Politecnico di Torino

European Research Council
Established by the European Commission

Master's Degree in Aerospace Engineering

Double Degree with ISAE-SUPAERO

Numerical Modeling of the Tayler–Spruit Dynamo Instability

Insights into the Radiative Layer of Stellar Reality

Supervisors:
Domenic **D'Ambrosio**
Florence **Marcotte**
Lionel **Bigot**
Gabriele **Nastro**

Candidate:
Giorgio **Appignanesi**
315419

March 2026

Abstract

In stellar interiors, despite various sources of shear such as core contraction and envelope expansion, stellar asteroseismology reveals nearly uniform rotation in radiative layers. These observations indicate that an efficient internal angular momentum transport mechanism must act from the inner regions to the outer layers to maintain this uniformity.

The Tayler instability could be the key mechanism, where differential rotation amplifies a toroidal field that becomes unstable to a non-axisymmetric $m = 1$ kink mode regenerating poloidal components, generating a dynamo loop.

This work was carried out at INRIA in the framework of a end-of-study internship using the `Xshells` code, building on results from a reference study performed with the `PARODY` code. As a preliminary step, the results of the reference paper Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1] were reproduced in order to validate the numerical setup.

The main objective of this stage was then to modify the boundary conditions from no slip to free slip and introduce a volumetric forcing term in order to investigate how the system saturates. These conditions are considered more representative of stellar interiors than the originally assumed rigid boundaries, since the forcing is applied throughout the volume rather than only at the boundaries.

Various technical difficulties were encountered during the stage, which prevented the completion of the initial objectives.

After presenting the general context of the study and the theoretical framework in chapter 1, the focus will shift to the numerical setup and the code used in chapter 2. Then, the study methodology and the reference article used as a starting point are presented in chapter 3, and finally the obtained results are discussed in chapter 4, concluding the document.

Table of Contents

Chapter 1

Theoretical Framework

1.1 Context

During the course of their life, stars interior undergoes two opposite changes: the core contracts while the envelope expands. In addition, magnetic braking via stellar winds further removes angular momentum from the convective envelope Aerts et al. 2019 [2]. Conservation of angular momentum dictates that the core should spin up and the envelope should spin down producing strong angular velocity gradients, a situation described hereafter as differential rotation.

However, asteroseismological measurements reveal nearly flat internal rotation profiles in stars at various stages of their evolution Basu 2016 [3]; Howe 2009 [4]. The solar convective envelope rotates differentially with latitude, but this differential rotation transitions abruptly into almost uniform rotation particularly in the radiative zone below $0.7\,R_\odot$ Spruit 1999 [5]. These observations indicate that an efficient internal angular momentum (AM) transport mechanism must efficiently transport angular momentum outward to maintain this uniformity.

Several processes have been proposed to explain AM transport in stellar interiors (summarized by Aerts et al. 2019 [2]). These include hydrodynamical processes such as meridional circulation and shear-driven turbulence, internal gravity waves, fossil magnetic fields, and magnetohydrodynamic instabilities. Among these, hydrodynamic processes alone are insufficient to explain the nearly solid-body rotation observed in evolved stars Eggenberger 2024 [6]. Internal gravity waves may contribute, especially in early evolutionary phases, but their effectiveness and propagation depend strongly on internal stratification and geometry. Fossil magnetic fields are a possible source of large-scale torques, but the conditions for their survival and stability over Gyr timescales remain uncertain Braithwaite and Spruit 2017 [7]. Instead, magnetic instabilities driven by differential rotation and shear have emerged as leading candidates.

In particular, Tayler 1973 [8] proposed a mechanisms based on a kink instability, better described later in subsection 1.4.1. Spruit 2002 [9] proposed a dynamo mechanism combining the Ω-effect and the Tayler instability. In Spruit's model, differential rotation (Ω-effect) generates a toroidal magnetic field from a weak poloidal seed field. Once this toroidal field becomes sufficiently strong, it is destabilized by the Tayler instability, which generates perturbations that reconstruct poloidal field components, thereby closing the dynamo loop. Unlike the convective Ω–α dynamos, described in section 1.3 and operating in convective envelopes, this mechanism does not rely on turbulent helical flows. The resulting magnetic field is predominantly toroidal (azimuthal), and the associated Maxwell stresses efficiently transport angular momentum. Fuller et al. 2019 [10] predicted AM transport using a different saturation mechanism compared to Spruit 2002 [9] and implemented this prescription into stellar evolutionary models to predict internal rotation rates, finding generally good agreement with asteroseismic measurements. However, whether such a dynamo can be truly self-sustaining and how the saturation mechanism operates remains debated.

For instance, Braithwaite 2007 [11] showed that under certain assumptions, dynamo-like behavior can emerge in simulations of rotating, stably stratified shells. Nevertheless, Zahn et al. 2007 [12] challenged this picture, arguing that the Tayler instability alone cannot regenerate the axisymmetric ($m = 0$) poloidal field required to sustain the Ω–effect. An important issue is that to linear order the Tayler instability grows fastest in the non-axisymmetric $m = 1$ mode. Therefore, the radial field generated by the instability is non-axisymmetric, and winding of this field by differential rotation produces no net increase in the axisymmetric component of B_ϕ. In their simulations, non-axisymmetric perturbations grew and saturated, but failed to maintain a global poloidal field, and angular momentum transport remained inefficient.

More recently, Petitdemange, Marcotte, and Gissinger 2023 [13]; Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1] used global 3D MHD simulations to demonstrate that a Tayler–Spruit-like dynamo can operate under certain conditions. They showed that sustained dynamo action is possible in a *supercritical* regime, where the Reynolds number must exceed a critical value $Re_c \propto \sqrt{Ra}/Pr$. While this condition is naturally met in real stars, it is computationally challenging to achieve in simulations.

Before proceeding, we briefly define the key dimensionless numbers introduced above and used throughout this work:

- **Reynolds number (Re):** The ratio of inertial forces to viscous forces, defined here as $Re = \frac{L^2 \Delta\Omega}{\nu}$. In the context of differential rotation, it is convenient to define the characteristic velocity scale using the difference in angular velocity between the outer and inner boundaries, denoted $\Delta\Omega$, and a characteristic

length scale L, such that the characteristic velocity is $U \sim L\,\Delta\Omega$.

- **Magnetic Reynolds number (Re_m):** The ratio of magnetic field advection to diffusion, defined as $Re_m = \frac{L^2\,\Delta\Omega}{\eta}$.
- **Rayleigh number (Ra):** A measure of buoyancy-driven flow strength relative to diffusion, given by $Ra = \frac{\alpha\, g\, \Delta T\, L^3}{\nu\, \kappa}$. Here g is gravitational acceleration, α the thermal expansion coefficient, ΔT the characteristic temperature difference, and κ the thermal diffusivity. In stably stratified flows, Ra characterizes the strength of stratification, with large Ra corresponding to strong stratification.
- **Prandtl number (Pr):** The ratio of momentum diffusivity to thermal diffusivity, $Pr = \frac{\nu}{\kappa}$. The Prandtl number influences how thermal stratification compares to momentum transport in the flow.
- **Magnetic Prandtl number (Pm):** The ratio of momentum diffusivity to magnetic diffusivity, $Pm = \frac{\nu}{\eta}$. The magnetic Prandtl number influences the relative scales of viscous and magnetic diffusion, affecting the dynamo behavior.
- **Ekman number (E):** The ratio of viscous forces to Coriolis forces, defined as $E = \frac{\nu}{\Omega\, L^2}$. This number quantifies the influence of rotation on the flow where low E corresponds to a strongly rotation-dominated regime.
- **Rossby number (Ro):** A measure of the differential rotation, defined as $Ro = \frac{\Delta\Omega}{\Omega}$. This compares the differential rotation rate to the overall rotation rate and, together with E, is fundamental in determining the shear and rotational regime of the system.

In Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1] the excitation of the Tayler-Spruit dynamo was non-linearly achieved through the prior linear excitation of a weaker, shear-driven dynamo instability.

The difference matters because of the numerical difficulty in achieving a parameter regime where the Ω-effect is sufficiently vigorous to meet the required toroidal field amplitude for the Tayler instability. This numerical difficulty may explain why some previous attempts to produce TS dynamos were unsuccessful.

By initializing simulations from previously saturated dynamo states, they could sustain dynamo action even at reduced shear or higher stratification and, through gradual parameter continuation, they reached regimes closer to astrophysical conditions (including $N/\Omega \sim 100$). Their simulations confirm that once a dynamo is established, it maintains persistent non-axisymmetric perturbations, with the Tayler mode likely responsible for regenerating the poloidal field.

The simulations also show strong Maxwell stresses transporting AM, consistent with the dimensional scalings predicted by Spruit 2002 [9].

However, contributions from the azimuthal magnetorotational instability (AMRI) cannot be entirely excluded and more numerical investigations are needed to fully understand the interplay between these instabilities.

The goal of this work is first to reproduce the results of Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1] using an alternative numerical framework based on `Xshells`, as a validation step. Building on this, the longer-term objective is to move beyond the reference setup by replacing rigid (no-slip) boundaries with more realistic conditions for stellar interiors (stress-free boundaries and a volumetric forcing of the shear), in order to investigate how the dynamo saturates when the shear is injected throughout the volume.

1.2 Magnetohydrodynamics Equations

To investigate this phenomenon, we employ the framework of magnetohydrodynamics (MHD). MHD describes the dynamics of electrically conducting fluids, such as stellar interiors, by combining the Navier–Stokes equations with Maxwell's equations. Here, we summarize the governing equations for an incompressible, stably stratified fluid in the Boussinesq approximation.

1.2.1 Induction Equation

The evolution of the magnetic field in a moving conductor is governed by Faraday's law of induction coupled with Ohm's law for a moving medium. Starting from Faraday's law:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}\,,$$

where $\mathbf{B}$ is the magnetic field and $\mathbf{E}$ the electric field.
Let l and τ be characteristic length and time scales of the system. Then

$$\left|\frac{\partial \mathbf{B}}{\partial t}\right| \sim B/\tau \quad \text{and} \quad |\nabla \times \mathbf{E}| \sim E/l$$

dimensionally $B/\tau \sim E/l$.

This yields an estimate $E \sim \frac{l}{\tau} B$. If we define a characteristic velocity $v = l/\tau$, we obtain

$$E \sim v\,B\,,$$

i.e. the electric field induced by a changing magnetic field is of order vB. This v can be interpreted as the characteristic velocity associated with the field evolution.

Now consider Ohm's law in a moving conductor, where σ is the electrical conductivity, assumed here to be constant everywhere in the fluid:

$$\mathbf{J} = \sigma\Big(\mathbf{E} + \mathbf{u} \times \mathbf{B}\Big)\,,$$

where $\mathbf{J}$ is the current density and $\mathbf{u}$ is the fluid velocity. Rearranging, the electric field can be written as $\mathbf{E} = -\,\mathbf{u} \times \mathbf{B} + \mathbf{J}/\sigma$. Substituting this into Faraday's law gives:

$$\frac{\partial \mathbf{B}}{\partial t} = -\,\nabla \times \Big(-\,\mathbf{u} \times \mathbf{B} + \frac{1}{\sigma}\mathbf{J}\Big) = \nabla \times (\mathbf{u} \times \mathbf{B}) \;-\; \frac{1}{\sigma}\,\nabla \times \mathbf{J}\,.$$

Next, we invoke Ampère's law:

$$\nabla \times \mathbf{B} = \mu_0\,\mathbf{J} + \varepsilon_0\mu_0\,\frac{\partial \mathbf{E}}{\partial t}\,,$$

where μ_0 is the magnetic permeability and ε_0 the electric permittivity of free space. Remembering the dimensional analysis of Faraday's law, the ratio of the displacement current term to the physical current term is of order

$$\frac{\varepsilon_0\mu_0\,|\frac{\partial \mathbf{E}}{\partial t}|}{|\nabla \times \mathbf{B}|} \sim \frac{1}{c^2}\,\frac{E/\tau}{B/l} \sim \frac{1}{c^2}\,\frac{vBl}{B\tau} = \left(\frac{v}{c}\right)^2\,,$$

where $c = 1/\sqrt{\varepsilon_0\mu_0}$ is the speed of light.

In stellar interiors, $v \ll c$, so $(v/c)^2 \ll 1$. Thus the displacement current is negligible on MHD timescales, and we can use the quasi-static approximation $\nabla \times \mathbf{B} \approx \mu_0\,\mathbf{J}$.

Using this approximation for $\mathbf{J}$, the induction equation becomes:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) \;-\; \frac{1}{\sigma\mu_0}\,\nabla \times (\nabla \times \mathbf{B}) \;\; = \nabla \times (\mathbf{u} \times \mathbf{B}) \;+\; \frac{1}{\sigma\mu_0}\Big[\nabla(\nabla \cdot \mathbf{B}) - \Delta\mathbf{B}\Big]\,,$$

where in the second line we have used the vector identity $\nabla\times(\nabla\times\mathbf{a}) = \nabla(\nabla\cdot\mathbf{a}) - \Delta\mathbf{a}$. By Gauss's law for magnetism, $\nabla \cdot \mathbf{B} = 0$, so this simplifies to

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \eta\,\Delta\mathbf{B}\,, \tag{1.1}$$

where we have introduced the magnetic diffusivity $\eta \equiv \frac{1}{\sigma\mu_0}$.

Finally, assuming the flow is incompressible ($\nabla \cdot \mathbf{u} = 0$), the term $\nabla \times (\mathbf{u} \times \mathbf{B})$ can be expanded using vector identities. Noting that $\nabla \cdot \mathbf{B} = 0$ as well, we have:

$$\nabla \times (\mathbf{u} \times \mathbf{B}) = (\mathbf{B} \cdot \nabla)\mathbf{u} \;-\; (\mathbf{u} \cdot \nabla)\mathbf{B}\,.$$

Substituting this into (1.1) and rearranging terms, we obtain the standard MHD induction equation in convective form:

$$\frac{\partial \mathbf{B}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{B} = (\mathbf{B}\cdot\nabla)\mathbf{u} + \eta\,\Delta\mathbf{B}\,, \tag{1.2}$$

which highlights the competition between field-line advection/stretching by the flow (the left-hand side and the first term on the right-hand side) and magnetic diffusion (the second term on the right-hand side). In non-dimensional form, this balance is governed by the magnetic Reynolds number: for $Re_m \gg 1$, the advection and stretching terms dominate (efficient field line freezing), whereas for $Re_m \ll 1$, Ohmic diffusion dominates the evolution of the magnetic field.

1.2.2 MHD Navier–Stokes Equations (Neutral Plasma)

Starting from momentum conservation for a viscous, incompressible fluid:

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = \nu\,\Delta\mathbf{u} - \frac{1}{\rho}\nabla P\,,$$

where $\mathbf{u}$ is the velocity field, ν the kinematic viscosity, ρ the density, and P the pressure. This form captures the balance between inertial transport (left-hand side) and the restoring forces of viscosity and pressure gradients (right-hand side).

In a frame rotating at angular velocity $\mathbf{\Omega}$, we must include the Coriolis acceleration. Transforming to the rotating frame adds a term $-2\,\mathbf{\Omega}\times\mathbf{u}$ to the acceleration. The centrifugal term, being time-independent and acting like a modification of the gravitational potential, can be absorbed into the equilibrium hydrostatic pressure and is omitted from the dynamical equations.

In a magnetized conducting fluid, the Lorentz force $\mathbf{J}\times\mathbf{B}$ must be included. Assuming quasi-neutrality (negligible net charge density), the electric force $\rho_c\mathbf{E}$ is zero, so the total electromagnetic force density is $\mathbf{F} = \mathbf{J}\times\mathbf{B}$. Remembering the result from subsection 1.2.1, $\mathbf{J} = \frac{1}{\mu_0}(\nabla\times\mathbf{B})$, the Lorentz force can be written as $\frac{1}{\mu_0}(\nabla\times\mathbf{B})\times\mathbf{B}$.

Finally, in the Boussinesq approximation, density variations are ignored except in the buoyancy term. For a stably stratified fluid, a small temperature perturbation T induces a density perturbation $\rho' \approx -\rho\alpha T$ (with α the thermal expansion coefficient). The buoyancy force is then $\rho'\mathbf{g} \approx -\rho\alpha T\,\mathbf{g}$, which for an environment with gravitational acceleration g_0 in the radial direction can be written as $\rho\alpha T\,g_0\,\mathbf{r}$.

Combining all the above, the momentum equation in the rotating frame with magnetic and buoyancy forces included is:

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = \nu\,\Delta\mathbf{u} - \frac{1}{\rho}\nabla P^* - 2\,\mathbf{\Omega}\times\mathbf{u} + \frac{1}{\mu_0}(\nabla\times\mathbf{B})\times\mathbf{B} + \alpha g_0\,T\,\mathbf{r}\,, \tag{1.3}$$

where P^* is the modified pressure, including the centrifugal term and any background hydrostatic pressure.

This is the MHD form of the Navier–Stokes equation for an incompressible, rotating, stratified fluid. In Equation 1.3, the terms on the right-hand side represent viscous diffusion, pressure gradient forces, Coriolis force, Lorentz force, and buoyancy force. These terms will be relevant for the numerical implementation of the equations.

It is worth noting that the relative importance of rotation in the momentum equation is characterized by the Ekman number E and the Rayleigh number Ra (defined in section 1.1). Slow rotators (high E, high Ra) tend to be dominated by buoyancy forces, often yielding flows and fields aligned with gravity (spherical symmetry in structures). Fast rotators (low E, low Ra), on the other hand, enter the *magnetostrophic* regime where Coriolis and Lorentz forces balance, and the flow/field structures become more cylindrical.

1.3 Dynamo Theory

Before analyzing the stability of magnetized flows (section 1.4), it is essential to understand the concept of a dynamo. In MHD, a *dynamo* refers to the process by which the motion of an electrically conducting fluid generates and maintains a magnetic field. This mechanism underlies the persistent magnetism observed in many astrophysical systems (planets, stars, galaxies) and results from the nonlinear feedback between fluid motion and magnetic induction. The velocity field amplifies magnetic fields via stretching and advection, while the magnetic field reacts back on the flow through the Lorentz force, establishing a self-sustaining loop once a dynamical equilibrium, referred to as saturation, is reached.

In astrophysical contexts, dynamos are typically described by two key mechanisms: differential rotation and mirror-symmetry-breaking perturbations. The first is the Ω–**effect**, in which large-scale shear winds up poloidal field lines into the azimuthal direction, thereby amplifying the toroidal component of the magnetic field. This process can be seen explicitly from the induction Equation 1.2.

Assuming an axisymmetric flow (rotation only in ϕ) and a purely poloidal seed field ($\mathbf{B}_p = B_r\,\hat{\mathbf{e}}_r + B_z\,\hat{\mathbf{e}}_z$), the toroidal induction is approximately

$$\frac{\partial B_\phi}{\partial t} \approx (\mathbf{B}_p \cdot \nabla) u_\phi \sim r\,(\mathbf{B}_p \cdot \nabla\Omega)\,,$$

showing that differential rotation winds up poloidal field into a toroidal field, resulting in linear amplification of the B_ϕ component.

However, the Ω–effect alone is insufficient to sustain a dynamo: while it generates a toroidal field from a poloidal field, it cannot regenerate the poloidal field. In convective regions, this regenerative role is played by the α–**effect**, which arises

from helical turbulence induced by rotation and stratification. Small-scale, mirror-asymmetric velocity fluctuations produce an average electromotive force $\langle \mathbf{u}' \times \mathbf{B}' \rangle$ proportional to the mean magnetic field, feeding back into the poloidal component. The combined action of the Ω–effect and α–effect closes the dynamo loop, forming the classical α–Ω **dynamo** model widely accepted for the convective envelopes of stars.

In radiative interiors, the situation is drastically different. Stable stratification suppresses convection and thus inhibits the α–effect. Differential rotation alone, therefore, cannot maintain a magnetic field: it can amplify B_ϕ from B_p, but it cannot regenerate B_p from B_ϕ.

Another mechanism must take the place of the α–effect to complete the dynamo loop in these non-convective zones.

1.4 Stability Analysis

In the framework of linear stability theory, this involves analyzing the evolution of small perturbations to an equilibrium.

We consider a background state with velocity $\mathbf{u}$ and investigate the linearized induction equation for a magnetic perturbation. Seeking solutions of the form $\mathbf{B}'(\mathbf{r}, t) = \hat{\mathbf{B}}(\mathbf{r})\, e^{\lambda t}$, where λ is an eigenvalue of the problem associated to Equation 1.2:

$$\frac{\partial \mathbf{B}'}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{B}' = (\mathbf{B}' \cdot \nabla)\mathbf{u} + \eta\, \nabla^2 \mathbf{B}' \,. \tag{1.4}$$

This equation describes how the interplay of shear (through the advection term $(\mathbf{u} \cdot \nabla)\hat{\mathbf{B}}$), field-line stretching (through $(\hat{\mathbf{B}} \cdot \nabla)\mathbf{u}$), and Ohmic dissipation (through $\eta \nabla^2 \hat{\mathbf{B}}$) governs the behavior of small magnetic perturbations. Instability requires a solution with $\Re(\lambda) > 0$, implying exponential growth of the perturbation.

Typically, such instability occurs when the magnetic Reynolds number exceeds a critical threshold Rm_c that depends on the flow and field geometry. In the subsequent nonlinear regime, as the field grows, the Lorentz force in the momentum Equation 1.3 feeds back on the flow, modifying $\mathbf{u}$ and eventually leading to a saturated state.

1.4.1 Tayler Instability and Growth Rates

Originally identified by Tayler 1973 [8], this instability affects any purely azimuthal field B_ϕ in a stably stratified star, growing on an Alfvén timescale regardless of field strength.
Spruit 1999 [5] identified it as the most promising candidate for the AM transport in radiative layers and later in Spruit 2002 [9] he refined this idea and proposed a theoretical **Tayler–Spruit dynamo** mechanism operating as follows:

- *Field amplification (Ω–effect):* Differential rotation shears an initial weak poloidal field into a strong toroidal field.
- *Tayler instability:* Once B_ϕ exceeds a critical threshold, it becomes unstable to $m = 1$ modes, generating non-axisymmetric motions.
- *Field regeneration:* These perturbations regenerate poloidal components, closing the dynamo loop.

In this framework, helical turbulence is replaced by Tayler-instability-driven fluctuations. The resulting toroidal field exerts Maxwell stresses ($B_r B_\phi/\mu_0$) that efficiently redistribute angular momentum across radiative zones. Global simulations by Petitdemange, Marcotte, and Gissinger 2023 [13]; Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1] confirm that a self-sustained Tayler-Spruit-like dynamo can arise in stably stratified fluid when the shear is sufficiently strong. These simulations also reproduce the key features of Spruit's theory, including non-axisymmetric magnetic fields and strong angular momentum transport.

A critical aspect of this dynamo model is the saturation mechanism that limits the field's growth and sets the efficiency of angular momentum transport. In Spruit's formulation, saturation occurs when the turbulence induced by the Tayler instability enhances magnetic diffusivity enough to balance toroidal field amplification. This yields moderate field strengths and allows for a residual shear in the radiative zone. In contrast, Fuller et al. 2019 [10] propose that only the non-axisymmetric perturbations, not the large-scale toroidal field, are dissipated. Saturation is instead reached when the energy input from differential rotation equals the dissipation rate of these fluctuations. As a result, stronger fields and Maxwell stresses develop, more efficiently suppressing differential rotation.

For the Tayler–Spruit dynamo to operate, a weak seed magnetic field must grow under differential rotation. In stably stratified radiative zones, the primary instability is the Tayler kink mode ($m = 1$). In ideal MHD without rotation, this mode grows on an Alfvén timescale, when both rotation and stratification are included, growth is reduced but still occurs. In the relevant stellar regime $\omega_A \ll \Omega \ll N$, the growth rate becomes

$$\sigma \sim \frac{\omega_A^2}{\Omega},$$

where $\omega_A = \frac{B_\phi}{\sqrt{\mu_0 \rho}\, r}$ is the local Alfvén frequency. Although rotation stabilizes short-wavelength modes, it does not fully suppress the Tayler instability Spruit 2002 [9].

These considerations are central to the dynamo mechanism discussed in Section1.3, and are supported by simulations showing that purely toroidal or poloidal fields are generically unstable on Alfvénic timescales Braithwaite and Spruit 2017 [7].

1.4.2 Supercritical vs. Subcritical Transitions

Dynamical instabilities can manifest via either *supercritical* (linear) or *subcritical* (nonlinear) transitions. In a supercritical scenario, as soon as a control parameter exceeds a critical value, infinitesimal perturbations will grow exponentially. By contrast, a subcritical transition requires finite-amplitude perturbations to trigger a new sustained state, often exhibiting hysteresis or sensitivity to initial conditions. As mentioned in Section1.1, recent simulations indicate that Tayler–Spruit dynamos can be excited through a subcritical pathway (Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1]) and this may explain why some earlier studies failed to observe sustained dynamo action.
In these simulations, an initial weak field does not spontaneously grow; however, if a sufficiently large initial perturbation or seed field is provided, the dynamo can be excited. Typically, shear first amplifies the magnetic field (primarily toroidal) until the Tayler instability criterion is met locally. Beyond this point, non-axisymmetric fluctuations generated by the instability sustain a mean-field dynamo, which then evolves into a steady or quasi-steady magnetized state Petitdemange, Marcotte, and Gissinger 2023 [13]; Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1]. This behavior explains why spontaneous growth from an infinitesimally small seed field (as one might expect from the idealized Ω–effect alone) is not observed in many numerical experiments: numerical diffusion and limited resolution can damp out the initial linear growth before reaching the instability threshold. However, once a dynamo is established, it generates strong Maxwell stresses and efficiently transports angular momentum, in line with Spruit 2002 [9]'s predictions.

One quantitative measure for the local fulfillment of the Tayler instability criterion is a "magnetic Rossby number" Rb (not to be confused with the Rossby number Ro of the flow), defined by

$$Rb = \frac{r^2\,\partial_r(\overline{B_\phi}/r)}{2\,\overline{B_\phi}}\,,$$

where the overline denotes an azimuthal average. When $Rb > 0$ in some region of the flow, the Tayler instability condition is met.

1.4.3 Relation to Other Instabilities

In simplified cylindrical configurations, the Tayler instability can be difficult to distinguish from the magnetorotational instability (MRI). The MRI is a linear instability that extracts free energy from differential rotation: when a weak magnetic field threads a differentially rotating fluid, magnetic tension couples neighboring fluid elements at different radii. If the angular momentum profile decreases outward ($\partial(r^2\Omega)/\partial r < 0$), this magnetic coupling destabilizes the flow, allowing fluid

elements to exchange angular momentum and spiral outward or inward. The growth rate of the MRI scales with the strength of the differential rotation and the magnetic field strength, making it most effective in environments with strong radial shear.

However, this strong dependence on differential rotation makes the MRI a less plausible candidate for stellar radiative interiors. As discussed in section 1.1, asteroseismic observations reveal nearly flat rotation profiles in stellar cores, precisely the opposite of the steep gradients required for efficient MRI operation. In such weakly sheared environments, the free energy available to drive the MRI is severely limited, and the instability either fails to develop or grows too slowly to be dynamically significant.

By contrast, the Tayler instability is driven by the kink-mode deformation of strong toroidal magnetic fields and does not fundamentally require differential rotation to operate: it can destabilize a toroidal field even in a uniformly rotating system, provided the field strength exceeds a critical threshold.

While differential rotation plays a role in amplifying the toroidal field via the Ω-effect, once this field is established, the Tayler instability can generate non-axisymmetric perturbations independently of the shear profile. This makes the Tayler mechanism more compatible with the observed near-uniform rotation in stellar interiors.

In the azimuthal variant (AMRI), the MRI acts primarily on azimuthal magnetic fields and can overlap with Tayler modes in parameter space, making them difficult to separate in local or cylindrical models.

However, in global spherical simulations under conditions of strong stratification and strong magnetic fields representative of stellar radiative zones, the observed phenomenology is more consistent with Tayler-driven dynamics Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1]. The characteristic $m = 1$ mode structure, the energy partition between magnetic and kinetic components, and the spatial localization of fluctuations all point to the Tayler mechanism as the dominant source of non-axisymmetric perturbations. Furthermore, once the dynamo saturates, the system enters a magnetostrophic regime where Coriolis and Lorentz forces dominate (with relatively weak inertial contributions). This state is characterized by a ratio of magnetic to kinetic energy that scales inversely with the Rossby number:

$$\frac{E_{\mathrm{mag}}}{E_{\mathrm{kin}}} \propto \frac{1}{Ro}\,.$$

Such a magnetostrophic balance sustains large Maxwell stresses that efficiently transport angular momentum across the radiative layer, providing a natural explanation for the flat rotation profiles observed in stellar cores that cannot be reproduced by hydrodynamic processes alone.

In summary, the theoretical framework and numerical evidence discussed in this chapter support the Tayler–Spruit dynamo as a viable and efficient mechanism

for internal angular momentum transport in stellar radiative interiors. With this theoretical foundation established, the next chapter describes the numerical methods and simulation setup we employ to investigate this dynamo process under conditions representative of stellar environments.

Chapter 2

The Xshells Code

2.1 Overview of the Xshells Code

`Xshells` is a high-performance, open-source numerical code designed for simulations of rotating fluid systems in spherical shells, with applications in geophysical and astrophysical magnetohydrodynamics (MHD). It solves the incompressible Navier-Stokes and induction equations using a pseudo-spectral approach, employing spherical harmonics in the angular directions and high-order finite differences in radius. Its flexibility allows modeling diverse physical scenarios, from convective dynamos to forced flows and precessional instabilities.

The code's architecture is modular, separating physical modeling, numerical discretization, and I/O operations. It is implemented in modern C++, with bindings to the SHTns library for fast spherical harmonic transforms and optional GPU acceleration. Time-stepping is handled through second-order predictor-corrector schemes, while pressure and solenoidal constraints are resolved via toroidal-poloidal decomposition. Parallelism is achieved using OpenMP and MPI, ensuring scalability across a wide range of computational platforms.

2.2 Spherical Harmonics and Spectral Expansion

Spherical harmonics $Y_\ell^m(\theta, \phi)$ form an orthonormal basis on the unit sphere and are eigenfunctions of the angular Laplacian operator:

$$\nabla_\Omega^2 Y_\ell^m = -\ell(\ell+1) Y_\ell^m.$$

They allow efficient and accurate representation of scalar and vector fields defined on spherical surfaces. In `Xshells`, fields such as velocity, magnetic field, and temperature are expanded spectrally in θ and ϕ using these harmonics, while

radial dependencies are discretized using finite differences. This hybrid pseudo-spectral method combines the spectral accuracy for smooth fields with the flexibility of local discretization in the radial direction.

Spectral transforms are performed using the SHTns library, which provides highly optimized routines for forward and inverse spherical harmonic transforms. Derivatives in angular coordinates are computed algebraically in spectral space, while nonlinear terms (e.g., advection or Lorentz force) are evaluated in physical space and projected back to spectral space via transforms.

2.3 The SHTns Library for Fast Transforms

`SHTns`, developed by N. Schaeffer Schaeffer 2013 [14], is a key component of `Xshells`' performance. It enables fast, memory-efficient spherical harmonic transforms by computing associated Legendre polynomials on-the-fly using vectorized algorithms. This approach avoids costly pre-computations and leverages cache-friendly memory access patterns.

Unlike conventional $\mathcal{O}(N^3)$ methods, `SHTns` achieves superior performance even when compared to theoretically faster libraries (e.g., those based on Driscoll-Healy grids), due to better real-world optimization. Its support for vector and tensor spherical harmonics allows accurate treatment of solenoidal vector fields using toroidal–poloidal decomposition.

2.4 Code Architecture and Operation

The `Xshells` code is structured into distinct components:

- **Parameter handling**: Inputs are read from configuration files specifying resolution, boundary conditions, and physical parameters.
- **Grid initialization**: The radial grid is built with optional stretching, while angular modes are set up via SHTns.
- **Field representation**: Scalar fields and toroidal/poloidal potentials are stored on the radial grid and expanded spectrally in angular directions.
- **Time integration**: A second-order predictor–corrector (PC2) scheme advances the fields in time. Viscous, Coriolis, and diffusive terms can be treated implicitly.
- **Nonlinear evaluation**: Advection and Lorentz force terms are computed in physical space and projected back.

- **Boundary conditions**: Modular routines allow implementation of no-slip, stress-free, insulating or conducting boundaries.

- **Parallel execution**: MPI and/or OpenMP ensure scalability; GPU support via CUDA/HIP accelerates transforms.

Configuration and input files. Two user-facing files are particularly relevant when configuring simulations with `Xshells`:

- `xshells.hpp` (build-time configuration): a lightweight header used to specialize or enable code paths at compile time (e.g., optional operators, problem-specific hooks, small compile-time constants). In this work, it is used to activate and parameterize the differential-rotation forcing and any ad hoc operators required by the shear driver.

- `xshells.par` (run-time parameters): the text input file parsed at start-up that controls the physical and numerical setup. It defines the dimensionless control parameters (e.g., E, Pr, Pm, Ra, Ro), the spatial and spectral resolution (N_r, $\ell_{\max}$, $m_{\max}$), initial conditions, boundary conditions and others time-integration options. Unless a feature must be decided at compile time, it should be set here.

The timestep Δt is dynamically adapted based on CFL constraints, accounting for flow velocity and Alfvén wave propagation. This ensures stability even in stiff regimes.

2.5 Governing Equations and Poloidal–Toroidal Decomposition

`Xshells` solves the incompressible MHD equations under the Boussinesq approximation in a rotating frame:

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + 2\texttt{Omega0}\mathbf{e}_z \times \mathbf{u} &= -\nabla p^* + \texttt{nu}\nabla^2\mathbf{u} + T\nabla\texttt{Phi} + (\nabla\times\mathbf{b})\times\mathbf{b}, \\
\frac{\partial \mathbf{b}}{\partial t} &= \nabla\times(\mathbf{u}\times\mathbf{b}) + \texttt{eta}\nabla^2\mathbf{b}, \\
\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla(T+T_0) &= \texttt{kappa}\nabla^2 T, \\
\nabla\cdot\mathbf{u} = 0, \qquad \nabla\cdot\mathbf{b} &= 0.
\end{aligned} \tag{2.1}$$

Must be noted that $\mathbf{b}$ is the Alfvén speed, defined as $\mathbf{b} = \mathbf{B}/\sqrt{\rho\mu_0}$, this will serve later on to analyze the results. T_0 is the base temperature profile that corresponds to the diffusive equilibrium. Moreover the following parameters in the equations implemented in `Xshells`: `eta`, `kappa`, `nu`, `Phi`, `Omega0` will be referred later as the controlling parameters of the code and will be defined depending our adimensionalization choices.
Solenoidal fields are represented via the poloidal–toroidal (PT) decomposition,

$$\mathbf{F} = \nabla \times \nabla \times (P_{\mathbf{F}}\,\mathbf{r}) + \nabla \times (T_{\mathbf{F}}\,\mathbf{r})\,.$$

Introducing also the horizontal component of the spherical Laplacian:

$$\Delta_H = -\frac{L_2}{r^2}, \quad L_2 = -\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2}\right]$$

We obtain the scalar identities:

$$\mathbf{r}\cdot\mathbf{F} = L_2 P_{\mathbf{F}}\,, \qquad \mathbf{r}\cdot(\nabla\times\mathbf{F}) = L_2 T_{\mathbf{F}}\,.$$

For general vectors (e.g. nonlinear products) the code uses the QST form

$$F_r = Q_F\,, \quad F_\theta = \frac{1}{\sin\theta}\frac{\partial T_F}{\partial\phi} + \frac{\partial S_F}{\partial\theta}\,, \quad F_\phi = \frac{1}{\sin\theta}\frac{\partial S_F}{\partial\phi} - \frac{\partial T_F}{\partial\theta}\,,$$

with

$$(\nabla\times\mathbf{F})\cdot\mathbf{r} = L_2 T_F\,, \qquad (\nabla\times\nabla\times\mathbf{F})\cdot\mathbf{r} = L_2\left[\frac{Q_F}{r} - \frac{1}{r}\frac{\partial}{\partial r}(rS_F)\right].$$

For the full demonstration look in Appendix A.
This representation allows efficient transforms between spectral and physical space, and straightforward projection of curls/double-curls entering the MHD operators.

2.5.1 MHD equations in PT/QST variables

Projecting Equation 2.1 onto PT/QST variables yields evolution equations for (T_u, P_u) and (T_b, P_b).
Hereafter we present the complete MHD system in PT/QST form, to do so we define the nonlinear vector:

$$\mathbf{f}_{nl} = \mathbf{u}\times(\nabla\times\mathbf{u} + 2\boldsymbol{\Omega}_0) + (\nabla\times\mathbf{b})\times\mathbf{b}\,,$$

For the demonstration look in Appendix B.

$$\left(\frac{\partial}{\partial t} - \nu\Delta\right) T_u = T_{f_{nl}}\,, \tag{2.2}$$

$$\left(\frac{\partial}{\partial t}\Delta - \nu\Delta^2\right) P_u = \left[\frac{Q_{f_{nl}}}{r} - \frac{1}{r}\frac{\partial}{\partial r}\left(rS_{f_{nl}}\right) + \frac{g_0}{r}\hat{T}\right]. \tag{2.3}$$

$$\left(\frac{\partial}{\partial t} - \eta\Delta\right) T_b = \frac{Q_{u\times b}}{r} - \frac{1}{r}\frac{\partial}{\partial r}\left(rS_{u\times b}\right), \tag{2.4}$$

$$\left(\frac{\partial}{\partial t} - \eta\Delta\right) P_b = T_{u\times b}\,. \tag{2.5}$$

$$\left(\frac{\partial}{\partial t} - \kappa\Delta\right) \hat{T} = -\mathbf{u}\cdot(\nabla\hat{T} + \nabla T_0) \tag{2.6}$$

The temperature equations are advanced in physical space with diffusion implicit and advection explicit, consistent with the time-stepping described below.

2.6 Benchmarking and Validation

`Xshells` has been validated many times,particularly for this work we present the dynamo benchmark introduced by Christensen et al. 2001 [15], which remains a widely adopted reference for global-scale geodynamo simulations. This benchmark includes three distinct test cases:

- **Case 0**: Non-magnetic rotating convection.
- **Case 1**: Self-consistent subcritical dynamo solution powered by magneto-convection (insulating boundaries).
- **Case 2**: Like the case before but with insulating mantle and a conductive core.

This structure makes the benchmark both flexible and comprehensive, enabling validation across hydrodynamic, magnetohydrodynamic, and fully dynamical dynamo regimes. In particular, the inclusion of the dynamo case is crucial for validating the coupling between velocity and magnetic fields, which is central to the simulations presented in this work.

Beyond serving as a validation tool, the Christensen et al. 2001 [15] benchmark also provides a natural foundation for the simulations discussed in this study, as further detailed in chapter 3. Its well-characterized setup and diagnostic metrics make it an ideal starting point for exploring dynamo processes in radiative stellar interiors.

Additional code performance benchmarks were compared against those from the BENCHMARK 2 campaign (Matsui et al. 2016 [16]), reinforcing confidence in `Xshells`' robustness and accuracy across diverse numerical platforms.

2.7 Body Force Implementation

In stellar contexts, no-slip boundary conditions are not physically realistic. Instead, free-slip conditions combined with volumetric shear forcing provide a more accurate representation. This approach makes it possible to study how the geometry of the instabilities evolves and how energy saturation changes. Consequently, it offers insight into the associated turbulence and angular momentum transport mechanisms.

Numerically, to solve the right-hand side of the MHD equations in PT/QST form, `Xshells` employs the `explicit_terms` function. This design allows the nonlinear terms to be computed in the spatial domain using a SH-to-SPAT transform; the right-hand side is evaluated there and then transformed back to the spectral domain through a SPAT-to-SH transform. To introduce a body-force term in the Navier–Stokes momentum equation, it is therefore convenient to modify this function by adding the forcing contribution, which we want to be proportional to the shear injected into the system. This implementation provides a source of angular momentum that is distinct from the differential rotation imposed through boundary conditions.

A convenient approach is to generate the volumetric shear directly in the physical domain, because the radial dependence is easier to implement there.

Here we present the solution adopted for the CPU version of the code. The function proposed here is arbitrary and it is not representative of a true shear profile in stellar interiors. While the same general logic applies to the GPU implementation, the evaluation of the right-hand side is performed inside GPU kernels, which makes the interaction between the code structure and the actual execution flow less transparent. This additional layer of complexity slowed the development and debugging process. As a result, the implementation of the volumetric forcing could not be fully validated within the available time.

```
1 static inline double shear_function(double r, double ri, double ro
     )
2 {
3     double A = jpar.shear_amp;  // amplitude of shear
4     double dr = ro - ri;
5     return 0.5 * A * (cos((r - ri)/dr * M_PI) + 1);
6 }
7
8 ...
9
```

```
10 for (int ir = Ulm.irs; ir <= Ulm.ire; ir++) {  // FLUID SHELLS //
11     ...
12 #ifdef XS_SOURCE_TERM
13     const double fr = shear_function(r[ir], ri, ro);
14     for (size_t k = 0; k < nspat; ++k)      // Cycle over the
       radial shells //
15         nl.p[k] += fr;
16 #endif
17     if (nlu_status) {
18         SPAT_SHV3(nl.r, nl.t, nl.p,
19
20     }
21     ...
```

As will be discussed later, running the our test configuration without GPU acceleration requires a larger number of cores than available at the laboratory, especially for the first months of this work. For this reason we were only able to conduct very little testing on the volumetric source term that are not relevant for this work. Consequently, it was not possible to complete a systematic comparison between the reference run and the implementation including volumetric shear forcing.

Chapter 3

Methodology and Work Done

In this chapter, we focus on the practical aspects of running the simulations and the we describe the reference run that we aim to reproduce. We discuss the available computational power along the way and how it influenced the strategy for reaching the target parameter regime. Since our test case required large computational resources and efficient parallelization, multiple computing clusters were utilized to handle the load.

3.1 Methodology

`Xshells` is based on the system of equations described in Equation 2.1. Since this formulation in non-dimensional, it requires defining a base set of units.

In our case, we chose:

- the unit of length to be the outer radius r_o;
- the unit of time to be the viscous timescale, $t = r_o^2/\nu$.
- the unit of temperature to be the imposed temperature difference ΔT between the inner and outer boundaries.

From this choices, it appears evident that viscosity is unitary. With these choices, a dimensional analysis of the `Xshells` equations reveals how to set the problem using the controlling parameters of the code. Remembering Equation 2.1 and keeping in mind that in xshells $\mathbf{b}$ is in unit of velocity. Performing a dimensional analysis we obtain the following relations:

$$\begin{aligned}\left[\frac{U\nu}{L^2}\right]+\left[\frac{U^2}{L}\right]+[\Omega U]&=\left[\frac{U\nu}{L^2}\right]+\left[\frac{\Delta T}{L}\alpha g_0 L\right]+\left[\frac{U^2}{L}\right],\\ \left[\frac{U\nu}{L^2}\right]&=\left[\frac{U^2}{L}\right]+\left[\frac{U\eta}{L^2}\right],\\ \left[\frac{\Delta T\nu}{L^2}\right]+\left[\frac{U\Delta T}{L}\right]&=\left[\frac{\kappa\Delta T}{L^2}\right],\end{aligned}\tag{3.1}$$

Noting that the velocity unit is $U=\nu/r_o$

$$\begin{aligned}\left[\frac{\nu^2}{L^3}\right]+\left[\frac{\nu^2}{L^3}\right]+\left[\Omega\frac{\nu}{L}\right]&=\left[\frac{\nu^2}{L^3}\right]+\left[\frac{\Delta T}{L}\alpha g_0 L\right]+\left[\frac{\nu^2}{L^3}\right],\\ \left[\frac{\nu^2}{L^3}\right]&=\left[\frac{\nu^2}{L^3}\right]+\left[\frac{\nu\eta}{L^3}\right],\\ \left[\frac{\Delta T\nu}{L^2}\right]+\left[\frac{\Delta T\nu}{L^2}\right]&=\left[\frac{\kappa\Delta T}{L^2}\right],\end{aligned}\tag{3.2}$$

After the dimensional analysis, all terms that do not involve control parameters are normalised to unity (this also serves as verification for a coherent scaling). So that only the relevant dimensionless groups remain. Recalling the definitions of the dimensionless parameters introduced in section 1.1, the governing equations can then be rewritten in terms of a reduced set of control parameters.

This procedure allows us to establish a clear correspondence between the numerical model and the underlying physics. By comparing the units and scalings, the control parameters of the code can be identified with the dimensionless numbers that regulate the physical processes being simulated. In this way, the numerical setup is consistently linked to the physical problem through its governing nondimensional quantities.

Recalling the definitions of the dimensionless parameters introduced in section 1.1, we can identify the control parameters of the code with the corresponding dimensionless numbers as follows:

- The control parameter `nu` is set to unity in accordance with our choice of the viscous timescale as the time unit.
- The control parameter `Omega0` is defined as: $\texttt{Omega0}=\left[\frac{\Omega L^2}{\nu}\right]\equiv\frac{1}{E}$.
- The control parameter `Phi` is defined as: $\texttt{Phi}=\left[\frac{\alpha\, g_0\, \Delta T\, L^3}{\nu\,\kappa}\frac{\nu}{\kappa}\right]\equiv\frac{Ra}{Pr}\texttt{radial}$.
- The control parameter `eta` is defined as: $\texttt{eta}=\left[\frac{\eta}{\nu}\right]\equiv\frac{1}{Pm}$.
- The control parameter `kappa` is defined as: $\texttt{kappa}=\left[\frac{\kappa}{\nu}\right]\equiv\frac{1}{Pr}$.

Also, since `Phi` dimensionally is a vector, `radial` is added as it is one of the linear profile available, the choice in justified later.

As a validation for this method, we used the benchmark by Christensen et al. 2001 [15] that uses a different definition of the Rayleigh number and adopts a different length scale. To gain experience with such transformations, we repeated the benchmark case 0 after modifying the length scale from $r_o - r_i$ to r_o. By appropriately rescaling the output parameters (since changing L affects the time unit and therefore the nondimensional energies), we recovered the expected results. This served as a useful verification of our method for adjusting the control parameters according to new adimensionalization choices.

3.2 Reference simulation

The simulation is governed by five key dimensionless numbers: the Ekman number E, Rayleigh number Ra, Prandtl number Pr, magnetic Prandtl number Pm (alternatively the magnetic Reynolds number Rm), and Rossby number Ro (alternatively the ordinary Reynolds number Re).

We chose values for these parameters corresponding to the reference case described by Petitdemange, Marcotte, and Gissinger 2023 [13], which is also the baseline for the extended study in Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1].

We consider an electrically conducting fluid confined between two concentric spherical shells of radius r_i and r_o, rotating about the same axis at different angular velocities (outer sphere rotation Ω, inner sphere rotation $\Omega + \Delta\Omega$). The aspect ratio is $\chi = r_i/r_o = 0.35$. The outer sphere is an electrical insulator while the inner ball has the same electrical conductivity as the fluid domain. A stable stratification is imposed via a fixed temperature difference ΔT between the inner and outer boundary, using the Boussinesq approximation to treat density as constant except in the buoyancy term. In deep stellar interiors, density ρ varies slowly with radius, and the gravitational acceleration can be approximated as $g \propto r$; our setup mimics these conditions using `radial`.

The main reference simulation was run with the following parameters:

- $N_r = 360$ (radial grid points)
- $\ell_{\max} = 256$
- $m_{\max} = 128$
- $E = 10^{-5}$
- $Pm = 1$

- $Pr = 0.1$
- $Ra = 1 \times 10^9$
- $Re = 2.75 \times 10^4$

We note that $\ell_{\max} = 256$ is higher than the value used in the reference article ($\ell_{\max} = 188$). But this is not problematic since the spectral content decays rapidly at high ℓ (by definition of spectral convergence). So it doesn't impact the spectral integration.

3.3 Computing resources

The parameter regime targeted in this study (low E, high Ra, and large imposed differential rotation) is computationally demanding. High spectral resolution and long integration times are required in order to resolve the magnetohydrodynamic coupling and to reach a steady states. Consequently, the simulations could not be performed on local workstations and required access to high-performance computing (HPC) facilities.

At the start of this work, the university's main HPC cluster was temporarily unavailable due to maintenance. Consequently, the project initially relied on departmental resources, in particular the *math5*, *math10*, and *math13* clusters. These systems differ significantly in architecture and performance, which directly influenced the simulation strategy.

math5 is the oldest CPU-based cluster available. Each node consists of two CPU sockets with a moderate core count. The system is configured with the Slurm scheduler, which allows efficient job submission and monitoring. However, the absence of a high-speed interconnect between nodes restricts simulations to single-node runs. This limitation makes large multi-node MPI simulations impossible.

math10 is a more recent CPU cluster equipped with faster processors, larger memory per node, and inter-node communication capabilities. Although the total number of nodes is limited to five, multi-node MPI simulations are possible.

To quantify the performance gain relative to math5, we ran the dynamo benchmark of Christensen et al. 2001 [15] under identical numerical conditions. The wall-clock time per time-step evaluation decreased from approximately 1.4×10^{-3} s on math5 to 8×10^{-4} s on math10, corresponding to nearly a factor of two speedup.

math13 is the laboratory GPU cluster, composed of two multi-GPU nodes. In principle, this architecture provides significant acceleration for spectral MHD simulations. However, initial runs were limited by MPI configuration issues and GPU library compatibility constraints. Multi-GPU execution became fully operational only after resolving these software-level conflicts.

In addition to departmental resources, a temporary allocation was obtained on an external CPU cluster at Aix-Marseille University (AMU). Although limited in total CPU-hours, this allocation enabled completion of intermediate-resolution simulations within acceptable wall-clock times.

Finally, the reference simulation was executed on the university's upgraded cluster *Azzurra*, which features NVIDIA H100 GPU nodes. This system provided the computational throughput required to complete the full-resolution reference run in less than one week.

3.4 Work done

The exploration of the target parameter regime was organized in successive stages, driven by both computational feasibility and scientific goals. As discussed in section 3.3, the early phase of the project was strongly constrained by the temporary unavailability of the main university HPC cluster and by the need to move repeatedly between different computing environments. As a result, the workflow had to be adapted progressively, starting from simpler test cases and gradually approaching the full reference configuration.

From the outset, it was clear that the target parameter regime, in particular the combination of low E, high Ra, and large imposed differential rotation, would require substantially more computing power than what was available on local machines. Initial tests on a personal workstation were therefore limited to compilation checks, short runs, and verification of the basic numerical setup.

A significant amount of time was also devoted to making the code portable across several clusters. In practice, this required resolving a number of typical compatibility issues that arise when deploying the same numerical code on heterogeneous systems: differences between MPI implementations, compiler-dependent behavior, inconsistent module environments, linking problems with FFT libraries, and mismatches between GPU drivers, CUDA toolkits, and the versions expected by the installed libraries. While these steps increased the time required to reach the production regime, they provided valuable experience in dealing with portability and configuration issues in HPC environments.

In addition, changes in architecture often required revalidating compilation flags and rerunning small benchmark cases to ensure that the code produced consistent results on each machine. Although this work was not directly scientific, it played an important role in shaping the overall pace of the project.

To set up the simulation, we started from an existing dynamo benchmark and progressively introduced the modifications required for the Tayler–Spruit dynamo scenario. As discussed in chapter 2, the `Xshells` code provides several example configurations, among which the dynamo benchmark of Christensen et al. 2001 [15]

offers a well-understood and robust starting point.

In particular, we selected case "2" of this benchmark, which allows the magnetic field to extend inside the inner sphere ($r < r_i$). This feature is essential to reproduce the electrical conductivity conditions adopted in the reference study.

Building on this configuration, we first introduced differential rotation of the inner sphere, corresponding to a Couette-type setup. This was achieved by adapting elements of the Couette flow test case provided with the code. We then implemented a forcing term for the inner sphere rotation through the configuration files (`xshells.hpp` and `xshells.par`, see line 24 of Appendix C).

We next modified the thermal forcing by inverting the sign of the imposed temperature gradient relative to the convective dynamo benchmark. This choice establishes a stably stratified configuration, with the inner sphere hotter than the outer boundary. As a result, convective motions are suppressed.

Under these conditions, any dynamo action must arise from magnetohydrodynamic instabilities, such as the Magneto-Rotational Instability (MRI) or the Tayler instability since it cannot be sustained by convective turbulence.

In summary, the final setup consists of a stably stratified, differentially rotating spherical shell and it is designed to isolate and investigate the Tayler–Spruit dynamo mechanism.

3.4.1 Reduced run

After the first setup and benchmark tests, we introduced a reduced case with lower forcing parameters,

$$Ra = 1 \times 10^7, \qquad Re = 5 \times 10^3,$$

before attempting the full reference simulation. This configuration decreases both thermal driving and rotational shear. It therefore reduces the overall dynamical complexity while remaining sufficiently non-trivial to serve as a meaningful intermediate test problem.

The reduced run served three purposes:

- validation of the numerical setup under stable stratification and differential rotation,
- verification of long-time stability and energy diagnostics,
- development and testing of the post-processing and analysis pipeline.

More specifically, this reduced configuration was introduced to clarify whether the main difficulty in reaching the reference regime was primarily physical or computational. The fact that even this milder case remained very slow on the first

departmental clusters indicated quite clearly that the dominant limitation was not the physical model itself, but the available hardware.

Although less demanding than the reference case, the reduced run remained computationally expensive and still required multi-node execution. In order to assess whether the main limitation was physical (i.e. related to the model setup) or purely computational, we adopted a progressive approach.

Starting from the dynamo benchmark of Christensen et al. 2001 [15], we gradually increased the complexity of the simulations by modifying one parameter at a time, moving step by step toward the reduced configuration. This strategy allowed us to isolate the impact of each parameter on the computational cost and to verify that the observed limitations were primarily due to insufficient computing power rather than to issues in the physical setup.

At each stage, the benchmark was first run in simple configurations and then extended to more demanding MPI layouts. This provided a systematic way to test both correctness and scalability, and to quantify how the required computational resources increased with the complexity of the problem.

Even on *math10*, however, the reduced run remained computationally expensive. Using all five available nodes still led to projected runtimes of several weeks to reach a steady regime. GPU acceleration therefore became a necessary rather than optional step.

Once the GPU version of `Xshells` was deployed on *math13*, single-node tests immediately showed a speedup relative to CPU-only runs. The main difficulty was then to obtain stable multi-GPU execution. This required resolving MPI-related communication issues together with incompatibilities between the installed GPU libraries and the versions expected by the code. Only after these problems had been addressed was it possible to perform reliable multi-GPU runs. At that point, the reduced simulation could be completed within a few days.

The reduced run also served as the main dataset for the development of the analysis workflow. Diagnostics, output handling, spectral post-processing, and visualization routines were all tested and refined on this simulation. This preparatory work proved essential, because it ensured that once the full run became feasible, the analysis pipeline was already available and could be applied immediately to production data.

3.4.2 Reference run

After validating the workflow on the reduced case and obtaining access to sufficient GPU resources, we launched the full reference simulation with parameters matching Petitdemange, Marcotte, and Gissinger 2023 [13]. The improved computational capacity of the *Azzurra* cluster enabled completion of the simulation within less than one week.

In practice, however, the full reference run was not attempted continuously throughout the project. It was abandoned relatively early in the first phase, once it became evident that the available departmental resources were not adequate to complete it within a reasonable timescale. The strategy was therefore to postpone the production run and to concentrate instead on reduced simulations, benchmark tests, and technical validation.

The reference case was only resumed near the end of the project, when access to *Azzurra* finally provided the computational power required to treat the problem in its intended regime.

The performance gap between *math13* and *Azzurra* was particularly significant. For comparison, a single node of *Azzurra*, equipped with two NVIDIA H100 accelerators, provided a computational throughput roughly four times larger than the combined performance of the two GPU nodes available on *math13*, each equipped with NVIDIA V100 accelerators.

In turn, *math13* already outperformed the full five-node configuration of *math10*, corresponding to a total of 180 CPU cores, highlighting the substantial advantage of GPU acceleration over CPU-based execution for this type of workload.

These comparisons illustrate the large disparity in effective performance per node and per unit wall-clock time between CPU and GPU architectures. This justifies why the full reference run could only be carried out after access to *Azzurra*, where the available hardware finally matched the computational requirements of the problem.

During this phase, analysis routines previously developed on the reduced run were applied directly to the production output. This ensured a faster diagnostic evaluation of results and comparison with the reference article's findings. In particular, while the reference run was progressing on *Azzurra*, we finalized the post-processing workflow so that once the production simulation ended, its output could be inspected immediately.

To determine the dynamo onset and critical parameters, we consulted Figure 1a of Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1], which shows the critical Rossby number Ro_c for dynamo action as a function of Ra and was obtained using a bisection approach to identify the relevant thresholds. Our aim was to compare our simulation outcomes with the reference results shown in Figure 2 of that article, and also to relate them to the earlier findings of Petitdemange, Marcotte, and Gissinger 2023 [13] (including its supplementary material).

3.4.3 Subcritical experiments

To investigate the subcritical behavior of the dynamo branch, we performed three additional simulations.

The fully developed dynamo state obtained in the reference simulation was used

as an initial condition, while the imposed shear was reduced to

$$Re = 2.25 \times 10^4 \quad \text{and then} \quad Re = 1.75 \times 10^4.$$

These simulations test whether a previously established Tayler–Spruit dynamo can persist when the forcing falls below the nominal onset threshold.

A third simulation was carried out at $Re = 1.75 \times 10^4$ using random initial conditions, identical in structure to those of the reference case. This control run distinguishes between dynamo maintenance and spontaneous excitation.

These additional experiments represent the natural final step of the simulation campaign. The comparison between restart and random-initial-condition runs provides direct evidence regarding the subcritical persistence of the established dynamo.

Chapter 4

Conclusions, Results and Future developments

In this chapter, we present the results obtained from the simulations described in chapter 3 and discuss their implications. We also summarize the main conclusions drawn from this work and outline potential future developments that could build on these.

4.1 Results

Before proceeding to the presentation of the results, it is important to address their nondimensionalization. As for the numerical setup, the way results are expressed depends critically on the chosen scaling. Different studies and numerical codes often adopt different nondimensionalizations, leading to distinct definitions of the control parameters. Consequently, a careful dimensional analysis is required to establish consistent correspondences between these choices. This step is fundamental in order to compare results across different approaches and to interpret them within a common physical framework.

As stated in chapter 2, the code works using the Alfven speed as a unit for the magnetic field. This means that all the magnetic field values are in fact adimensionalized as a velocity.

So in `Xshells`, the dimensionless output amplitude for the magnetic field is:

$$\tilde{b}_{XS} = \frac{B\, r_o}{\nu\sqrt{\rho\mu_0}}.$$

Since we are comparing with results from `PARODY` we must take into account

that this code adimensionalizes the magnetic field differently as.

$$\tilde{B}_P = \frac{B}{\sqrt{\rho\mu_0 r_i r_o \Delta\Omega(\Delta\Omega + \Omega)}}.$$

$$\tilde{B}_P = \frac{\tilde{b}_{XS}\,\nu\sqrt{\rho\mu_0}}{r_o\sqrt{\rho\mu_0 r_i r_o \Delta\Omega(\Delta\Omega + \Omega)}} = \tilde{b}_{XS}\sqrt{\frac{E}{Re(1+Ro)}}.$$

Knowing also that `PARODY` outputs the magnetic energy density and `Xshells` the volume-integrated magnetic energy, we obtain:

$$E_{mag,\text{density},P} = E_{mag,XS}\frac{E}{Re(1+Ro)V}$$

where V is the dimensionless magnetic reference volume.

Also, when presenting the results, it is sometimes more convenient to describe the differential rotation using the Rossby number Ro, and other times using the Reynolds number Re, depending on which physical aspect one wants to highlight. The Rossby number is more directly related to the relative importance of rotation, while the Reynolds number better reflects the strength of the shear and the level of turbulence. In practice, the choice between the two is often guided by the scientific community: astrophysical studies usually adopt the Rossby number, whereas fluid dynamics more commonly uses the Reynolds number.

Given our choice of control parameters, for the reference run, these two quantities are linked as follows:

$$Ro = \frac{Re\,E}{\chi} \sim 0.78,$$

Here we present the results obtained from the main run and the reduced runs described in chapter 3. The analysis primarily focuses on comparing the outcomes of our reference run with the corresponding simulation presented in the reference article, in order to validate our implementation and assess the consistency of the results within the same parameter regime.

Due to time constraints, we were not able to perform a simulation including volumetric shear forcing within the parameter regime of interest for this work. For this reason, such a case is not included in the present analysis. The discussion is therefore restricted to the available runs, which nonetheless provide a solid basis for comparison and for investigating the main physical features of the system.

From Figure 4.1, we can see that the outcome of our main simulation (red point) is very close to the corresponding case in the reference article's Fig. 2.

We did not include our first reduced run (with lower Ra and lower Re) in this comparison plot. Although a branch at $Ra = 10^7$ is present in the reference study, the corresponding integration time is not specified. Without this information, it is

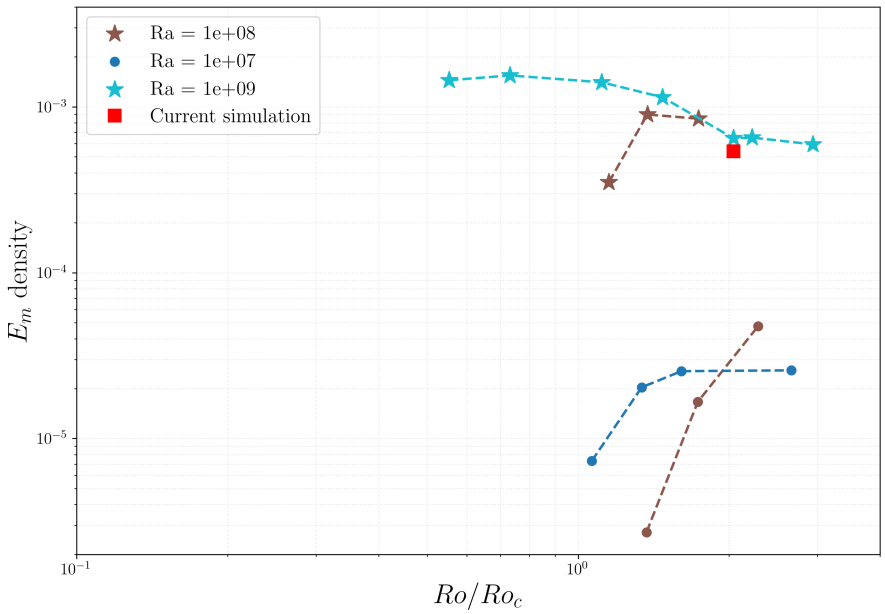

Figure 4.1: Reproduction of Fig. 2 from Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1], with our main run's data overplotted in red for comparison.

not possible to carry out a meaningful comparison with the figure extracted from the reference article, as the system may not have reached a comparable dynamical state.

This reduced run is therefore not shown in the figure. However, even though it does not fall within the Tayler–Spruit dynamo regime (as expected), it proved useful for gaining confidence in the code's response to complex simulations. It will also be used later for a qualitative comparison of the general flow structures.

It should be noted that due to time constraints, our main run was somewhat shorter than ideal. Additionally, we do not have precise information on how the time-averages were computed in the reference studies. These factors explain the modest discrepancy (on the order of 10%) in the measured average magnetic energy density. Nonetheless, such a discrepancy is acceptable given the context, and it does not undermine our conclusions.

In any event, the agreement of the main run with published results serves as a strong validation of our workflow and the `Xshells` code for this application. We achieved the first results of this project: obtaining a working Tayler-Spruit-like dynamo in the intended parameter regime and comparing it with the reference results.

Figure 4.2 shows that once the dynamo is established in the system, reducing Re, and so Ro, does not significantly affect the overall dynamics of the solution. This

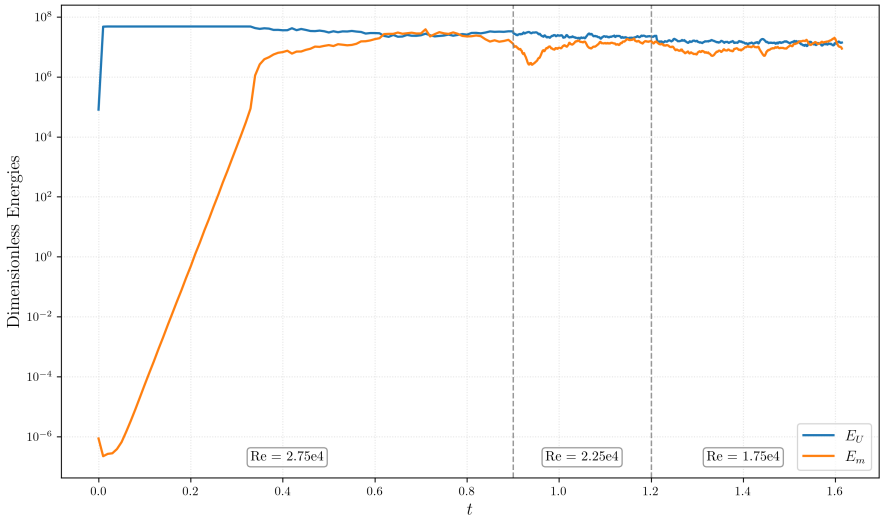

Figure 4.2: Evolution of dimensionless kinetic and magnetic energies over viscous time for three runs (let us remind that both the velocity and the magnetic field are expressed in units of ν/r_o). The second and third runs were restarted from the final state of the previous ones.

behavior is physically expected (as discussed in chapter 1) because the magnetic field, once generated, can continue to be sustained even with a weaker driving rotation. The figure illustrates that the dynamo persists, with the magnetic energy remaining at a strong level after the drop in Re. This result is especially interesting in light of the fact that another simulation with the same final parameters but started from scratch did not produce a dynamo. In other words, Figure 4.3 confirms that if the system is initialized with random small seed fields under subcritical forcing, it fails to amplify those seeds into a sustained dynamo. This successfully replicates the results of Figure 1 of Petitdemange, Marcotte, and Gissinger 2023 [13].

As expected (and in line with Fig. 1b of Petitdemange, Marcotte, and Gissinger 2023 [13]), the first simulation maintained a dynamo even at the lower Rossby number Figure 4.2, whereas the second simulation did not spontaneously develop a dynamo, Figure 4.3.

Returning to the physical context presented in section 1.1, these experiments are important because, in a real star, differential rotation is first generated by evolutionary processes and can then trigger the Tayler instability and the dynamo. Once the dynamo is active, it enhances the transport of angular momentum, which reduces the differential rotation that generated it. For this reason, it is crucial that the mechanism can persist even at subcritical levels, after it has been triggered, if it is to explain the nearly uniform rotation profiles observed in stars.

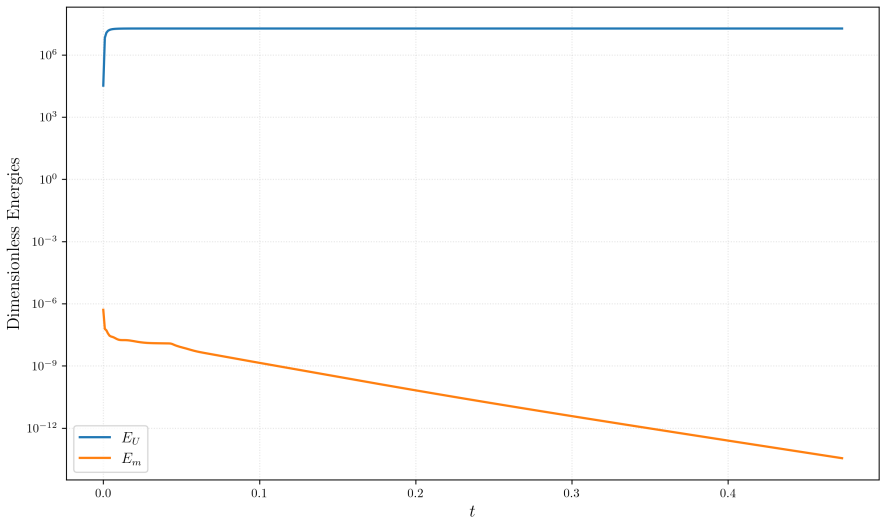

Figure 4.3: Result of the run with $Re = 1.75e4$ started with the same initialization conditions as the main run. No dynamo was observed in this case, as indicated by the decaying magnetic energy.

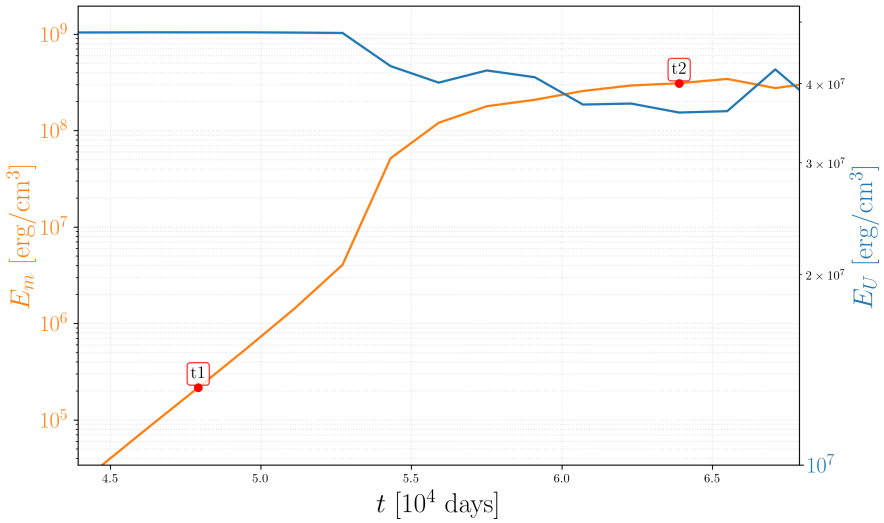

Figure 4.4: Zoom on the energies evolution close to the onset of the Tayler instability, in red the two points where the field snapshots of Figure 4.5 and Figure 4.6 were taken.

Figure 4.4 shows the moment when the shear instability shifted to a Tayler instability, leading to the transition of a turbulent state. In this case dimensional units were used for a better comparison with the reference figure in Petitdemange,

Marcotte, and Gissinger 2023 [13]. We can notice the linear regime in the first part of the image, since the plot is logarithmic on the y-axis we can relate the slope of the magnetic energy to the growth rate (dimensionless $\lambda \sim 90$) of the primary, shear-driven dynamo instability. Moreover we can notice that the kinetic energy has a slight decay at the moment where the Tayler instability sets in, this is due to the fact that part of the kinetic energy is converted into magnetic energy during the instability growth.

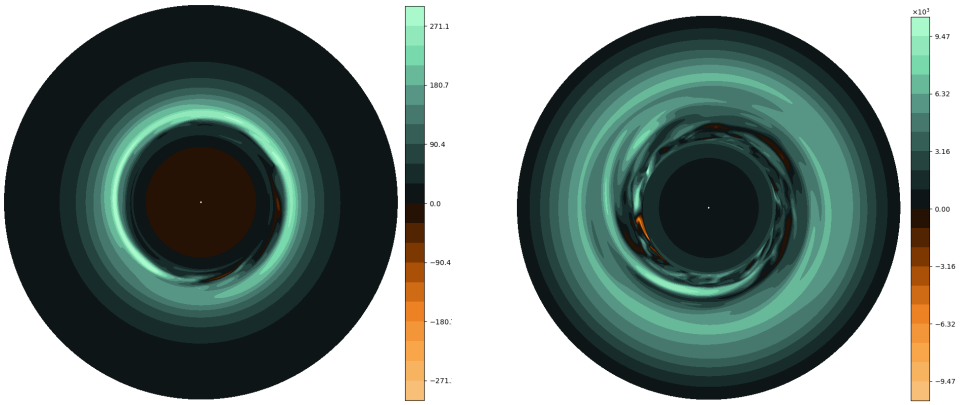

Figure 4.5: Azimuthal magnetic field (scaled as a dimensionless Alfvén velocity) on meridional plane before (left) and after (right) the turbulent transition.

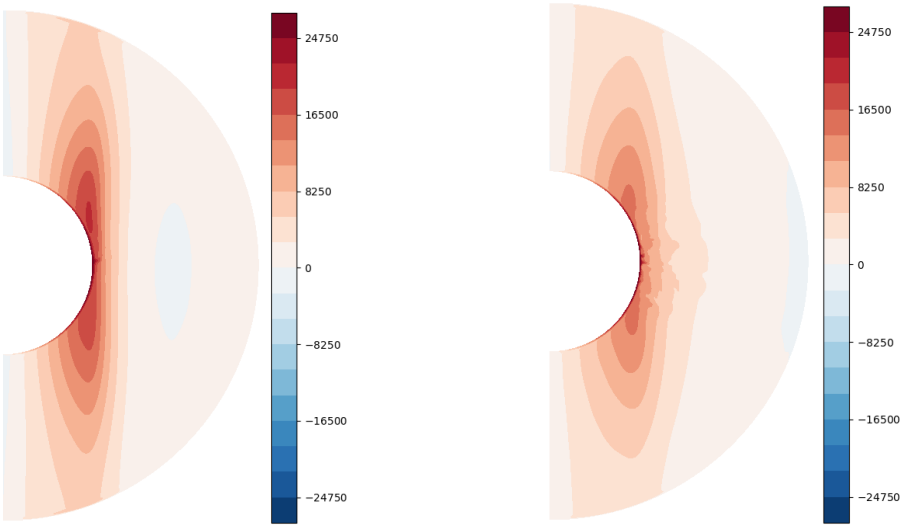

Figure 4.6: Azimuthal velocity field (dimensionless amplitude) on meridional plane before (left) and after (right) the turbulent transition.

Figure 4.6 and Figure 4.5 illustrate the changes in the azimuthal velocity and

magnetic fields before and after the onset of turbulence associated with the dynamo action. The snapshots are taken just before and after the onset of the Tayler instability and are indicated in Figure 4.4 with the two red points.

Here we present a short qualitative comparison of the general structure of the flow and the magnetic field between the reduced run and the reference one. This is not related directly to the Tayler-Spruit dynamo instability but rather to present the different behavior of the flow in the regimes presented in subsection 1.2.2.

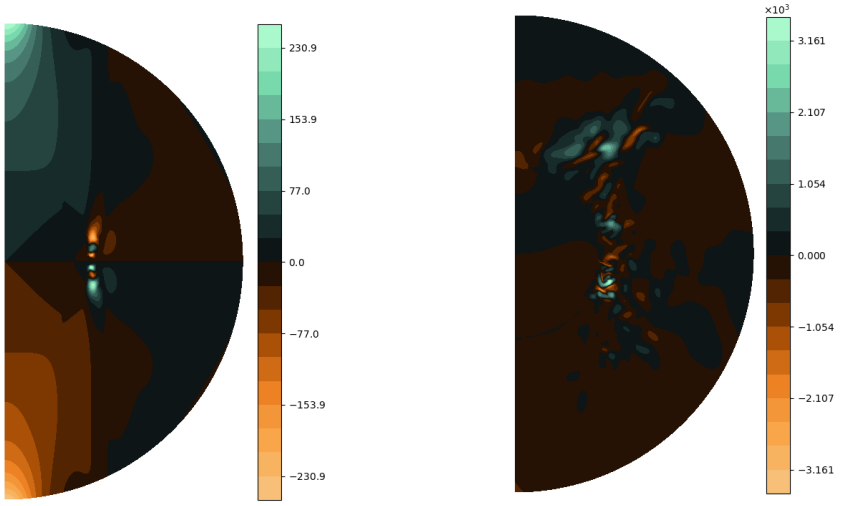

Figure 4.7: Radial magnetic field (scaled as a dimensionless Alfvén velocity) on meridional plane on the reduced (left) vs reference run (right).

Figure 4.7 and Figure 4.8 shows the difference in the magnetic field structure between the reduced run and the reference run. First the reduced run is more laminar and the structure is more axisymmetric, while in the reference run the field is more turbulent and non-axisymmetric. Also the reference run shows slight sign of a more spherical symmetry, this is due to an intensified stratification compared to the rotation rate which is constant in the two simulations. The magnetostrophic regime shows clearly in the vertical lines tangent to the inner sphere, called the tangent cylinder, on both left side images. Finally in Figure 4.8 on the right image, referring to the reference run, it appears clearly the region interested by the strong non-symmetric phenomenon associated with the Tayler-Spruit dynamo in the expected spatial region.

Another point worth mentioning is the computational performance aspect. The reference simulations were performed using the `PARODY` code coupled with the SHTns library. Since we do not have details about the runtime or resources used for those simulations, a direct efficiency comparison between `Xshells` and `PARODY` is not possible.

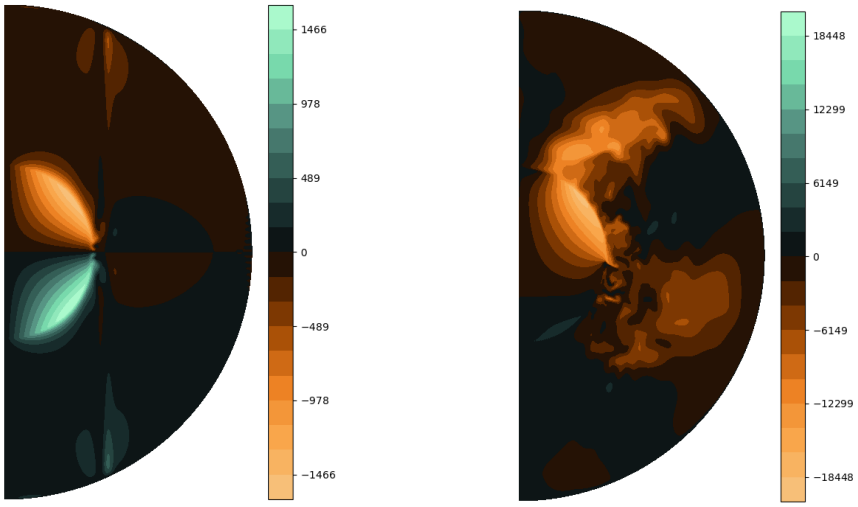

Figure 4.8: Azimuthal magnetic field (scaled as a dimensionless Alfvén velocity) on meridional plane on the reduced (left) vs reference run (right).

4.2 Conclusions

In this work, we carried out large-scale numerical simulations aimed at reproducing the Tayler–Spruit dynamo mechanism in a rotating, stably stratified spherical shell. We developed a dedicated simulation framework based on the `Xshells` code, progressively building complexity from standard benchmark cases up to a high-resolution reference run. This approach allowed us to ensure consistency with the setups of Petitdemange, Marcotte, and Gissinger 2023 [13]; Petitdemange, Marcotte, Gissinger, and Daniel 2024 [1], while adapting the implementation to a different numerical environment.

A significant portion of the project was devoted to overcoming computational challenges. Running global MHD simulations at the required resolution proved to be demanding both in terms of software configuration and hardware performance. We tested the code across multiple HPC architectures, including CPU and GPU clusters, and addressed issues ranging from compilation and library compatibility to memory constraints and runtime stability. This iterative process was essential to identify the most efficient configuration and to ensure the robustness of the simulations. In particular, access to modern GPU nodes on the University cluster played a crucial role, making it possible to carry out the full-scale simulation within a reasonable timeframe.

From a physical perspective, our results provide clear evidence for the existence of a self-sustained Tayler–Spruit-type dynamo under the considered conditions. We recover a threshold for dynamo onset, expressed in terms of Rossby (or equivalently

Reynolds) number, that is consistent with previous studies. Moreover, we show that once the dynamo is established, it can persist even when the driving rotation is reduced, whereas it does not spontaneously develop from trivial initial conditions in that same regime.

Despite these encouraging results, several aspects remain only partially explored. Due to time constraints, a detailed quantitative analysis of angular momentum transport and scaling laws could not be completed, even though these are central elements of the reference studies. Similarly, the implementation of more realistic boundary conditions, such as stress-free boundaries that better mimic stellar interiors, was not achieved within the available timeframe and remains an important direction for future work.

Nevertheless, the present study contributes to the growing body of evidence supporting the Tayler–Spruit dynamo as a viable mechanism for angular momentum transport in radiative stellar zones. Improving the characterization of this process is of particular importance for stellar evolution models. Current prescriptions often rely on simplified or ad-hoc parameterizations, whereas a physically grounded description of the dynamo could provide more robust and predictive modeling of internal rotation profiles. In this context, our results represent a step toward bridging numerical simulations and stellar evolution theory, with the long-term goal of incorporating such mechanisms into evolutionary codes in a consistent and physically motivated way.

Appendix A

Polo-toroidal decomposition

A.1 Useful identities

$$\nabla \times (A\mathbf{r}) = \nabla A \times \mathbf{r}, \quad \nabla \times \nabla \times (A\mathbf{r}) = \nabla \times (\nabla A \times \mathbf{r}) = 2\nabla A - \Delta A\mathbf{r} + (r\frac{\partial}{\partial r}\nabla A)$$

$$\nabla \times \nabla \times \nabla \times (A\mathbf{r}) = -\nabla \times (\Delta A\mathbf{r}),^{1} \quad \nabla \times \nabla \times \nabla \times \nabla \times (A\mathbf{r}) = -\nabla \times \nabla \times (\Delta A\mathbf{r})$$

A.1.1 Vectorial decomposition

Introducing now the poloidal–toroidal decomposition, a restricted form of the Helmholtz decomposition.

If $\mathbf{f}$ is solenoidal ($\nabla \cdot \mathbf{f} = 0$), it can be decomposed as follows:

$$\mathbf{f} = \nabla \times \nabla \times (P_f\mathbf{r}) + \nabla \times (T_f\mathbf{r})$$

Introducing also the horizontal component of the spherical Laplacian:

$$\Delta_H = -\frac{L_2}{r^2}, \quad L_2 = -\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2}\right]$$

In the same manner:

$$\Delta\mathbf{f} = \nabla \times \nabla \times (P_{\Delta\mathbf{f}}\mathbf{r}) + \nabla \times (T_{\Delta\mathbf{f}}\mathbf{r})$$

$$\Delta\mathbf{f} = \underbrace{\nabla(\nabla \cdot \mathbf{f})}_{=0} - \nabla \times \nabla \times \mathbf{f}$$

[1] $\nabla \times (r\frac{\partial}{\partial r}\nabla A) = r\nabla \times (\frac{\partial}{\partial r}\nabla A) + \nabla r \times (\frac{\partial}{\partial r}\nabla A) = 0$

$$\nabla\times\nabla\times(P_{\Delta\mathbf{f}}\mathbf{r})+\nabla\times(T_{\Delta f}\mathbf{r})=-\nabla\times\nabla\times\Big[\nabla\times\nabla\times(P_f\mathbf{r})\Big]-\nabla\times\nabla\times\Big[\nabla\times(T_f\mathbf{r})\Big]$$

$$\nabla\times\nabla\times(P_{\Delta\mathbf{f}}\mathbf{r})+\nabla\times(T_{\Delta f}\mathbf{r})=\nabla\times\nabla\times(\Delta P_f\mathbf{r})+\nabla\times(\Delta T_f\mathbf{r})$$

Thus:

$$P_{\Delta\mathbf{f}}=\Delta P_f,\quad T_{\Delta\mathbf{f}}=\Delta T_f \tag{A.1}$$

First identity:

$$\mathbf{f}\cdot\mathbf{r}=\mathbf{r}\cdot\nabla\times\nabla\times(P_f\mathbf{r})+\underbrace{\mathbf{r}\cdot\nabla\times(T_f\mathbf{r})}_{=0}$$

Expand the First Curl:

$$=\mathbf{r}\cdot\nabla\times\left(0,\frac{1}{\sin\theta}\frac{\partial P_f}{\partial\phi},-\frac{\partial P_f}{\partial\theta}\right)$$

Expand the Second Curl and take scalar product with $\mathbf{r}$:

$$=-\left[(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial P_f}{\partial\theta}\right)+\frac{1}{\sin^2\theta}\frac{\partial^2 P_f}{\partial\phi^2}\right]$$

Finally we identify:

$$\mathbf{f}\cdot\mathbf{r}=L_2P_f \tag{A.2}$$

Second Identity:

$$(\nabla\times\mathbf{f})\cdot\mathbf{r}=\Big[\nabla\times\nabla\times\nabla\times(P_f\mathbf{r})+\nabla\times\nabla\times(T_f\mathbf{r})\Big]\cdot\mathbf{r}=\ ^{2}$$

$$=\nabla\cdot\Big[\mathbf{r}\times\Big(\nabla\times\nabla\times(P_f\mathbf{r})\Big)\Big]+\Big(\underbrace{\nabla\times\nabla\times(P_f\mathbf{r})\cdot\nabla\times\mathbf{r}}_{=0}\Big)+\underbrace{\mathrm{L}_2T_f}_{\text{Prev.}}\ ^{3}$$

Let's analyse the first term, we can rewrite the argoument of the divergence as:

$$\mathbf{r}\times\Big(\nabla\times\mathbf{X}\Big)=\Big(\nabla\mathbf{X}\Big)\cdot\mathbf{r}-\Big(\mathbf{r}\cdot\nabla\Big)\mathbf{X}\quad\text{where }\mathbf{X}=\nabla\times(P_f\mathbf{r})$$

$$=r\nabla X_r-r\frac{\partial}{\partial r}\mathbf{X}$$

[2] $(\nabla\times\mathbf{X})\cdot\mathbf{r}=\nabla\cdot(\mathbf{X}\times\mathbf{r})+(\nabla\times\mathbf{r})\cdot\mathbf{X}$, where $\mathbf{X}=\nabla\times\nabla\times(P_f\mathbf{r})$, Rule 9 of NRL Plasma Formulary

[3] $[\nabla\times\nabla\times(A\mathbf{r})]\cdot\mathbf{r}=L_2A$

Since we know that $X_r = 0$, we can focus on the second term:

$$= -r\frac{\partial}{\partial r}\Big(\nabla \times (P_f \mathbf{r})\Big) = (0, \frac{r}{\sin\theta}\frac{\partial^2 P_f}{\partial\phi\partial r}, -r\frac{\partial^2 P_f}{\partial\theta\partial r})$$

Taking the divergence of this vector:

$$\nabla\cdot\Big(0, \frac{r}{\sin\theta}\frac{\partial^2 P_f}{\partial\phi\partial r}, -r\frac{\partial^2 P_f}{\partial\theta\partial r}\Big) = \frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{r}{\sin\theta}\frac{\partial^2 P_f}{\partial\phi\partial r}\right) + \frac{1}{r\sin\theta}\frac{\partial}{\partial\phi}\left(-r\frac{\partial^2 P_f}{\partial\theta\partial r}\right)$$

$$= \frac{1}{\sin\theta}\left[\frac{\partial^3 P_f}{\partial\theta\partial\phi\partial r} - \frac{\partial^3 P_f}{\partial\phi\partial\theta\partial r}\right] = 0^{4}$$

Finally:

$$\Big(\nabla \times \mathbf{f}\Big)\cdot\mathbf{r} = L_2 T_f \tag{A.3}$$

A.1.2 QST decomposition

Even if a vector is not solenoidal, it can be decomposed as follows:

$$F_r = Q_F$$

$$F_\theta = \frac{1}{\sin\theta}\frac{\partial T_F}{\partial\phi} + \frac{\partial S_F}{\partial\theta}$$

$$F_\phi = \frac{1}{\sin\theta}\frac{\partial S_F}{\partial\phi} - \frac{\partial T_F}{\partial\theta}$$

Single curl identity for QST decomposition

$$(\nabla\times\mathbf{F})\cdot\mathbf{r} = \frac{1}{\sin\theta}\left[\frac{\partial}{\partial\theta}\left(\sin\theta F_\phi\right) - \frac{\partial}{\partial\phi}\left(F_\theta\right)\right]$$

$$= \frac{1}{\sin\theta}\left[\frac{\partial}{\partial\theta}\left(\frac{\partial S_F}{\partial\phi} - \sin\theta\frac{\partial T_F}{\partial\theta}\right) - \frac{\partial}{\partial\phi}\left(\frac{1}{\sin\theta}\frac{\partial T_F}{\partial\phi} + \frac{\partial S_F}{\partial\theta}\right)\right]$$

$$= -\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial T_F}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 T_F}{\partial\phi^2}\right] = L_2 T_F$$

$$(\nabla\times\mathbf{F})\cdot\mathbf{r} = L_2 T_F \tag{A.4}$$

[4] r, θ and ϕ are independent variables, so the mixed derivatives are equal.

Double curl identity for QST decomposition

$$(\nabla \times \nabla \times \mathbf{F}) \cdot \mathbf{r} = \nabla \times \begin{bmatrix} \frac{1}{r \sin\theta}\left(\frac{\partial}{\partial\theta}(\sin\theta\, F_\phi) - \frac{\partial F_\theta}{\partial\phi}\right) \\ \frac{1}{r}\left(\frac{1}{\sin\theta}\frac{\partial Q_F}{\partial\phi} - \frac{\partial}{\partial r}(r\, F_\phi)\right) \\ \frac{1}{r}\left(\frac{\partial}{\partial r}(r\, F_\theta) - \frac{\partial F_r}{\partial\theta}\right) \end{bmatrix} \cdot \mathbf{r}$$

First Term:

$$\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left\{\sin\theta\,\frac{1}{r}\left[\frac{\partial}{\partial r}\left(\frac{r}{\sin\theta}\,\frac{\partial T_F}{\partial\phi} + r\frac{\partial S_F}{\partial\theta}\right) - \frac{\partial Q_F}{\partial\theta}\right]\right\}$$

$$= \frac{1}{r\sin\theta}\left\{\frac{\partial}{\partial r}\left(r\,\frac{\partial^2 T_F}{\partial\phi\partial\theta}\right) + \frac{\partial}{\partial\theta}\left[\sin\theta\frac{\partial}{\partial r}\left(r\frac{\partial S_F}{\partial\theta}\right)\right] - \frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Q_F}{\partial\theta}\right)\right\}$$

Second term:

$$-\frac{1}{\sin\theta}\frac{\partial}{\partial\phi}\left\{\frac{1}{r}\left[\frac{1}{\sin\theta}\,\frac{\partial Q_F}{\partial\phi} - \frac{\partial}{\partial r}\left(\frac{r}{\sin\theta}\,\frac{\partial S_F}{\partial\phi} - r\frac{\partial T_F}{\partial\theta}\right)\right]\right\}$$

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

Together:

$$= \frac{1}{r\sin\theta}\left\{\frac{\partial}{\partial\theta}\left[\sin\theta\frac{\partial}{\partial r}\left(r\frac{\partial S_F}{\partial\theta}\right)\right] - \frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Q_F}{\partial\theta}\right) - \frac{1}{\sin\theta}\frac{\partial^2 Q_F}{\partial\phi^2} + \frac{1}{\sin\theta}\frac{\partial}{\partial r}\left(r\frac{\partial^2 S_F}{\partial\phi^2}\right)\right\}\Bigg|$$

$$= \left\{\frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2} + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right)\right\}\left(\frac{1}{r}\frac{\partial}{\partial r}rS_F\right) - \left\{\frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2} + \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\sin\theta\frac{\partial}{\partial\theta}\right\}\left(\frac{Q_F}{r}\right)\Bigg|$$

$$(\nabla \times \nabla \times \mathbf{F}) \cdot \mathbf{r} = L_2\left[\frac{Q_F}{r} - \frac{1}{r}\frac{\partial}{\partial r}\left(rS_F\right)\right] \tag{A.5}$$

Appendix B

MHD in Polo-Toroidal / QST decomposition

B.1 Navier-Stokes Equations

Equation for T_u

Starting with the conservation of momentum including the Coriolis and Lorentz forces:

$$\frac{\partial u}{\partial t} - \nu\Delta\mathbf{u} = \mathbf{u}\times\nabla\times\mathbf{u} - \nabla p - g_0\hat{T}\mathbf{r} + 2\mathbf{u}\times\boldsymbol{\Omega} + \nabla\times\mathbf{b}\times\mathbf{b}$$

Taking the curl:

$$\frac{\partial}{\partial t}(\nabla\times\mathbf{u}) - \nu\Delta(\nabla\times\mathbf{u}) = \nabla\times\Big[\mathbf{u}\times\nabla\times\mathbf{u} - g_0\dot{T}\mathbf{r} + 2\mathbf{u}\times\boldsymbol{\Omega} + \nabla\times\mathbf{b}\times\mathbf{b}\Big]$$

Scalar with $\mathbf{r}$:

$$\frac{\partial}{\partial t}\Big[(\nabla\times\mathbf{u})\cdot\mathbf{r}\Big] - \nu\Big[(\nabla\times\Delta\mathbf{u})\cdot\mathbf{r}\Big] = \nabla\times\Big[\underbrace{\mathbf{u}\times(\nabla\times\mathbf{u}+2\boldsymbol{\Omega}) + \nabla\times\mathbf{b}\times\mathbf{b}}_{=f_{nl}} - g_0\dot{T}\mathbf{r}\Big]\cdot\mathbf{r}$$ [1]

Since $\mathbf{u}$ is solenoidal, remembering Equation A.3 and Equation A.4:

$$\frac{\partial}{\partial t}L_2T_u - \nu L_2T_{\Delta u} = L_2T_{f_{nl}}$$

or equivalently, using Equation A.1:

$$\left(\frac{\partial}{\partial t} - \nu\Delta\right)T_u = T_{f_{nl}} \tag{B.1}$$

[1] $\nabla\times(g_0\dot{T}\mathbf{r})\cdot\mathbf{r} = 0$

Equation for P_u

Equivalently we can take the double curl of the momentum equation and multiply it by $\mathbf{r}$:

$$\frac{\partial}{\partial t}\Big[(\nabla\times\nabla\times\mathbf{u})\cdot\mathbf{r}\Big]-\nu\Big[(\nabla\times\nabla\times\Delta\mathbf{u})\cdot\mathbf{r}\Big]=[\nabla\times\nabla\times(\mathbf{f}_{nl}-g_0\dot{T}\mathbf{r})]\cdot\mathbf{r}=$$ [2]

$$\frac{\partial}{\partial t}\Big[-\Delta\mathbf{u}\cdot\mathbf{r}\Big]-\nu\Big[-\Delta^2\mathbf{u}\cdot\mathbf{r}\Big]=[\nabla\times\nabla\times(\mathbf{f}_{nl}-g_0\dot{T}\mathbf{r})]\cdot\mathbf{r}$$

Remembering Equation A.1 and Equation A.2, we obtain:

$$L_2\left(\frac{\partial}{\partial t}\Delta-\nu\Delta^2\right)P_u=-[\nabla\times\nabla\times(\mathbf{f}_{nl}-g_0\dot{T}\mathbf{r})]\cdot\mathbf{r}$$

Using Equation A.5, we obtain:

$$\left(\frac{\partial}{\partial t}\Delta-\nu\Delta^2\right)P_u=\left[\frac{Q_{f_{nl}}}{r}-\frac{1}{r}\frac{\partial}{\partial r}\Big(rS_{f_{nl}}\Big)+\frac{g_0}{r}\hat{T}\right]\tag{B.2}$$

B.2 Induction equations

Equation for T_b

Remebering Equation 1.1 and proceding as for the Navier-Stokes equation:

$$\frac{\partial b}{\partial t}+\eta\Big(\nabla\times\nabla\times\mathbf{b}\Big)=\nabla\times\Big(\mathbf{u}\times\mathbf{b}\Big)$$

Taking the curl:

$$\frac{\partial}{\partial t}(\nabla\times\mathbf{b})+\eta\Big(\nabla\times\nabla\times\nabla\times\mathbf{b}\Big)=\nabla\times\nabla\times\Big(\mathbf{u}\times\mathbf{b}\Big)$$

Scalar with $\mathbf{r}$:

$$\frac{\partial}{\partial t}\Big[(\nabla\times\mathbf{b})\cdot\mathbf{r}\Big]+\eta\Big[(\nabla\times\nabla\times\nabla\times\mathbf{b})\cdot\mathbf{r}\Big]=[\nabla\times\nabla\times(\mathbf{u}\times\mathbf{b})]\cdot\mathbf{r}$$

Remembering Equation A.2 and Equation A.3:

$$\frac{\partial}{\partial t}L_2T_b+\eta L_2T_{\Delta b}=[\nabla\times\nabla\times(\mathbf{u}\times\mathbf{b})]\cdot\mathbf{r}$$

Using Equation A.5, we obtain:

$$\left(\frac{\partial}{\partial t}-\eta\Delta\right)T_b=\frac{Q_{u\times b}}{r}-\frac{1}{r}\frac{\partial}{\partial r}\Big(rS_{u\times b}\Big)\tag{B.3}$$

[2] $\Delta\mathbf{A}=\nabla(\nabla\cdot\mathbf{A})-\nabla\times\nabla\times\mathbf{A}$ Rule 14 NRL Plasma Formulary

Equation for P_b

For P we have to take the double curl of the induction equation and multiply it by $\mathbf{r}$:

$$\frac{\partial b}{\partial t} + \eta\Big(\nabla \times \nabla \times \mathbf{b}\Big) = \nabla \times \Big(\mathbf{u} \times \mathbf{b}\Big)$$

Taking the double curl:

$$\frac{\partial}{\partial t}(\nabla \times \nabla \times \mathbf{b}) + \eta\Big(\nabla \times \nabla \times \nabla \times \nabla \times \mathbf{b}\Big) = \nabla \times \nabla \times \Big(\nabla \times \Big(\mathbf{u} \times \mathbf{b}\Big)\Big)$$

Scalar with $\mathbf{r}$:

$$\frac{\partial}{\partial t}\Big[(\nabla \times \nabla \times \mathbf{b}) \cdot \mathbf{r}\Big] + \eta\Big[(\nabla \times \nabla \times \nabla \times \nabla \times \mathbf{b}) \cdot \mathbf{r}\Big] = [\nabla \times \nabla \times (\nabla \times \Big(\mathbf{u} \times \mathbf{b}\Big))] \cdot \mathbf{r}$$

Remembering Equation A.1:

$$L_2 \left(\frac{\partial}{\partial t}\Delta + \eta\Delta^2\right) P_b = [\nabla \times \nabla \times (\nabla \times \Big(\mathbf{u} \times \mathbf{b}\Big))] \cdot \mathbf{r}$$

$$\left(\frac{\partial}{\partial t} - \eta\Delta\right) P_b = T_{u\times b} \tag{B.4}$$

B.3 Temperature equation

$$\left(\frac{\partial}{\partial t} - \kappa\Delta\right) \hat{T} = -\mathbf{u} \cdot (\nabla\hat{T} + \nabla T_0) \tag{B.5}$$

Appendix C

Parameters file

```
1  ######################################################################
2  # XSHELLS : eXtendable Spherical Harmonic Earth-Like Liquid Simulator
3  #  > this is the input parameter file
4  # syntax : name = value
5  ######################################################################
6
7  ### parameters to reproduce the reference run
8
9  job = reference_run
10
11 E = 1e-5
12 Pm = 1
13 Pr = 0.1
14 Ra = 1e9
15 Re = 2.75e4
16 chi = 0.35
17 Ro = Re * E / chi
18
19 ### PHYSICAL PARAMS ###
20 Omega0 = 1/E              # Global rotation rate
21 nu = 1.0                 # viscosity
22 kappa = 1/Pr              # thermal diffusivity
23 eta = 1/Pm                # magnetic diffusivity
24 DeltaOmega = Ro *  Omega0     # Differsantial Rotation
25 t_forcing = 0             # Constant forcing
26
27 ### INITIAL FIELDS ###
```

```
28
29 u = 0
30 b = random      # initial magnetic field
31 tp = 0       # initial temperature field
32 tp0 = delta     # imposed (base) temperature field
33 phi0 = radial*Ra/Pr    # radial gravity field
34
35 ### BOUNDARY CONDITIONS AND RADIAL DOMAINS ###
36 BC_U = 1,1     # inner,outer boundary conditions (0=zero velocity, 1=no-
       slip, 2=free-slip)
37 BC_T = 1,1     # 1=fixed temperature, 2=fixed flux.
38 R_U = chi : 1   # Velocity field boundaries
39 R_T = chi : 1   # Temperature field boundaries
40 R_B = 0 : 1   # Magnetic field boundaries
41
42 ### NUMERICAL SCHEME ###
43 NR = 360   # total number of radial shells (overriden by r = ...)
44 N_BL = 2,2  # number of radial shells reserved for inner and outer
       boundaries.
45 dt_adjust = 1 # 0: fixed time-step (default), 1: variable time-step
46 dt = 1e-5   # time step
47 iter_max = 100  # iteration number (total number of text and energy file
       ouputs)
48 sub_iter = 500  # sub-iterations (the time between outputs = 2*dt*
       sub_iter is fixed even with variable time-step)
49 iter_save = 10  # number of iterations between field writes (if movie >
       0, see below).
50
51 ### SHT ###
52 Lmax = 256  # max degree of spherical harmonics
53 Mmax = 128  # max fourier mode (phi)
54 Mres = 1  # phi-periodicity.
55 #Nlat = 32  # number of latitudinal points (theta). Optimal chosen if
       not specified.
56 #Nphi = 16  # number of longitudinal points (phi). Optimal chosen if not
       specified.
57
58 ### OPTIONS ####
59 #interp = 1   # 1: allow interpolation of fields in case of grid mismatch.
       0: fail if grids mismatch (default).
```

```
restart = 1   # 1: try to restart from a previous run with same name. 0:
    no auto-restart (default).
movie = 1   # 0=field output at the end only (default), 1=output every
    iter_save, 2=also writes time-averaged fields
#lmax_out = -1    # lmax for movie output (-1 = same as Lmax, which is
    also the default)
#mmax_out = -1    # mmax for movie output (-1 = same as Mmax, which is
    also the default)
#prec_out = 1   # 1: single precision movie output (default), 2: double
    precision movie outputs.
#zavg = 2     # 1=output z-averaged field, 2=also output rms z-averaged.
backup_time = 300   # ensures that full fields are saved to disk at least
    every backup_time minutes, for restart.
#nbackup = 0     # number of backups before terminating program (useful
    for time-limited jobs). 0 = no limit (default)

### ALGORITHM FINE TUNING ###

## C_vort and C_alfv control the time-step adjustment (active if
    dt_adjust=1),
## regarding vorticity and Alfven criteria.
## if dt_tol_lo < dt/dt_target < dt_tol_hi, no adjustement is done.
#C_u = 0.7
#C_vort = 0.3   # default: 0.2
#C_alfv = 1.1   # default: 1.0
#dt_tol_lo = 0.7    # default: 0.8
#dt_tol_hi = 1.2    # default: 1.1

## fine-tuning of PC2 time-scheme:
#pc2_ax = 0.4   # Most stable = 0.4. Most accurate = 0.5 (default).
    Should be >= 0.4
#pc2_ai = 0.99   # Most stable = 0.99. Most accurate = 0.5 (default).
    Should be >= 0.5

## sht_type : 0 = gauss-legendre (default), 1 = fastest with DCT enabled
##         2 = fastest on regular grid, 3 = full DCT, 4 = debug (
    quick_init), 6 = on-the-fly (good for parallel)
##  4 has the smallest init-time, useful for test/debug. Otherwise 0 or
    6 should be used.
sht_type = 6

```

```
89 ## sht_polar_opt_max = SHT polar optimization threshold : polar
       coefficients below that threshold are neglected (for high ms).
90 ##   value under wich the polar values of the Legendre Polynomials Plm
       are neglected, leading to increased performance.
91 ##   0 = no polar optimization;  1.e-14 = VERY safe (default);  1.e-10 =
       safe;  1.e-6 = aggresive.
92 sht_polar_opt_max = 0
```

common/xshells.par.Ra9_H100_mpi

Bibliography

[1] L. Petitdemange, F. Marcotte, C. Gissinger, and F. Daniel. *Tayler-Spruit Dynamos in Simulated Radiative Stellar Layers*. Jan. 17, 2024. DOI: 10.48550/arXiv.2306.11711. Pre-published (cit. on pp. i, 2–4, 9–11, 22, 27, 31, 36).

[2] C. Aerts, S. Mathis, and T. Rogers. «Angular Momentum Transport in Stellar Interiors». In: (Aug. 18, 2019). DOI: 10.1146/annurev-astro-091918-104359 (cit. on p. 1).

[3] S. Basu. «Global Seismology of the Sun». In: (Dec. 2016). DOI: 10.1007/s41116-016-0003-4 (cit. on p. 1).

[4] R. Howe. «Solar Interior Rotation and Its Variation». In: (2009). DOI: 10.12942/lrsp-2009-1 (cit. on p. 1).

[5] H. C. Spruit. *Differential Rotation and Magnetic Fields in Stellar Interiors*. July 12, 1999. DOI: 10.48550/arXiv.astro-ph/9907138. Pre-published (cit. on pp. 1, 8).

[6] P. Eggenberger. *Mixing Processes in Stars*. Sept. 17, 2024. DOI: 10.48550/arXiv.2409.11354. Pre-published (cit. on p. 1).

[7] J. Braithwaite and H. C. Spruit. *Magnetic Fields in Non-Convective Regions of Stars*. Apr. 7, 2017. DOI: 10.48550/arXiv.1510.03198. Pre-published (cit. on pp. 1, 9).

[8] R. J. Tayler. «The Adiabatic Stability of Stars Containing Magnetic Fields-I: TOROIDAL FIELDS». In: (Apr. 1, 1973). DOI: 10.1093/mnras/161.4.365 (cit. on pp. 2, 8).

[9] H. C. Spruit. «Dynamo Action by Differential Rotation in a Stably Stratified Stellar Interior». In: (Jan. 2002). DOI: 10.1051/0004-6361:20011465 (cit. on pp. 2, 4, 8–10).

[10] J. Fuller, A. L. Piro, and A. S. Jermyn. «Slowing the Spins of Stellar Cores». In: (May 21, 2019). DOI: 10.1093/mnras/stz514 (cit. on pp. 2, 9).

[11] J. Braithwaite. «The Stability of Poloidal Magnetic Fields in Rotating Stars». In: (July 2007). DOI: 10.1051/0004-6361:20065903 (cit. on p. 2).

[12] J.-P. Zahn, A. S. Brun, and S. Mathis. «On Magnetic Instabilities and Dynamo Action in Stellar Radiation Zones». In: (Oct. 2007). DOI: 10.1051/0004-6361:20077653 (cit. on p. 2).

[13] L. Petitdemange, F. Marcotte, and C. Gissinger. «Spin-down by Dynamo Action in Simulated Radiative Stellar Layers». In: (Jan. 20, 2023). DOI: 10.1126/science.abk2169 (cit. on pp. 2, 9, 10, 22, 26, 27, 32, 33, 36).

[14] N. Schaeffer. «Efficient Spherical Harmonic Transforms Aimed at Pseudospectral Numerical Simulations». In: (Mar. 2013). DOI: 10.1002/ggge.20071 (cit. on p. 14).

[15] U. R. Christensen et al. «A Numerical Dynamo Benchmark». In: (Dec. 2001). DOI: 10.1016/S0031-9201(01)00275-8 (cit. on pp. 17, 22–24, 26).

[16] Hiroaki Matsui et al. «Performance Benchmarks for a next Generation Numerical Dynamo Model». In: (May 2016). DOI: 10.1002/2015GC006159 (cit. on p. 18).