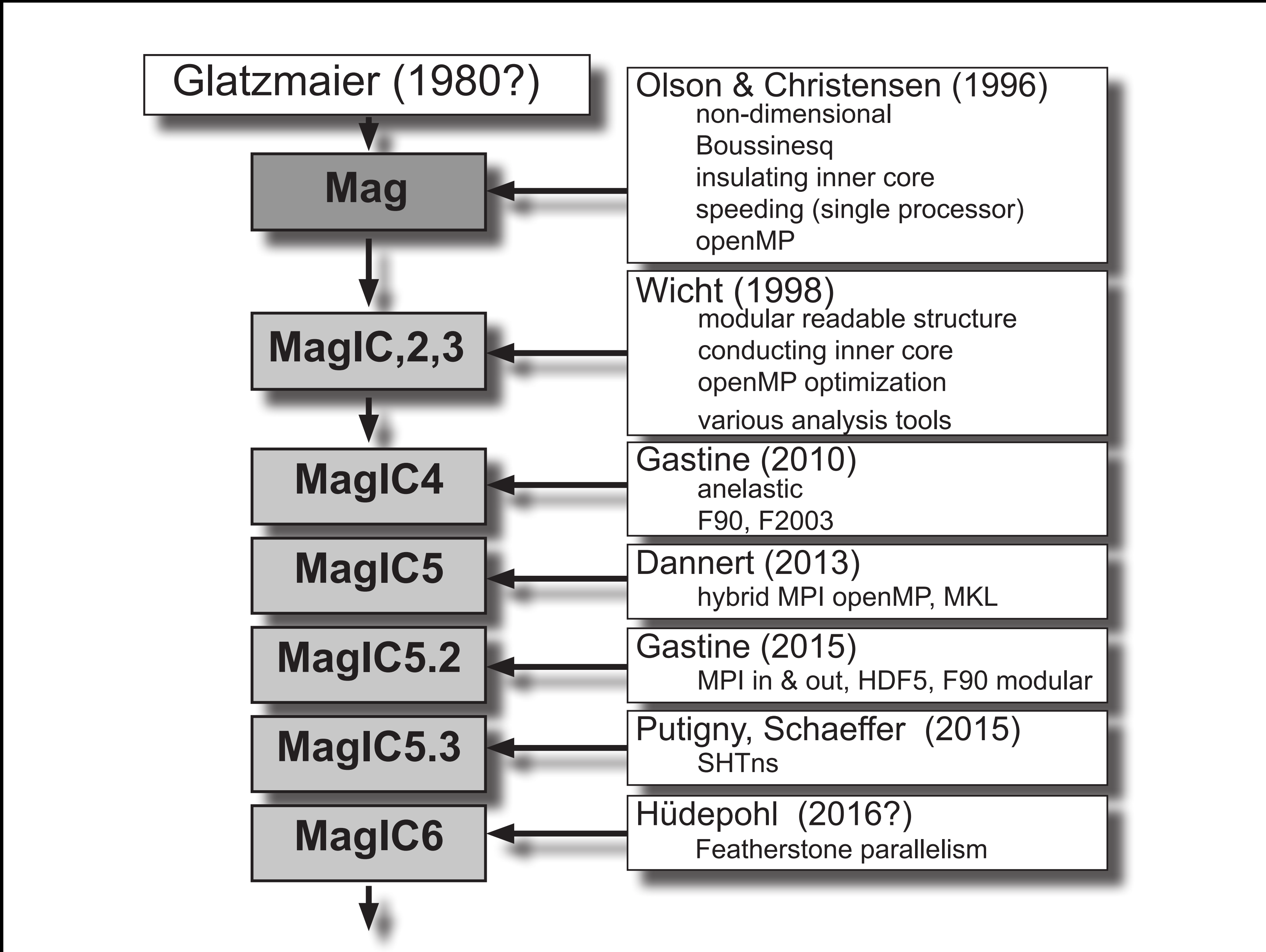


Setup



$\vec{\Omega}$
tangent cylinder
outer boundary ($r=r_o$):
electrically insulating
heat sink
fluid volume:
electrically conducting
heat source, compositional sink
inner boundary ($r=r_i$):
electrically conducting
heat source, compositional source

MagIC Heritage



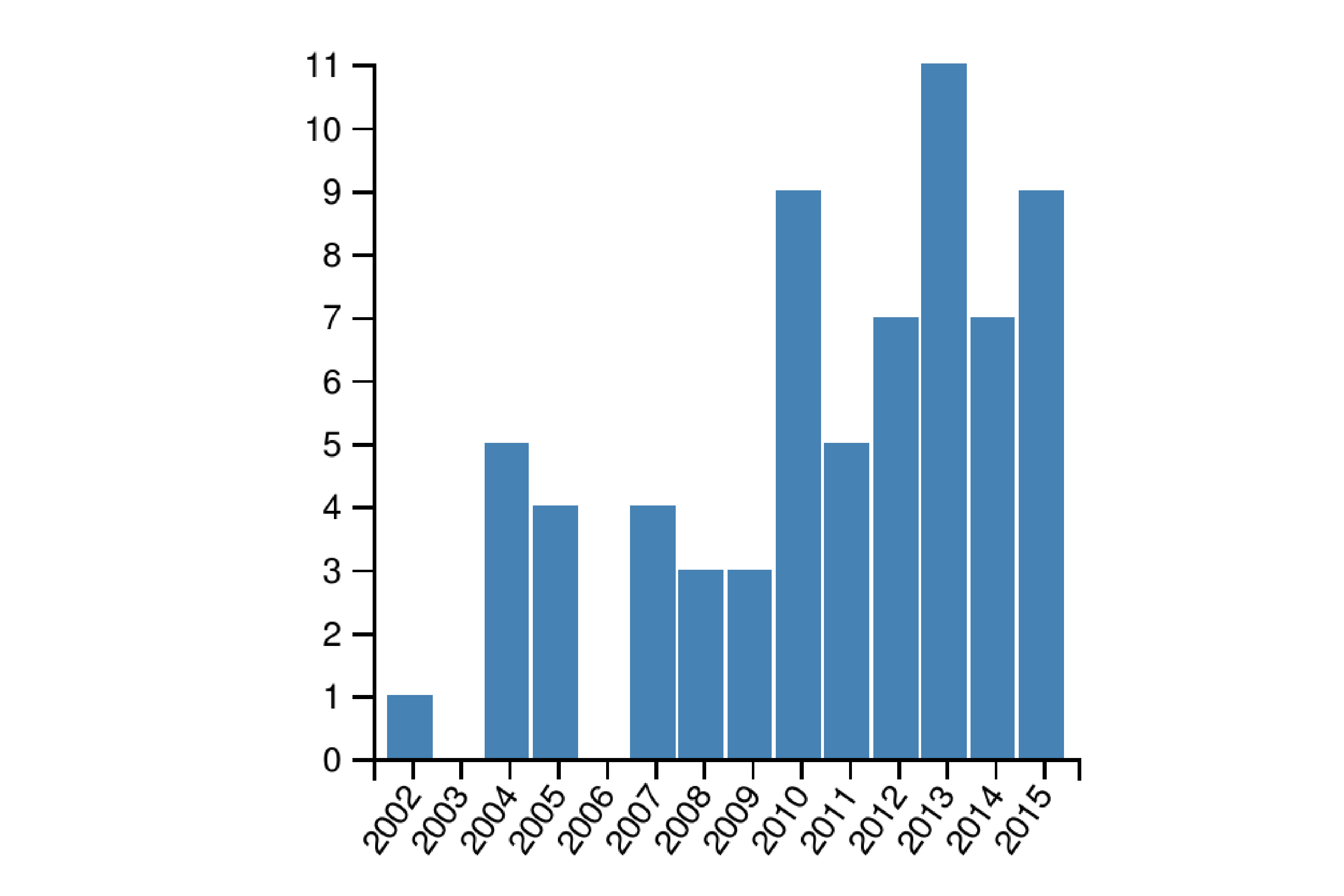
MagIC in Words

- MagIC solves for thermodynamic evolution, fluid flow and magnet field generation
- Domain = spherical shell
- Region below and above domain treated as boundary conditions or parametrized
- Frame of reference rotating with system rotation Ω
- MagIC uses a dimensionless formulation
- Poloidal/toroidal decomposition in employed
- MagIC is a pseudo spectral code
- MagIC uses a mixed impicite, explicite time stepping

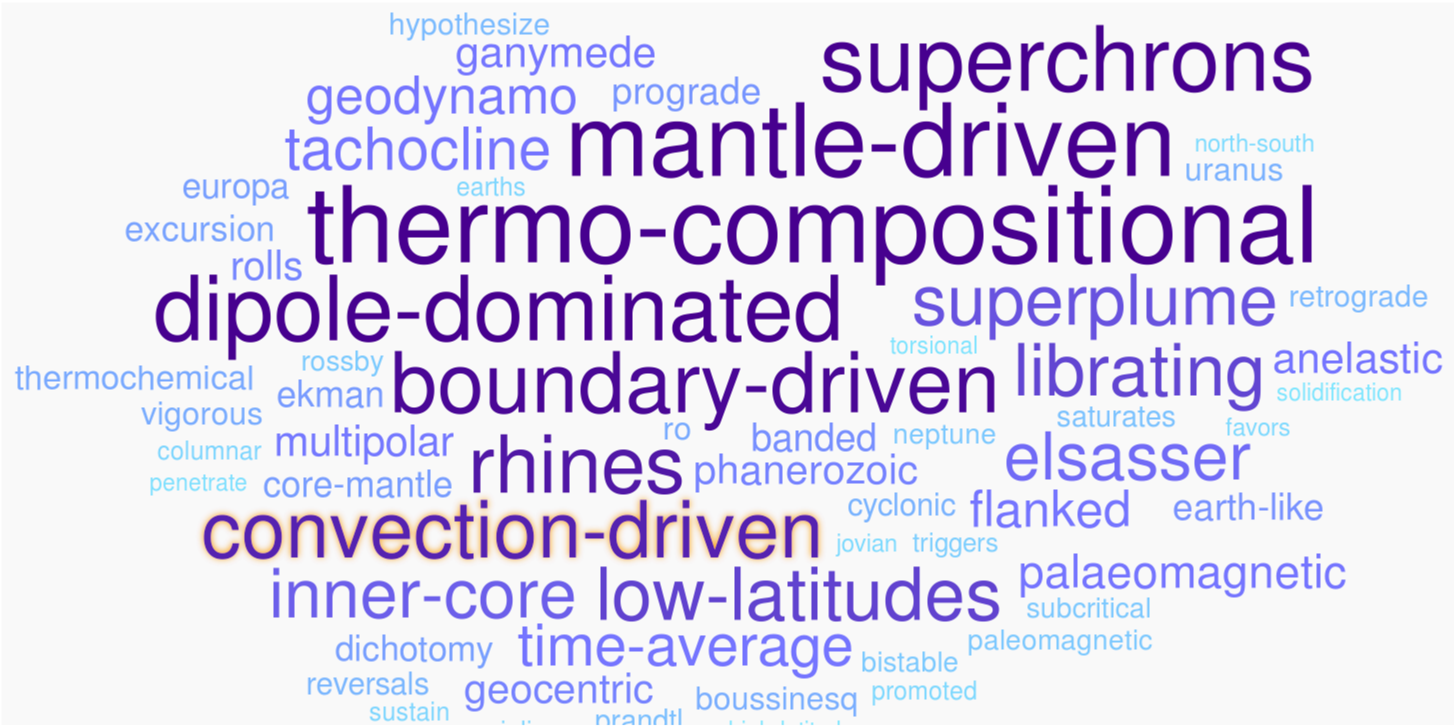
Why using MagIC?

- MagIC is open source, available on Githup.
- MagIC is the fastest code on up to 1000 cores.
  (See recent speed benchmark)
- MagIC is well documented. (Ankit, Thomas)
- MagIC has been tested extensively.
  (Autotest implemented by Thomas)
- MagIC offers a lot of usefull output and supporting analysis tools
- Matching visualization tools are available.
- MagIC has active and accessible users & developers.
- MagIC has been used in around 80 publications.

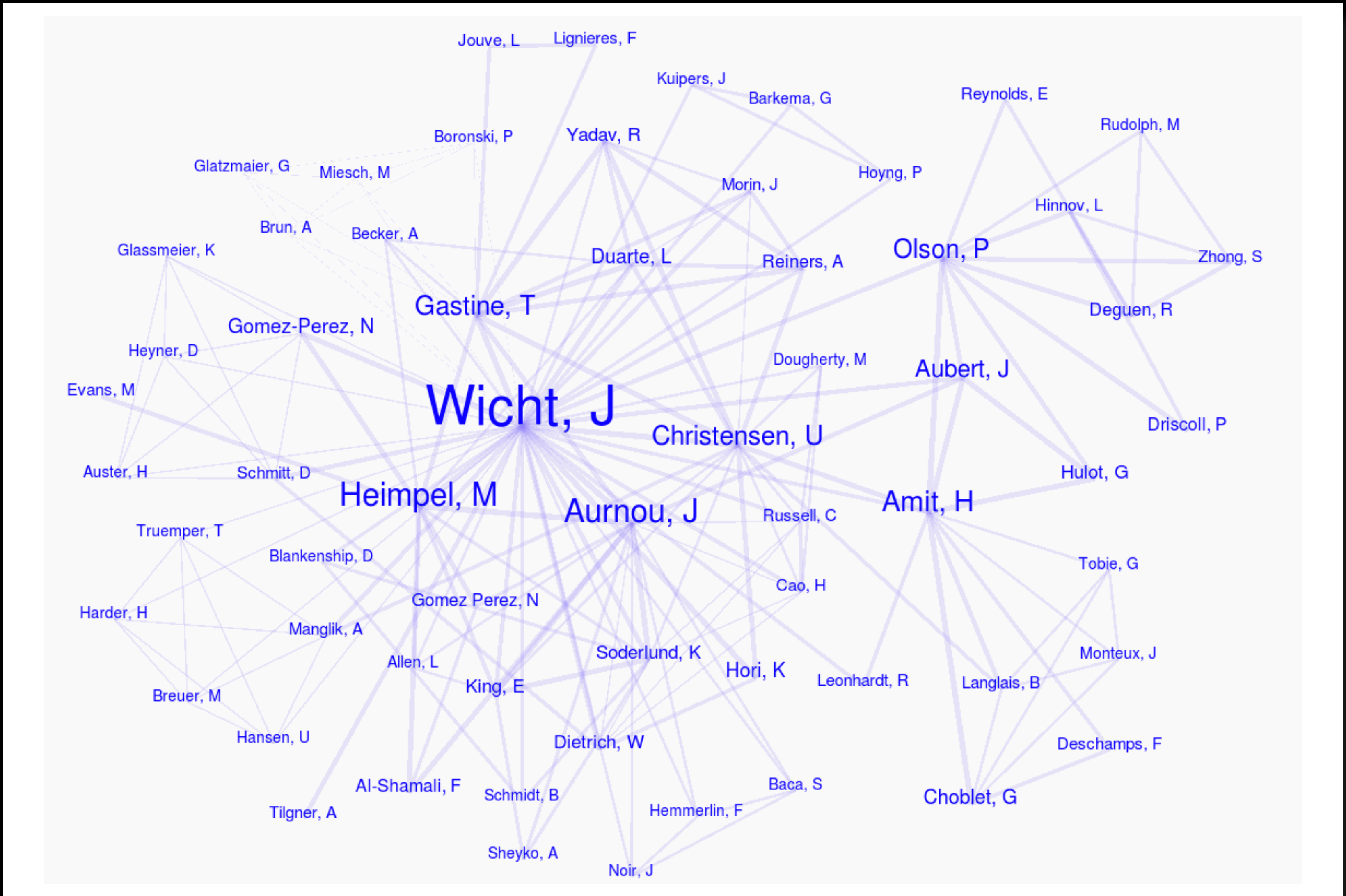
MagIC Success



MagIC Topics

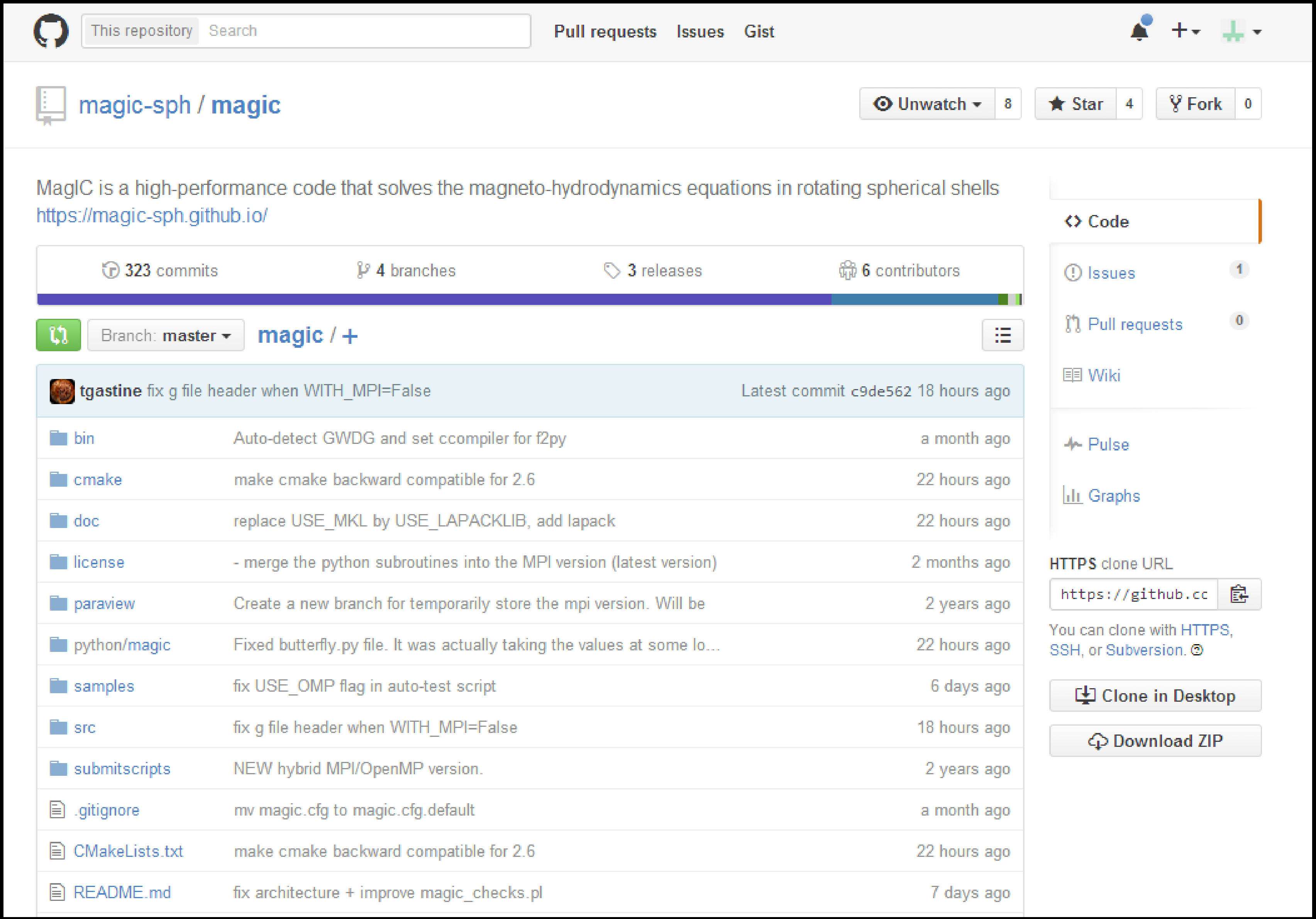


MagIC Users

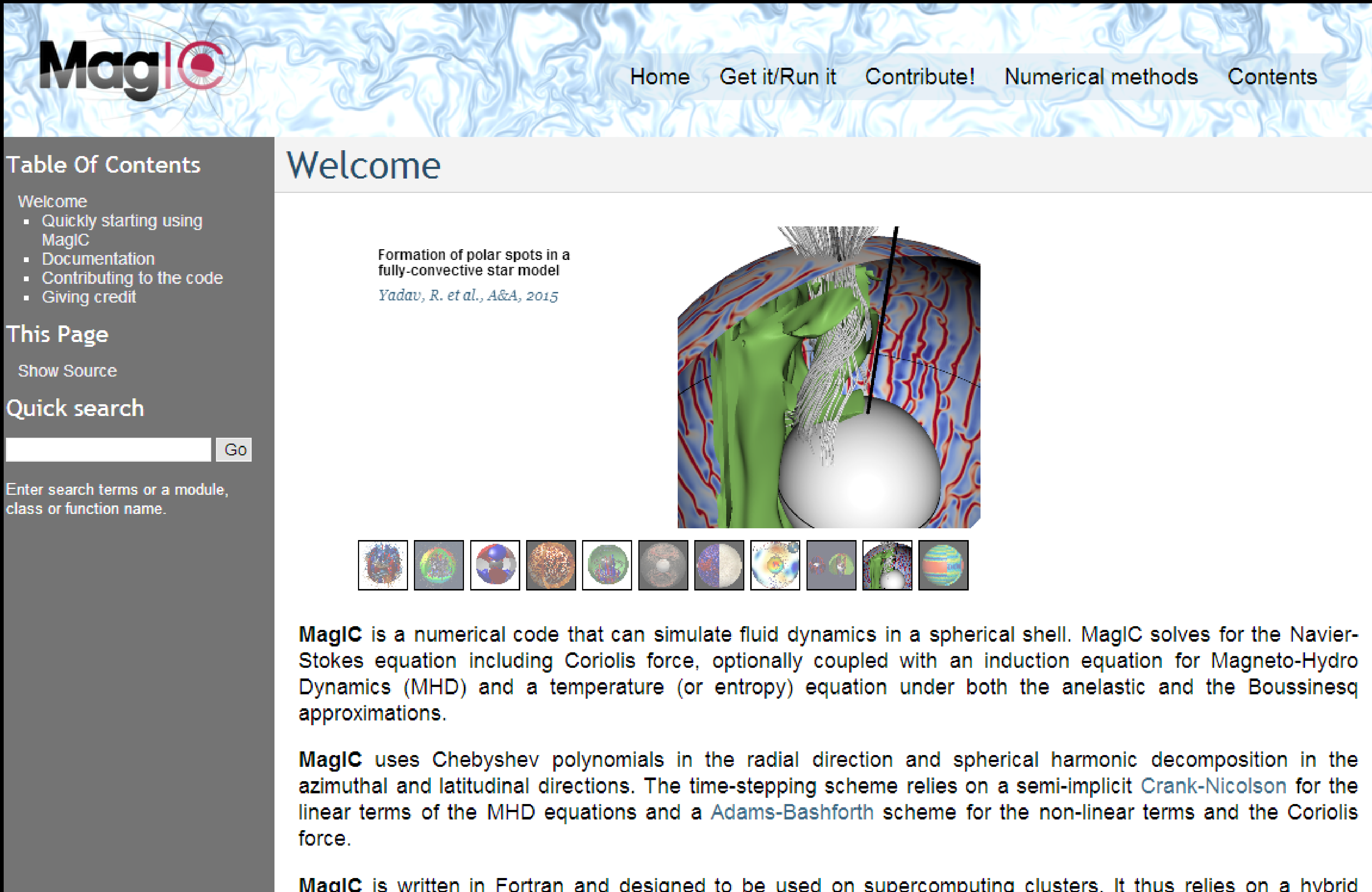


Jouve, L
Lignieres, F
Kuipers, J
Barkema, G
Reynolds, E
Rudolph, M
Boronski, P
Yadav, R
Glatzmaier, G
Miesch, M
Morin, J
Hoyng, P
Hinnov, L
Brun, A
Becker, A
Glassmeier, K
Duarte, L
Reiners, A
Olson, P
Zhong, S
Gastine, T
Deguen, R
Gomez-Perez, N
Heyner, D
Dougherty, M
Aubert, J
Evans, M
Wicht, J
Christensen, U
Driscoll, P
Auster, H
Schmitt, D
Heimpel, M
Aurnou, J
Hulot, G
Amit, H
Russell, C
Truemper, T
Blankenship, D
Tobie, G
Cao, H
Gomez Perez, N
Harder, H
Manglik, A
Allen, L
Soderlund, K
Monteux, J
Hori, K
Leonhardt, R
Langlais, B
King, E
Breuer, M
Hansen, U
Dietrich, W
Deschamps, F
Al-Shamali, F
Schmidt, B
Baca, S
Choblet, G
Tilgner, A
Hemmerlin, F
Sheyko, A
Noir, J

MagIC github



MagIC homepage



MagIC

Home | Get it/Run it | Contribute! | Numerical methods | Contents

Table Of Contents

- Welcome
  - Quickly starting using MagIC
  - Documentation
  - Contributing to the code
  - Giving credit

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

Welcome

Formation of polar spots in a fully-convective star model

Yadav, R. et al., A&A, 2015

MagIC is a numerical code that can simulate fluid dynamics in a spherical shell. MagIC solves for the Navier-Stokes equation including Coriolis force, optionally coupled with an induction equation for Magneto-Hydro Dynamics (MHD) and a temperature (or entropy) equation under both the anelastic and the Boussinesq approximations.

MagIC uses Chebyshev polynomials in the radial direction and spherical harmonic decomposition in the azimuthal and latitudinal directions. The time-stepping scheme relies on a semi-implicit Crank-Nicolson for the linear terms of the MHD equations and a Adams-Bashforth scheme for the non-linear terms and the Coriolis force.

MagIC is written in Fortran and designed to be used on supercomputing clusters. It thus relies on a hybrid

MagIC documentation

Home Get it/Run it Contribute! Numerical methods Contents

Magic 5.2 documentation » previous | next | modules | fortran modules | index

Table Of Contents

Previous topic

Get MagIC and run it

Formulation of the (magneto)-hydrodynamics problem

The general equations describing thermal convection and dynamo action of a rotating compressible fluid are the starting point from which the Boussinesq or the anelastic approximations are developed. In MagIC, we consider a spherical shell rotating about the vertical z axis with a constant angular velocity Ω. The conservation of mass is expressed by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \rho \vec{u} = 0, \tag{1}$$

The conservation of momentum by

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla}\, \vec{u} \right) = -\vec{\nabla} p + \frac{1}{\mu_0} (\vec{\nabla} \times \vec{B}) \times \vec{B} + \rho \vec{g} - 2\rho \vec{\Omega} \times \vec{u} + \vec{\nabla} \cdot \mathsf{S}, \tag{2}$$

where S corresponds to the rate-of-strain tensor given by:

$$S_{ij} = 2\nu\rho \left[e_{ij} - \frac{1}{3} \delta_{ij}\, \vec{\nabla} \cdot \vec{u} \right],$$

$$e_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

Concerning the energy equation, several forms are possible (using internal energy, temperature or entropy). Here we use entropy s as the main variable, which leads to:

$$\rho T \left(\frac{\partial s}{\partial t} + \vec{u} \cdot \vec{\nabla} s \right) = \vec{\nabla} \cdot (K \vec{\nabla} T) + \Phi_\nu + \lambda \left(\vec{\nabla} \times \vec{B} \right)^2, \tag{3}$$

where Φ_ν corresponds to the viscous heating expressed by

Equation of motion

Navier-Stokes equation:
described changes of momentum density at a given position due to forces

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\,\boldsymbol{u}\right) = -\boldsymbol{\nabla} p + \frac{1}{\mu_0}(\boldsymbol{\nabla}\times\boldsymbol{B})\times\boldsymbol{B} + \rho\boldsymbol{g}$$

$$-\,2\rho\boldsymbol{\Omega}\times\boldsymbol{u} + \boldsymbol{\nabla}\cdot\mathsf{S}$$

rate of strain tensor for Newtonian viscosity:

$$S_{ij} = 2\nu\rho\left[e_{ij} - \frac{1}{3}\delta_{ij}\,\boldsymbol{\nabla}\cdot\boldsymbol{u}\right],$$

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right).$$

Density Variations

Continuity equation:

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

Equation of state:

$$\frac{1}{\rho}\partial \rho = -\alpha \partial T + \beta \partial p + \delta \partial \chi$$

with thermodynamical properties

thermal expansivity $$\alpha = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p$$

compressibility $$\beta = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial p}\right)_T$$

Density Variations

Energy equation:

$$\rho T\left(\frac{\partial s}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla} s\right)=\boldsymbol{\nabla}\cdot\left(k\boldsymbol{\nabla} T\right)+\Phi_\nu+\lambda\left(\boldsymbol{\nabla}\times\boldsymbol{B}\right)^2+\epsilon$$

with viscous heating

$$\Phi_\nu=2\rho\left[e_{ij}e_{ji}-\frac{1}{3}\left(\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)^2\right]$$

If compositional changes are considered another equivalent respective evolution equation is required.

Dynamo equation

Non-relativistic Maxwell equations provide

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B} - \lambda \boldsymbol{\nabla} \times \boldsymbol{B})$$

And if the magnetic diffusivity λ is homogeneous

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla} \times (\boldsymbol{u} \times \boldsymbol{B}) + \lambda \boldsymbol{\Delta} \boldsymbol{B}$$

Disturbance Around a Background State

Small disturbance (prime) around a reference state (tilde)

$$\epsilon \sim \frac{T'}{\tilde{T}} \sim \frac{p'}{\tilde{p}} \sim \frac{\rho'}{\tilde{\rho}} \sim \ldots \ll 1$$

The reference state is hydrostatic, adiabatic, and non magnetic:

$$\nabla \tilde{p} = \tilde{\rho} \tilde{\boldsymbol{g}}$$

$$\frac{\nabla \tilde{T}}{\tilde{T}} = \frac{1}{\tilde{T}} \left(\frac{\partial T}{\partial p} \right)_s \nabla p = \frac{\alpha}{c_p} \tilde{\boldsymbol{g}}$$

$$\frac{\nabla \tilde{\rho}}{\tilde{\rho}} = \frac{1}{\tilde{\rho}} \left(\frac{\partial \rho}{\partial p} \right)_s \nabla p = \beta \tilde{\rho} \tilde{\boldsymbol{g}}$$

It can be characterized by the two numbers $Di = \dfrac{\alpha d}{c_p} \tilde{g} \quad Co = d \beta \tilde{\rho} \tilde{g}$

Simplified Continuity Equation

Continuity equation:

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

Plug in $\rho = \tilde{\rho} + \rho'$ leads to:

$$\frac{\partial \tilde{\rho}}{\partial t} + \frac{\partial \rho'}{\partial t} = -\boldsymbol{\nabla} \cdot (\tilde{\rho} \boldsymbol{u}) - \boldsymbol{\nabla} \cdot (\rho' \boldsymbol{u})$$

Estimate of ratio:

$$\frac{[\partial \rho / \partial t]}{[\boldsymbol{\nabla} \cdot \rho \boldsymbol{u}]} \approx \frac{\rho'}{\tilde{\rho}} \approx \epsilon$$

First order equation thus reads (used for anelastic approximation):

$$\boldsymbol{\nabla} \cdot (\tilde{\rho} \boldsymbol{u}) = 0$$

Boussinesq Approximation

Appropriate for terrestrial planets where Di and Co are small.
For example for Earth Di~Co~0.2.
Formal limit Di ➔0, Co ➔0

Further simplification of Boussinesq approximation for Co➔0

$$\frac{1}{\tilde{\rho}}\boldsymbol{\nabla}\cdot\tilde{\rho}\boldsymbol{u}=\frac{\boldsymbol{u}}{\tilde{\rho}}\cdot\nabla\tilde{\rho}+\nabla\cdot\boldsymbol{u}\approx\nabla\cdot\boldsymbol{u}=0$$

Vanishing viscous and Ohmic heating.

Boussinesq Navier-Stokes Equations

$$\tilde{\rho}\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\ \boldsymbol{u}\right) = -\boldsymbol{\nabla} p' - 2\rho\boldsymbol{\Omega}\times\boldsymbol{u} + \alpha g_o T' \frac{\boldsymbol{r}}{r_o}$$
$$+ \frac{1}{\mu_0}(\boldsymbol{\nabla}\times\boldsymbol{B})\times\boldsymbol{B} + \tilde{\rho}\nu\Delta\boldsymbol{u}$$

Rescaling to dimensionless form:

$$r \to r\ d, t \to (d^2/\nu)\ t, T \to \Delta T\ T, B \to (\mu\lambda\tilde{\rho}\Omega)^{1/2}\ B$$

$$\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\ \boldsymbol{u}\right) = -\boldsymbol{\nabla} p' - \frac{2}{E}\boldsymbol{e_z}\times\boldsymbol{u} + \frac{Ra}{Pr}T'\frac{\boldsymbol{r}}{r_o}$$
$$+ \frac{1}{EPm}(\boldsymbol{\nabla}\times\boldsymbol{B})\times\boldsymbol{B} + \Delta\boldsymbol{u}$$

Remaining Equations

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

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

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

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

From Physical Properties to Dimensionless Numbers

From 11 properties to **five dimensionless control parameter**!

1) Ekman number $$E = \frac{\nu}{\Omega d^2},$$

2) Rayleigh number $$Ra = \frac{\alpha_o g_o T_o d^3 \Delta s}{c_p \kappa_o \nu_o}$$

3) Prandtl number $$Pr = \frac{\nu_o}{\kappa_o},$$

4) magnetic Prandtl $$Pm = \frac{\nu_o}{\lambda_i}$$

5) aspect ratio $$\eta = \frac{r_i}{r_o}$$

Poloidal/Toroidal Decomposition

From 9 equations for 8 unknown to **6 unknowns and equations!**

Fulfill continuity equations by using

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

$$\boldsymbol{B} = \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times g\, \boldsymbol{e_r}) + \boldsymbol{\nabla} \times h\, \boldsymbol{e_r}$$

W and g are called the poloidal potentials.

Z and h are called the toroidal potentials.

NOTE:

$$\boldsymbol{u} = -\Delta_H\, \boldsymbol{e_r}\, W + \boldsymbol{\nabla}_H \frac{\partial}{\partial r}\, W + \boldsymbol{\nabla}_H \times \boldsymbol{e}_r\, Z$$

Radial component is purely poloidal.

Horizontal poloidal component depends on radial derivative.

Poloidal/Toroidal Decomposition

From 9 equations for 8 unknown to **6 unknowns and equations!**

To solve for the 6 unknowns W g, Z, h, T' and p'
we use
poloidal and toroidal Navier-Stokes equation,
poloidal and toroidal dynamo equation,
heat equation,
'pressure' equation derived from Navier-Stokes equation.

NOTE: other people get rid of pressure by taking an addition curl.

Poloidal/Toroidal Equations

From vectorial to toroidal and poloidal equations via operators

$$\boldsymbol{e_r}\cdot\tilde{\rho}\boldsymbol{u} = -\Delta_H\, W,$$

$$\boldsymbol{e_r}\cdot(\boldsymbol{\nabla}\times\boldsymbol{u}) = -\Delta_H\, Z,$$

with

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

so that

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

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

'pressure' equation:

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

Flow Boundary Conditions

Radial flow vanishes when $W = 0$

a) rigid boundaries: horizontal flow vanishes when

$$\frac{\partial W}{\partial r} = 0 \;\; \text{and} \;\; Z = 0$$

b) stress-free: radial derivative of U/r vanishes when

$$\left(\frac{\partial^2}{\partial r^2} - \frac{2}{r}\frac{\partial}{\partial r} \right) W = 0 \quad \text{and} \quad \left(\frac{\partial}{\partial r} - \frac{2}{r} \right) Z = 0$$

Temperature and Magnetic Conditions

1) Temperature:

a) fixed temperature $T = \text{const.}$

b) fixed flux $\frac{\partial}{\partial r} T = \text{const.}$

c) patterns in terms of spherical hamonics

2) Matching condition to potential magnetic field $\boldsymbol{B}^I = -\boldsymbol{\nabla} V$:

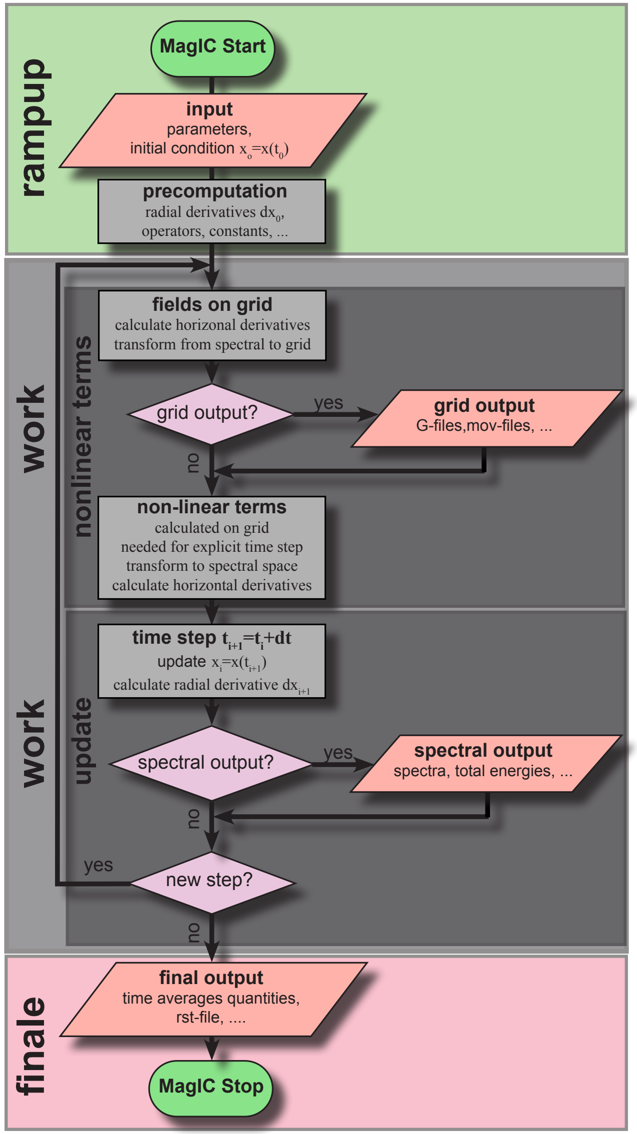
$$h = 0$$

$$\frac{\nabla_H^2}{r^2} g = \frac{\partial}{\partial r} V^I$$

$$\nabla_H \frac{\partial}{\partial r} g = -\nabla_H V^I$$

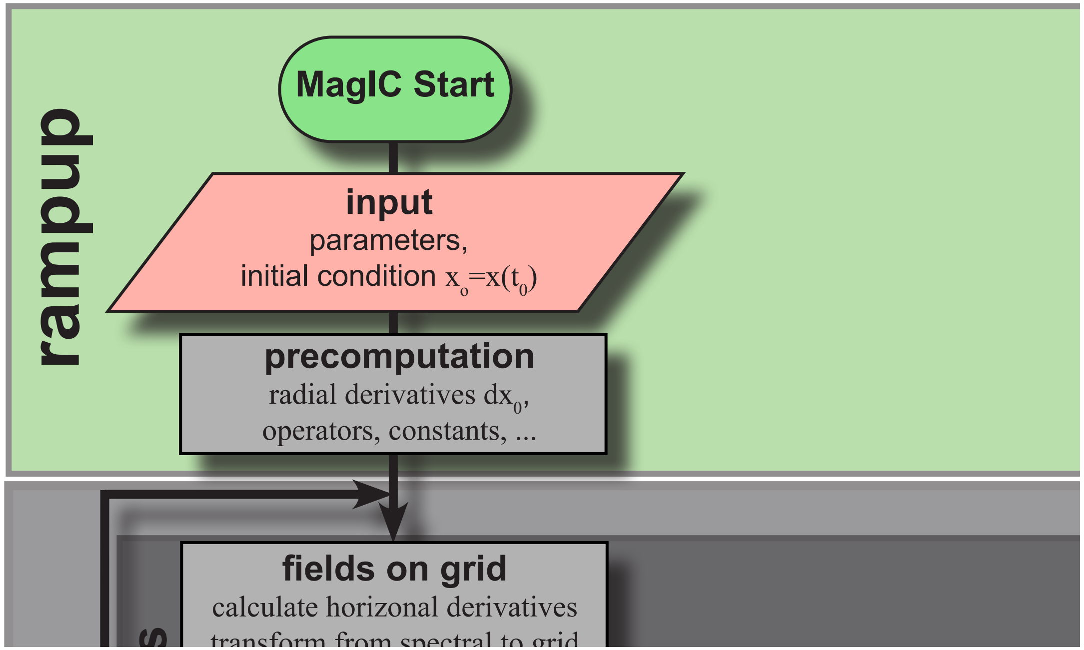
The latter two conditions are combined to eliminate V^I

MagIC Structure



MagIC flow chart
DIN 66001

MagIC Rampup

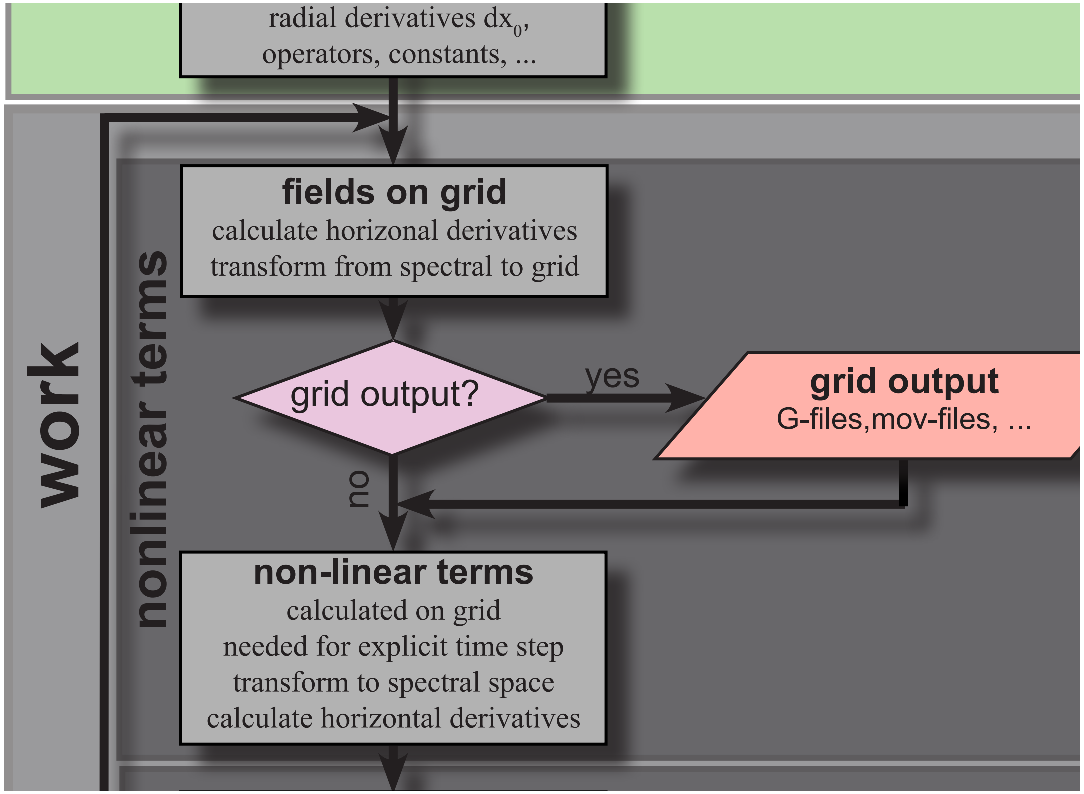


Parameter input via **namelists**

Initial condition via **rst-file** containing all fields from previous run plus respective explicit

Alternatively the fields can be initialized with an **analytical guess or noise**.

MagIC Nonlinear Terms



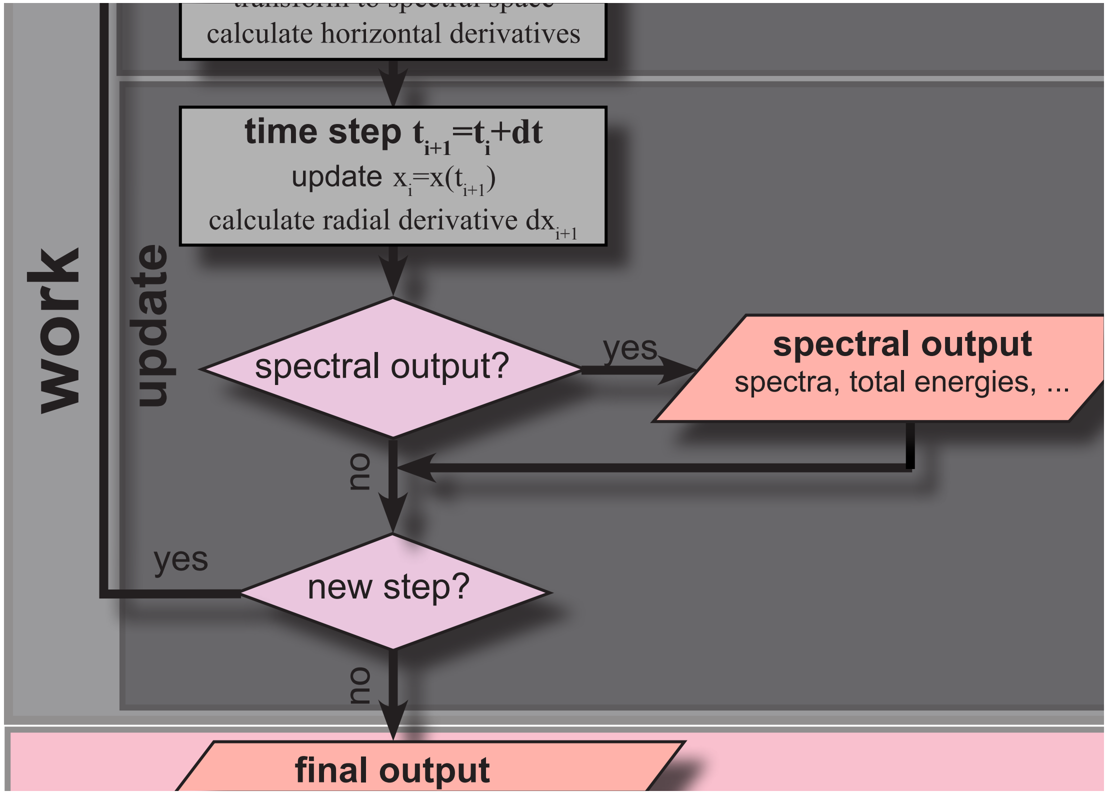
Initial fields in (r,l,m) space:
$x(r,l,m)$
horizontal derivatives calculated

transform to grid via Gauss-Legendre and Fourier transforms:
$x(r,\theta,\Phi)$

Output of any desired quantities on grid $x(r,\theta,\Phi)$ in G- or mov-files.

Nonlinear term calculated
$NL(r,\theta,\Phi)=x_1(r,\theta,\Phi)x_2(r,\theta,\Phi)$
transformed back:
$NL(r,l,m)$
Additional horiz. derivatives

MagIC Time Step

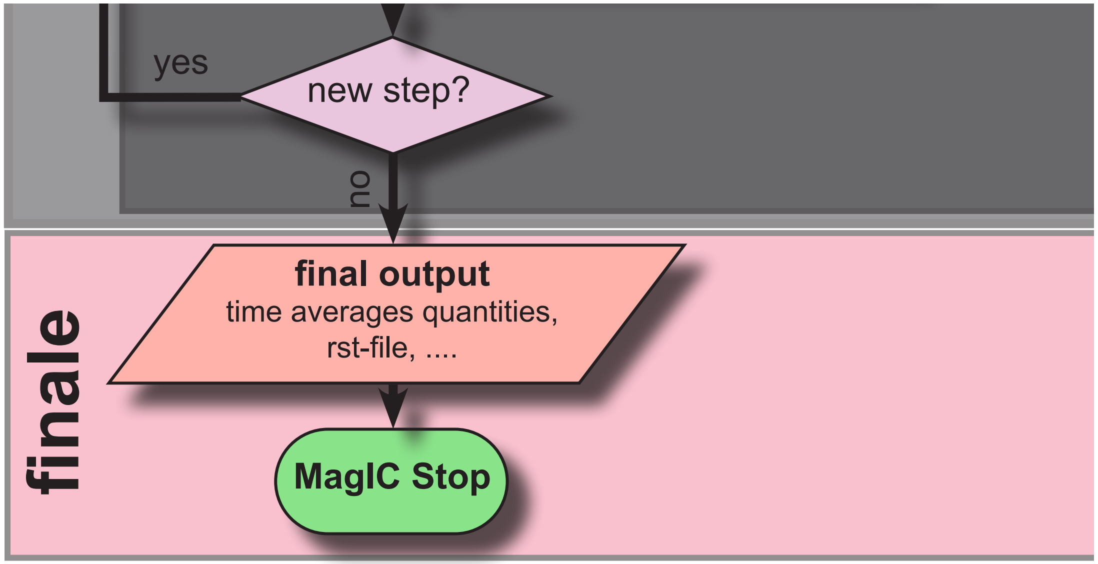


Mixed implicit/explicit time step in (r,l,m) space.

Updated fields given in (n,l,m) space: x(n,l,m)

Radial derivative calculated in (n,l,m) space.
Cheb-transform to (r,l,m)

MagIC Finale

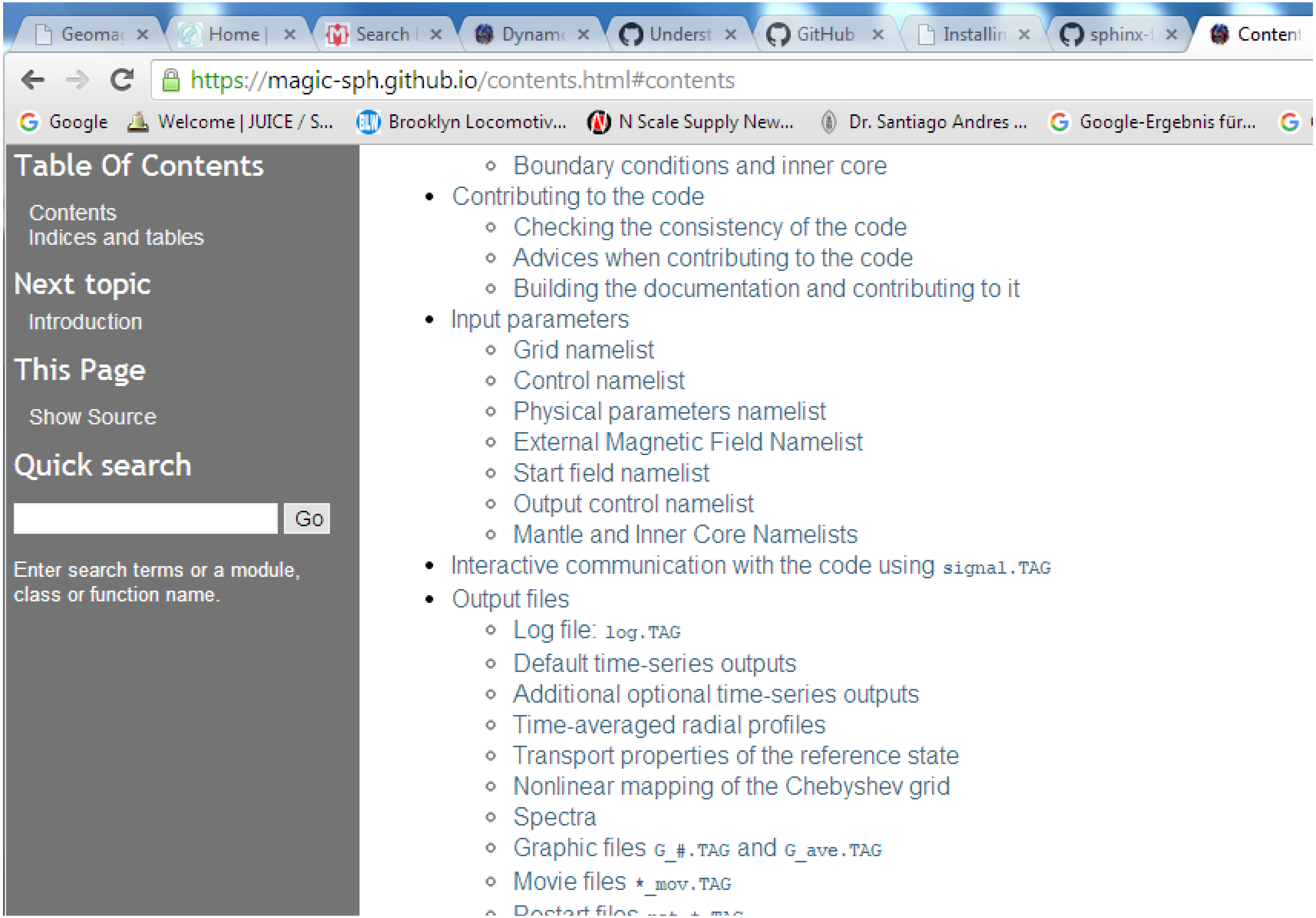


Volume averages calculat-ed in spectral space for output: e_kin, e_mag..

rst-file stored in (r,l,m) space

MagIC finishes with storing some diagnostics in the **log-file**.

Input


Geoma
Home |
Search
Dynam
Underst
GitHub
Installin
sphinx-
Content
https://magic-sph.github.io/contents.html#contents
Google
Welcome | JUICE / S...
Brooklyn Locomotiv...
N Scale Supply New...
Dr. Santiago Andres ...
Google-Ergebnis für...
Table Of Contents
Contents
Indices and tables
Next topic
Introduction
This Page
Show Source
Quick search
Go
Enter search terms or a module, class or function name.
Boundary conditions and inner core
Contributing to the code
Checking the consistency of the code
Advices when contributing to the code
Building the documentation and contributing to it
Input parameters
Grid namelist
Control namelist
Physical parameters namelist
External Magnetic Field Namelist
Start field namelist
Output control namelist
Mantle and Inner Core Namelists
Interactive communication with the code using signal.TAG
Output files
Log file: log.TAG
Default time-series outputs
Additional optional time-series outputs
Time-averaged radial profiles
Transport properties of the reference state
Nonlinear mapping of the Chebyshev grid
Spectra
Graphic files G_#.TAG and G_ave.TAG
Movie files *_mov.TAG

Grid-Namelists

input.nml

```
&grid
 n_r_max        =33,
 n_cheb_max    =31,
 n_phi_tot       =48,
 n_r_ic_max     =17,
n_cheb_ic_max=15,
 minc             =1,
/
```

manual explanation

- **n_r_max** (default `n_r_max=33`) is an integer which gives the number of grid points in the radial direction in the outer core ($[r_i, r_o]$). It must be of the form `4*n+1`, where `n` is an integer.

> **Note:** The possible values for `n_r_max` are thus: 17, 21, 25, 33, 37, 41, 49, 61, 65, 73? 81, 97, 101, 121, 129, 145, 161, 257, 401, 513, ...

- **n_cheb_max** (default `n_cheb_max=31`) is an integer which is the number of terms in the Chebyshev polynomial expansion to be used in the radial direction - the highest degree of Chebyshev polynomial used being `n_cheb_max-1`. Note that `n_cheb_max <= n_r_max`.

> **Note:** Adopting `n_cheb_max=n_r_max-2` is usually a good choice

- **n_phi_tot** (default `n_phi_tot=192`) is an integer which gives the number of longitudinal/azimuthal grid points. It has the following contraints:
  - `n_phi_tot`` must be a multiple of `minc` (see below)
  - `n_phi_tot/minc` must be a multiple of 4
  - `n_phi_tot` must be a multiple of 16

> **Note:** The possible values for `n_phi_max` are thus: 16, 32, 48, 64, 96, 128, 192, 256, 288? 320, 384, 400, 512, 576, 640, 768, 864, 1024, 1280, 1536, 1792, 2048, ...

Control-Namelist

input.nml

```
&control
 mode          =0,
 tag           =”test”,
 n_time_steps=1000,
 courfac       =2.5D0,
 alffac        =1.0D0,
 dtmax         =1.0D-4,
 n_cour_step =5,
 alpha         =0.6D0,
 runHours      =12,
 runMinutes    =00,
/
```

manual explanation

This namelist defines the numerical parameters of the problem plus the variables that control and organize the run.

- **mode** (default `mode=0`) is an integer which controls the type of calculation performed.

mode=0	Self-consistent dynamo
mode=1	Convection
mode=2	Kinematic dynamo
mode=3	Magnetic decay modes
mode=4	Magneto convection
mode=5	Linear onset of convection
mode=6	Self-consistent dynamo, but with no Lorentz force
mode=7	Super-rotating inner core or mantle, no convection and no magnetic field
mode=8	Super-rotating inner core or mantle, no convection
mode=9	Super-rotating inner core or mantle, no convection and no Lorentz force
mode=10	Super-rotating inner core or mantle, no convection, no magnetic field, no Lorentz force and no advection

- **tag** (default `tag="default"`) is a character string, used as an extension for all output files.
- **n_time_steps** (default `n_time_steps=100`) is an integer, the number of time steps to be performed.

Physical Parameter-Namelist

input.nml

```
&phys_param
 ra        =1.1D5,
 ek        =1.0D-3,
 pr        =1.0D0,
 prmag     =5.0D0,
 radratio  =0.35D0,
 ktops     =1,
 kbots     =1,
 ktopv     =2,
 kbotv     =2,
 kbotb     =3
/
```

manual explanation

- **ktops** (default `ktops=1`) is an integer to specify the outer boundary entropy (or temperature) boundary condition:

`ktops=1`	Fixed entropy at outer boundary: $s(r_o) = s_{top}$
`ktops=2`	Fixed entropy flux at outer boundary: $\partial s(r_o)/\partial r = s_{top}$

- **ktopv** (default `ktopv=2`) is an integer, which corresponds to the mechanical boundary condition for $r = r_o$.

`ktopv=1`	Stress-free outer boundary for $r = r_o$:
`ktopv=2`	Rigid outer boundary for $r = r_o$:

- **ktopb** (default `ktopb=1`) is an integer, which corresponds to the magnetic boundary condition for $r = r_o$.

`ktopb=1`	Insulating outer boundary:
`ktopb=3`	Finitely conducting mantle
`ktopb=4`	Pseudo-vacuum outer boundary:

Start Field Namelist

input.nml

```
&start_field
 l_start_file=.FALSE.,
 start_file   ="NONE",
 init_b1      =3,
 amp_b1      =5,
 init_s1      =0404,
 amp_s1      =0.1,
/
```

manual explanation

Reading an input file of start fields

- **l_start_file** (default `l_start_file=.false.`) is a logical that controls whether the code should to read a file named `start_file` or not.
- **start_file** (default `start_file="no_start_file"`) is a character string. This is the name of the restart file.

Initialisation of magnetic field

- **init_b1** (default `init_b1=0`) is an integer that controls the initial magnetic field. The following values are possible:
  - `init_b1=3`: $(\ell=1, m=0)$ poloidal field whose field strength is `amp_b1` at $r=r_i$. The radial dependence is chosen such that the current density j is independent of r:, i.e. $\partial j/\partial r = 0$. $(\ell=2, m=0)$ toroidal field with maximum strength `amp_b1`.

The log-file

Provides all important information about the run

1) MagIC version

2) all parameters and other inputs including default ones

3) information on parallelization, run time etc.

4) log of important events: important output files,
changing time step, ...

5) important time averaged quantities, measures .....

Output-Namelist

input.nml

```
&output_control
 n_log_step =1,
 n_graphs    =1,
 n_rsts        =1,
 n_stores     =0,
 runid       ="Benchmark 2",
 l_movie     =.FALSE.,
 l_RMS       =.FALSE.,
/
```

manual explanation

- **n_log_step** (default `n_log_step=50`) is an integer. This is the number of timesteps between two log outputs.

  Warning: Be careful: when using too small `n_log_step`, the disk access will dramatically increases, thus decreasing the code performance.

- **n_logs** (default `n_logs=0`) is an integer. This is the number of log-information sets to be written.
- **t_log** (default `t_log=-1.0 -1.0 ...`) is real array, which contains the times when log outputs are requested.
- **dt_log** (default `dt_log=0.0`) is a real, which defines the time interval between log outputs.
- **t_log_start** (default `t_log_start=0.0`) is a real, which defines the time to start writing log outputs.
- **t_log_stop** (default `t_log_stop=0.0`) is a real, which defines the time to stop writing log outputs.

Time series files

```
e_kin.TAG
```

This file contains the kinetic energy of the outer core, defined by

$$E_k = \frac{1}{2}\int_V \bar{\rho} u^2 \, \mathrm{d}V = E_{pol} + E_{tor}$$

No. of column	Contents
1	time
2	poloidal energy
3	toroidal energy
4	axisymmetric poloidal energy
5	axisymmetric toroidal energy
6	equatorial symmetric poloidal energy
7	equatorial symmetric toroidal energy
8	equatorial symmetric and axisymmetric poloidal energy
9	equatorial symmetric and axisymmetric toroidal energy

This file can be read using `MagicTs` with the following options:

Other Output Files

Graphic files `G_#.TAG` and `G_ave.TAG`

All fields in single precission for graphic visualization (except pressure).

Movie files `*_mov.TAG`

Pre-defined (derived) fields in pre-defined cuts (or full 3d) for several time steps. Examples: z-vorticity, dynamo action, ...

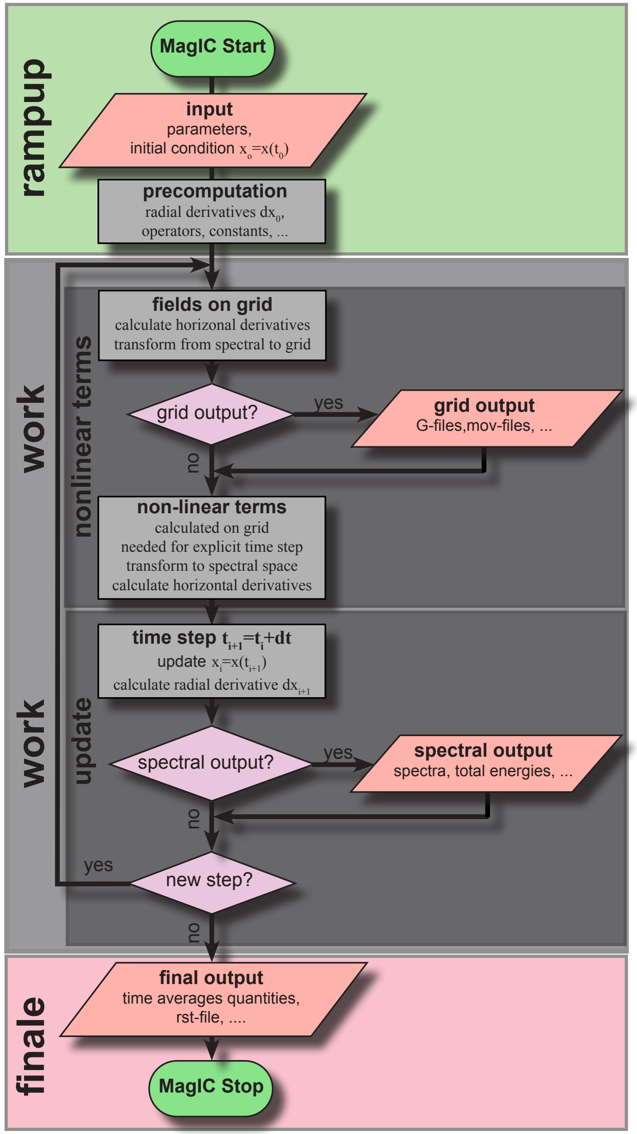
Restart files `rst_*.TAG`

All fields plus explicit time step vector in full precision.
Used for continuing an integration of as safety backup.

Conclusion

- MagIC is the most efficient code out there.
- Its publicly available.
- All the neccessary (and more) documentation in online.
- Have fun doing the first MagIC runs!

MagIC Structure



MagIC flow chart
DIN 66001