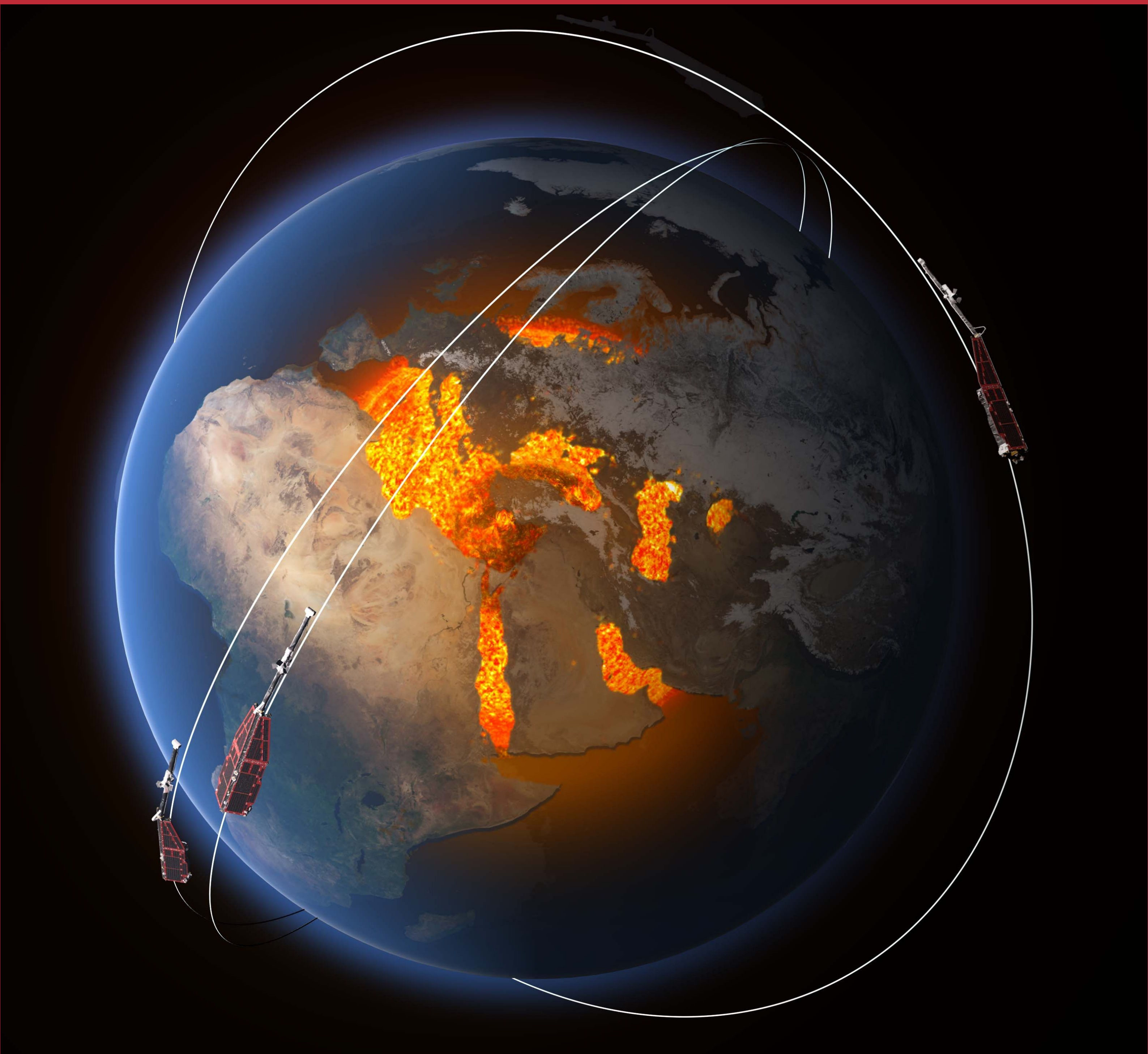
DTU Space
National Space Institute

Core Flows Inferred From SWARM Satellite Magnetic Data

Master Thesis

Author: Gustav Alexander Jørgensen, s143656

Supervisor: Chris Finlay, DTU Space



Core Flows Inferred From SWARM Satellite Magnetic Data

Master Thesis
January, 2021

By
Gustav Alexander Jørgensen

Cover image: European Space Agency
Published by: DTU, National Space Institute, Centrifugevej, Building 356, 2800 Kgs. Lyngby Denmark
www.space.dtu.dk
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

This thesis has been prepared over five months at the Geomagnetism division of DTU Space at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

Gustav Alexander Jørgensen - s143656

....

Signature

....

Date

Abstract

Observed gradual changes in the geomagnetic field, known as secular variation, are believed to be governed by the flow of liquid metal in the outer region of the Earth's core, near the core-mantle boundary. These core flows and the secular variation observed above the Earth's surface are related through the magnetic induction equation. Satellite and ground observatory measurements of secular variation and the induction equation thus present an inverse problem for determining core flows at the core-mantle boundary. Kloss and Finlay 2019 previously presented a method for solving this inverse problem for the period 2000 to 2018, by parametrizing core flow as a series of normal modes of rapidly rotating flow in a spherical container. In this study, we extend their method by further allowing for smaller-scale, equatorially anti-symmetric flows and accounting for likely contributions of magnetic diffusion to the observed secular variation. We implement this modified method with a new regularization scheme for the inverse problem, and by augmenting the model vector to include secular variation due to diffusion. We apply it to SWARM satellite data covering the period 2014-2019. We find that allowing for more small-scale equatorial anti-symmetry (localized equator crossings) and diffusion allows us to estimate flows that well explain the observed secular variation with flows that are similar to, but simpler than, those described by Kloss and Finlay 2019. In particular, we find a predominantly steady, planetary-scale, eccentric gyre of westward flow along with inter-annual reversals of low-latitude azimuthal flow. We conclude that flows with significant local equator crossings and contributions from diffusion provide consistent explanations of the secular variation observed from 2014 to 2019, demonstrating the non-uniqueness of the inverse problem while adding to the evidence for the robustness of the aforementioned flow features.

Acknowledgements

I want to thank Chris Finlay for his excellent guidance as my supervisor throughout this study and for providing me with the geomagnetic secular variation data as well as the CHAOS-7.2 geomagnetic field model. I also want to thank Clemens Kloss for providing me with his MATLAB code, which served as a crucial foundation for further development in this study. Finally, I would like to thank Magnus Hammer for valuable communications regarding the secular variation data and Olivier Barrois for providing me with dynamo realizations from the Coupled Earth Model.

Contents

1 Introduction

The Earth's magnetic field is produced by three main sources: Dynamo action in the Earth's core, induced and remanent magnetization of the Lithosphere, and current systems in the Ionosphere and Magnetosphere driven by solar wind (see figure 1.1).

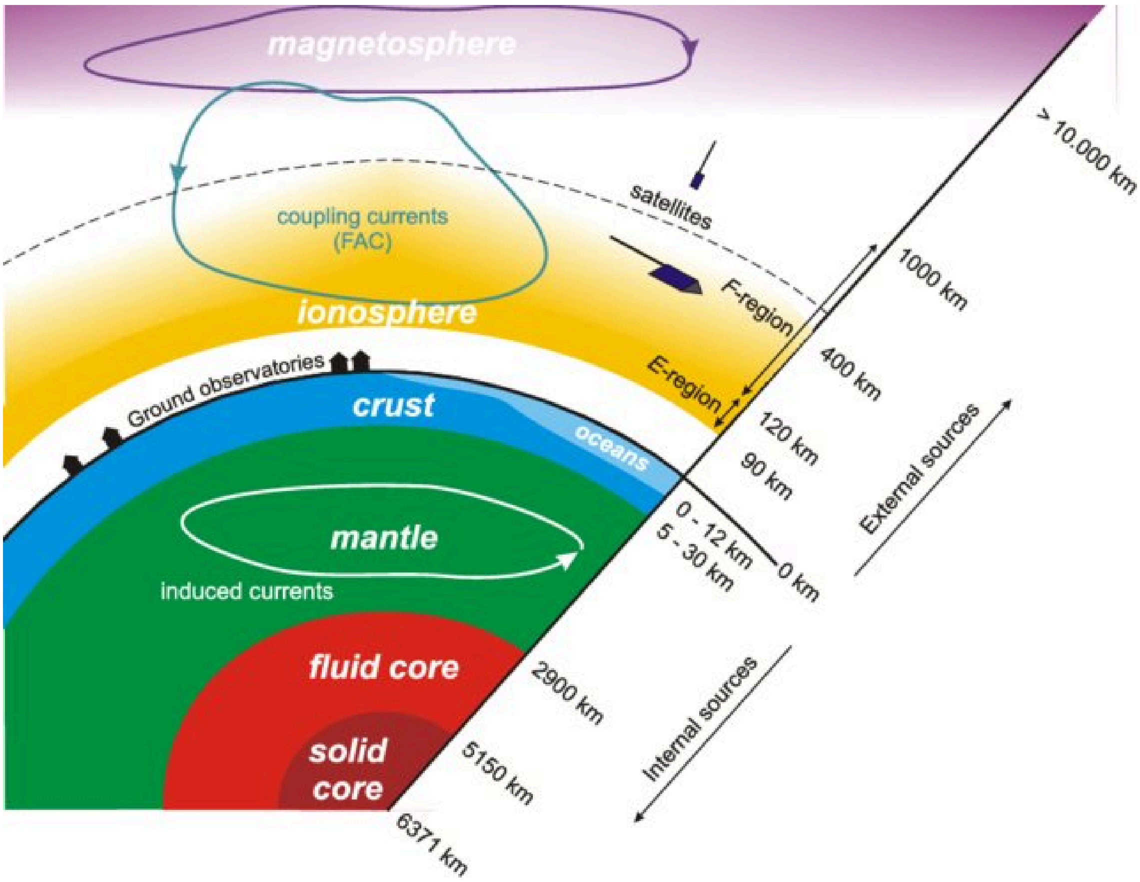


Figure 1.1: Sources of the geomagnetic field. Figure taken from Olsen et al. 2010.

The core field is also known as the main field, because it is the source of about 95% of the magnetic field observed on the Earth's surface. This main field is believed to be generated by dynamo action because core temperatures are too hot for permanent magnetization. Permanent magnetization can also be excluded because the field undergoes slow but noticable changes on yearly to decadal timescales (Finlay et al. 2010). These temporal changes are also referred to as secular variation (SV) and are driven by the dynamic flow of molten, conductive metals in the outer core. This flow is mainly driven by cooling of the core from its hot, initial state and freezing of the liquid outer core to the solid inner core, which releases buoyant material carrying latent heat. These energy sources, the flows they drive, and the Earth's rotation are all necessary components of a self-sustaining dynamo.

SV observed from satellites and ground observatories contains information about the core flow. Specifically, SV can be related to the velocity of conducting material through the induction equation, which we introduce in chapter 2. The SWARM satellites represent the latest generation of satellites intended to study the geomagnetic field and its temporal evolution (Olsen et al. 2013). The SWARM trio was launched by ESA in 2013 and consists of the identical satellites Alpha, Bravo, and Charlie, each carrying both an absolute scalar magnetometer and a vector magnetometer, as seen in figure 1.2. They orbit the Earth about 15 times daily, resulting in excellent spatial coverage. This study uses data derived from measurements by the vector magnetometers which were developed by DTU Space. Apart from SWARM data, vector magnetometer data from INTERMAGNET[1] ground observatories are also used.

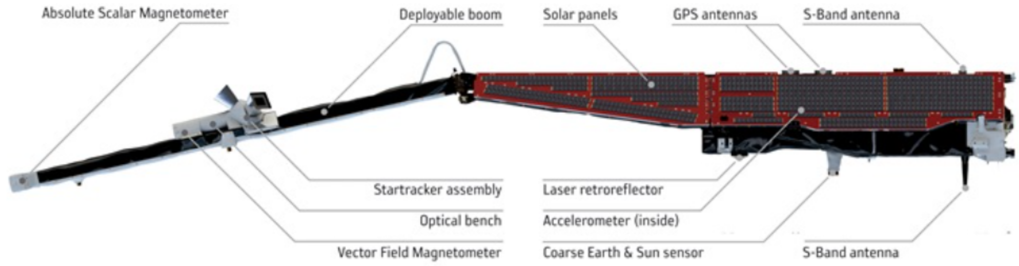


Figure 1.2: A SWARM satellite and its components.

The following study aims to further the research of Kloss and Finlay 2019 by replicating their method of estimating flow in the core from observations of SV and further developing it with a new and improved regularization scheme for the inversion of the induction equation. We aim to modify the regularization to produce models that allow more flow that is anti-symmetric with respect to the geographical equator on short length scales, such that local equator crossings are permitted. Numerical simulations of core dynamics show that such flows are possible, even in the presence of rapid rotation (Schaeffer and Pais 2011). We also aim to produce models that can account for possible contributions from magnetic diffusion, where part of the observed SV can result from large field gradients within the core, and the finite electrical conductivity (see chapter 2). Core flow will be modelled at the top of the core, i.e. the Core-Mantle Boundary (CMB) for the period spanning September 2014 to September 2019 at low latitudes. This study thus makes use of all available SV data from the SWARM satellite mission, as well as the most up-to-date data from ground observatories at the time of writing. Kloss and Finlay 2019 modelled flow for the period 2000 to 2018 and also made use of data from the CHAMP satellite which went out of commision in 2010. Since this study only aims to model flow for the period currently covered by SWARM data, we enjoy the benefits of a more consistent data set and avoiding the period of no satellite coverage from 2010 to 2013. Similar to Kloss and Finlay 2019, we solve this highly non-unique inverse problem by parametrizing the flow as a series of normal modes of rapidly rotating core flow, as described by K. Zhang and Liao 2017.

The scientific aim of this study is to investigate whether the inter-annual alternations of azimuthal flow at equatorial latitudes indicated e.g. under the Pacific by previous studies (e.g. Kloss and Finlay 2019; Gillet et al. 2015) survive the introduction of short-scale anti-symmetric flows and diffusion. This would demonstrate that such flows are plausible explanations of the observed SV and further demonstrate the robustness of the phenomenon of inter-annual, azimuthal flow alternations.

[1] www.intermagnet.org

In chapter 2, we present the theoretical background of this study, including (but not limited to) the induction equation and mathematical formulations of the above mentioned modes. Towards the end of the chapter, we present formulations of the forward and inverse problems linking observed SV with the flow of liquid metal at the top of the core. In chapter 3, we introduce and briefly investigate the SV data used in this study. In chapter 4, we present the results of the preferred models. In chapter 5 the results are discussed and we investigate possible error sources and compare to other research before final conclusion are drawn in chapter 6.

2 Theory

This chapter presents the theory of geomagnetic SV due to core processes, core flow parameterization, and the regularization scheme used for inversion to produce the models presented in chapter 4.

2.1 The Induction Equation

This study aims to estimate the flow in the Earth's core through inversion of geomagnetic SV data, using the magnetic induction equation. This section will therefore present a derivation and examination of this crucial equation.

The molten metal in the Earth's core flows with some velocity, $\boldsymbol{u}$, and is moving through the geomagnetic field, $\boldsymbol{B}$. This gives rise to an effective electric field

$$\boldsymbol{E}' = \boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B} \tag{2.1}$$

Inserting the effective electric field into Ohm's law, we have

$$\boldsymbol{J} = \sigma \boldsymbol{E}' => \boldsymbol{E}' = \frac{1}{\sigma}\boldsymbol{J} \tag{2.2}$$

where $\boldsymbol{J}$ is the current density and σ is the conductivity of the molten metal. Substituting this expression into equation 2.1 gives

$$\boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B} = \frac{1}{\sigma}\boldsymbol{J} => \boldsymbol{E} = \frac{1}{\sigma}\boldsymbol{J} - \boldsymbol{u} \times \boldsymbol{B} \tag{2.3}$$

Under the quasi-static approximation, Ampere's law relates the curl of the magnetic field and the current density as

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J} => \boldsymbol{J} = \frac{1}{\mu_0}(\nabla \times \boldsymbol{B}) \tag{2.4}$$

where μ_0 is the magnetic permeability constant. Inserting this expression into equation 2.3 gives

$$\boldsymbol{E} = \eta(\nabla \times \boldsymbol{B}) - \boldsymbol{u} \times \boldsymbol{B} \tag{2.5}$$

where $\eta = 1/\mu_0\sigma$ is the magnetic diffusivity. Faraday's law relates the curl of the electric field to time changes of the magnetic field

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{2.6}$$

Inserting equation 2.5 into equation 2.6 and making use of the identity

$\nabla \times \nabla \times \boldsymbol{B} = \nabla(\nabla \cdot \boldsymbol{B}) - \nabla^2 \boldsymbol{B} = -\nabla^2 \boldsymbol{B}$ then gives

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) + \eta \nabla^2 \boldsymbol{B} \tag{2.7}$$

which is the magnetic induction equation. The first term expresses the induction (generation) of the magnetic field, while the second term expresses its diffusion (dissipation). Note that both the electric field and the current density have been eliminated, so that the evolution of the magnetic field is only a function of the magnetic field itself and the velocity of the conductor.

If there is no movement of conducting material, $\boldsymbol{u} = \boldsymbol{0}$, the induction term becomes zero, and we are left with only diffusion.

$$\frac{\partial \boldsymbol{B}}{\partial t} = \eta \nabla^2 \boldsymbol{B} \tag{2.8}$$

We then have a solution $\boldsymbol{B} \propto e^{-t/\tau_d}$. Using $L = 1000km$ as a typical length scale in the Earth's core and a conductivity of $\sigma = 5 \cdot 10^5 S/m$ results in a typical decay-time of

$$\tau_d = \frac{L^2}{\eta} = \sigma \mu_0 L^2 = 6.3 \cdot 10^{11} s \approx 20.000 yr. \tag{2.9}$$

Conversely, allowing for movement of conducting material with a conductivity $\sigma \to \infty$, i.e. $\eta \to 0$, the diffusion term becomes zero and we are left with only induction

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) \tag{2.10}$$

This is called the frozen flux approximation, because the magnetic field lines follow the flow and are thus "frozen" in the core. Using a typical flow velocity of $|\boldsymbol{U}| = 10km/yr$, the frozen flux equation can then be expressed with typical quantities as

$$\frac{|\boldsymbol{B}|}{\tau_\nu} = \frac{|\boldsymbol{U}| \cdot |\boldsymbol{B}|}{L} => \tau_\nu = \frac{L}{|\boldsymbol{U}|} = \frac{1000km}{10km/yr} = 100yr \tag{2.11}$$

where τ_ν is a typical timescale of induction. Justification for use of the frozen flux approximation is typically given with use of the magnetic Reynolds number, which is the ratio between the two terms of the induction equation. For the timescales estimated above the magnetic Reynold's number is

$$R_m = \frac{|\nabla \times (\boldsymbol{u} \times \boldsymbol{B})|}{|\eta \nabla^2 \boldsymbol{B}|} = \frac{\tau_d}{\tau_\nu} = 200 \tag{2.12}$$

The frozen flux approximation is generally considered valid when $R_m >> 1$. This holds for large length scales of the Earth's core, such as $1000km$. It is however conceivable that contributions from diffusion on smaller length scales (particularly with regard to the unknown radial length scales) could be significant and ultimately have some impact on SV observed above the Earth's surface. The end of this Theory chapter will present the attempt in this study to take diffusion into account when estimating the core flow. It should be noted that although the contribution from diffusion will be considered non-zero, the frozen flux approximation is still used as the primary part of the inversion scheme.

2.2 Core Flow at the CMB Including Small-Scale Error

Assuming that the CMB is a closed spherical container enclosing the fluid in the Earth's core, it is possible to simplify the frozen flux version of the induction equation (equation 2.10) for flow at the CMB to (e.g. Bloxham and Jackson 1991)

$$\frac{\partial B_r}{\partial t} = -\nabla_H \cdot (\boldsymbol{u}_H B_r) \tag{2.13}$$

where $\nabla_H = [\partial/\partial\theta\ ;\ \partial/\partial\phi]$ is the horizontal part of the nabla operator. This is possible because radial flows are prevented by the CMB. If the radial field and the SV of the radial field are known, it is thus possible to estimate the core flow at the CMB. Unfortunately, the small-scale part of the main field is obscured by the lithospheric field, when measured above the Earth's surface. As in Kloss and Finlay 2019 we handle this by decomposing the radial field in equation 2.13 into a large-scale part, $\overline{B_r}$, and a small-scale part, $\tilde{B_r}$ which can not be resolved from observations, so

$$\frac{\partial B_r}{\partial t} = -\overline{\nabla_H \cdot \left(\boldsymbol{u}_H \overline{B_r}\right)} + \boldsymbol{e} \tag{2.14}$$

where

$$\boldsymbol{e} = -\overline{\nabla_H \cdot \left(\boldsymbol{u}_H \tilde{B_r}\right)} \tag{2.15}$$

is the SV produced by the unresolved small-scale radial field interacting with the flow. We refer to $\boldsymbol{e}$ as the small-scale error.

2.3 Expressing Core Field and SV Using Spherical Harmonics

Under the so-called quasi-stationary assumption, significant SV is assumed to happen slowly enough, so that the pre-Maxwell equations are reasonable approximations. The pre-Maxwell equations are

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J} \tag{2.16}$$

and

$$\nabla \cdot \boldsymbol{J} = 0 \tag{2.17}$$

In current-free regions, $\boldsymbol{J} = 0$, the magnetic field has no curl, according to equation 2.16. This implies that the magnetic field is a Laplacian potential field

$$\boldsymbol{B} = -\nabla V \tag{2.18}$$

where V is the scalar potential. Inserting this expression into Gauss' law, $\nabla \cdot \boldsymbol{B} = 0$, gives

$$\nabla \cdot \boldsymbol{B} = -\nabla^2 V = 0 \tag{2.19}$$

The potential is thus a solution to Laplace's equation

$$\nabla^2 V = 0 \tag{2.20}$$

Solutions to Laplace's equation in spherical geometry include so-called Spherical Harmonics (SH). Since we wish to model core flow at the CMB using measurements above the Earth's surface, we are only interested in the SH solution to Laplace's equation for *internal* sources. That solution is

$$V(r,\theta,\phi) = r_{surf} \sum_{n=1}^{N} \sum_{m=0}^{n} \left(\frac{r_{surf}}{r}\right)^{n+1} \left(g_n^m cos\left(m\phi\right) + h_n^m sin\left(m\phi\right)\right) P_n^m\left(cos\left(\theta\right)\right) \tag{2.21}$$

where r is the distance from Earth's center, θ is co-latitude, and ϕ is longitude, $r_{surf} = 6371.2km$ is the reference radius of the Earth, and P_n^m is the associated Legendre function. n and m respectively denote the degree and order of the solution. N is then the truncation of n. A higher degree solution will be more complex in that it resolves smaller length scales of the field. Finally, g_n^m and h_n^m are the so-called gauss coefficients. Given equation 2.21 it is possible to define a model of the geomagnetic field with the gauss coefficients alone. The geomagnetic field can then be evaluated at any depth or altitude by adjusting the parameter r. Downward continuation of the field from surface measurements necessitates the assumption that the mantle is insulating (e.g. Barrois et al. 2019). The field can also be evaluated at any point on the sphere with radius r, by adjusting the parameters θ and ϕ.

In spherical geometry, the gradient of a scalar, f, is given by

$$\nabla f = \frac{\partial f}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial f}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{1}{r \cdot sin\theta}\frac{\partial f}{\partial \phi}\hat{\boldsymbol{\phi}} \tag{2.22}$$

An expression for the large-scale radial main field is thus found by inserting the potential from equation 2.21 into equation 2.18, such that

$$\overline{B_r}\left(r,\theta,\phi\right)=-\frac{\partial V}{\partial r}=\sum_{n=1}^{N_{mf}}\sum_{m=0}^{n}\left(n+1\right)\left(\frac{r_{surf}}{r}\right)^{n+2}\left(g_n^m cos\left(m\phi\right)+h_n^m sin\left(m\phi\right)\right)P_n^m\left(cos\left(\theta\right)\right) \tag{2.23}$$

with N_{mf} denoting the truncation degree of the large-scale main field. The small-scale main field, subject to the small-scale error, can then be expressed as

$$\tilde{B}_r\left(r,\theta,\phi\right)=\sum_{n=\left(N_{mf}+1\right)}^{N_e}\sum_{m=0}^{n}\left(n+1\right)\left(\frac{r_{surf}}{r}\right)^{n+2}\left(\tilde{g}_n^m cos\left(m\phi\right)+\tilde{h}_n^m sin\left(m\phi\right)\right)P_n^m\left(cos\left(\theta\right)\right) \tag{2.24}$$

where N_e denotes the upper truncation degree of the small-scale field. The radial SV can also be found with an identical expression, but using SV gauss coefficients, $\dot{g}_n^m$ and $\dot{h}_n^m$, such that

$$\frac{\partial B_r}{\partial t}\left(r,\theta,\phi\right)=\sum_{n=1}^{N_{sv}}\sum_{m=0}^{n}\left(n+1\right)\left(\frac{r_{surf}}{r}\right)^{n+2}\left(\dot{g}_n^m cos\left(m\phi\right)+\dot{h}_n^m sin\left(m\phi\right)\right)P_n^m\left(cos\left(\theta\right)\right) \tag{2.25}$$

where N_{sv} denotes a separate truncation degree for the SV.

As was implied by equation 2.13, this study requires model predictions of the core field and SV at the CMB to estimate the flow. It is assumed that the large-scale part of the core field itself is known, and given by a field model. Here, the CHAOS-7.2 geomagnetic field model is used. A detailed presentation of the model's construction can be found in Finlay et al. 2020. The model spans the period 1999-2020 and is derived from revised monthly means of ground observatory measurements as well as data from the Ørsted, CHAMP, Cryosat-2, and SWARM satellites. CHAOS-7.2 is a time-dependent spherical harmonic model of both the internal and external fields. Since this study aims to model core flow, only the internal part of the model was used, i.e. the previously mentioned gauss coefficients of the geomagnetic field, g_n^m and h_n^m. The SV as defined by $\dot{g}_n^m$ and $\dot{h}_n^m$, is also used in some tests. These can be inserted into equations 2.23 and 2.25 with $r=3485km$ as the estimated radius of the core to obtain the desired predictions.

For all models in this study, we use truncation degrees $N_{mf}=14$, $N_{sv}=16$, and $N_e=30$.

2.4 The Navier-Stokes Momentum Equation

We now turn to theory for flow on the Earth's core. Consider a closed, rotating, liquid-filled container. In a rotating reference frame with axes fixed in the container, the Navier-Stokes momentum equation would be expressed as

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}+2\boldsymbol{\Omega}\times\boldsymbol{u}+\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\boldsymbol{r}\right)\right]=-\nabla p+\rho\boldsymbol{g}+\mu\left[\nabla^2\boldsymbol{u}+\frac{1}{3}\nabla\left(\nabla\cdot\boldsymbol{u}\right)\right]+\rho\boldsymbol{r}\times\left(\frac{\partial\boldsymbol{\Omega}}{\partial t}\right)+\rho\boldsymbol{f} \tag{2.26}$$

where ρ is the density of the fluid, $\mathbf{\Omega}$ is the angular velocity, $\boldsymbol{r}$ is the position vector of a fluid element, p is the pressure imposed on that fluid element, μ is the coefficient of dynamic viscosity (assumed constant over space and time), and $\boldsymbol{f}$ is any external forcing. In this scenario the flow, $\boldsymbol{u}$, is the velocity relative to the rotating frame.

If the range of variation in temperature and pressure is small, they can be considered independent of one another. The pressure can then be treated as linearly dependent on the difference between the temperature, T, and some reference temperature, T_0, such that

$$\rho = \rho_0 \left[1 - \alpha \left(T - T_0\right)\right] \tag{2.27}$$

where ρ_0 is the density at T_0 and α is the thermal expansion coefficient of the liquid. Applying this expression, as well as the Boussinesq approximation

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.28}$$

which assumes that the variations in density throughout the liquid are sufficiently small to be ignored, the momentum equation then becomes

$$\left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + 2\mathbf{\Omega} \times \boldsymbol{u}\right] = -\frac{1}{\rho_0}\nabla P - \boldsymbol{g}\alpha \left(T - T_0\right) + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{r} \times \left(\frac{\partial \mathbf{\Omega}}{\partial t}\right) + \boldsymbol{f} \tag{2.29}$$

where $\nu = \mu/\rho_0$ is the coefficient of kinematic viscosity and

$$P = p - p_0 - \frac{p_0}{2}\left(\mathbf{\Omega} \times \boldsymbol{r}\right) \cdot \left(\mathbf{\Omega} \times \boldsymbol{r}\right) \tag{2.30}$$

is the departure from the reference pressure, including the contribution from the centrifugal force.

Further assumptions can be made to simplify the momentum equation. This study applies a simplified momentum equation to the Earth's core by making the following assumptions:

1. The core is rapidly rotating, such that the Coriolis force dominates and controls the fluid dynamics.
2. The core is filled with a homogeneous fluid.
3. The fluid is an ideal fluid.
4. The reference frame, i.e. the mantle, rotates at a constant angular velocity.
5. There are no external forcings.
6. The core flow deviates only slightly from solid body rotation.
7. The CMB is considered a closed spherical container.

Assumptions 2 to 5 respectively set the last four terms on the right-hand side of equation 2.29 equal to zero. Furthermore, assumption 6 allows for setting $\boldsymbol{u} \cdot \nabla \boldsymbol{u} = 0$. Equation 2.29 thus becomes

$$\frac{\partial \boldsymbol{u}}{\partial t} + 2\boldsymbol{\Omega} \times \boldsymbol{u} = -\frac{1}{p_0} \nabla P \tag{2.31}$$

A spherical coordinate system (r, θ, ϕ) is now introduced with $r = 0$ at the center of the core, $r = 1$ at the CMB and $\theta = 0$ at the axis of rotation, z. For the angular velocity $\boldsymbol{\Omega} = \Omega \hat{\boldsymbol{z}}$, using the inverse angular velocity, Ω^{-1} as the time unit and the radius of the core, r_0, as length unit, equation 2.31 is reduced to the non-dimensional form

$$\frac{\partial \boldsymbol{u}}{\partial t} + 2\hat{\boldsymbol{z}} \times \boldsymbol{u} = -\frac{1}{p_0} \nabla P \tag{2.32}$$

where $\hat{\boldsymbol{z}}$ is the unit vector in the z-direction.

Assumption 7 provides the following boundary condition

$$\hat{\boldsymbol{r}} \cdot \boldsymbol{u} = 0 \quad \text{at} \quad r = 1 \tag{2.33}$$

Equations 2.32 and 2.33 then define a boundary value problem for the flow, $\boldsymbol{u}$.

Under the previously listed assumptions, oscillatory motions can occur in the flow. K. Zhang and Liao 2017 show that solutions to this specific problem can be expressed as a linear combination of a single axisymmetric, geostrophic, steady mode and an infinite number of time-dependent, inertial modes. More on this in the following section.

2.5 Modes

2.5.1 Geostrophic Modes

The geostrophic mode is a steady flow solution to the boundary value problem defined by equations 2.32 and 2.33. For a steady flow, time-dependence is ignored, so the momentum equation (2.32) becomes

$$2\hat{\boldsymbol{z}} \times \boldsymbol{u} = -\frac{1}{p_0} \nabla P \tag{2.34}$$

Taking the curl of both sides of this expression results in the Taylor-Proudman theorem

$$\frac{\partial \boldsymbol{u}(\boldsymbol{r})}{\partial z} = 0 \tag{2.35}$$

which states that the flow of the rapidly rotating fluid is uniform along the rotation axis, z. For the flow to obey both equation 2.35 and 2.33 requires that the flow is purely azimuthal at the CMB. The geostrophic mode can thus be described as an infinite series of geostrophic polynomials, following the expression (K. Zhang and Liao 2017)

$$\boldsymbol{u}^G(r, \theta) = \sum_{k=1} a_k^G G_{2k-1}(r, \theta)\, \hat{\boldsymbol{\phi}} \tag{2.36}$$

where $\boldsymbol{u}^G$ is the geostrophic flow, a_k^G are unknown geostrophic coefficients, G_{2k-1} are the geostrophic polynomials, and k determines the degree of those polynomials. The geostrophic polynomials are expressed as

$$G_{2k-1}(r,\theta) = \sum_{j=1}^{k} \frac{(-1)^{k-j}\,[2(k+j)-1]!!}{2^{k-1}(k-j)!\,(j-1)!\,(2j)!!}(r\sin\theta)^{2j-1} \tag{2.37}$$

2.5.2 Inertial Modes

For inertial, time-varying flows, oscillatory solutions to equation 2.32 presented by K. Zhang and Liao 2017 are of the form

$$\boldsymbol{u}(\boldsymbol{r},t) = \boldsymbol{u}(\boldsymbol{r})\,e^{i2\sigma t} \tag{2.38}$$

where t is time, i is the imaginary number $\sqrt{-1}$, σ is the half-frequency of oscillatory motions, bounded by $0 < |\sigma| < 1$, and $\boldsymbol{r}$ is a position vector. For our polar spherical coordinate system $\boldsymbol{r} = (r,\theta,\phi)$, so the solution form becomes

$$\boldsymbol{u}((r,\theta,\phi),t) = \boldsymbol{u}(r,\theta,\phi)\,e^{i2\sigma t} = [u_r(r,\theta,\phi), u_\theta(r,\theta,\phi), u_\phi(r,\theta,\phi)]\,e^{i2\sigma t} \tag{2.39}$$

where u_r, u_θ, and u_ϕ are the vector components of the total flow, $\boldsymbol{u}$. The modes that make up the inertial flow solutions can be classified according to their symmetry with respect to the equatorial plane. The two classes are the equatorially symmetric and the equatorially anti-symmetric modes. Both of these can further be classified according to their symmetry with respect to the rotation axis, as either axisymmetric or non-axisymmetric. The solutions, which will be presented shortly, are all series, which run over the natural integer indices m, n, and k. As these indices increase, the solutions become more complex. The index m is a measure of the number of oscillations in the azimuthal, ϕ-direction. The modes with $m = 0$ are thus axisymmetric and those with $m \geq 1$ are non-axisymmetric. The index $n \geq 1$ is a measure of complexity in the z-direction, and the index $k \geq 0$ is a measure of the complexity as you move away from the z-axis towards the core surface, parallel to the equatorial plane.

2.5.2.1 Equatorially Symmetric Modes

The equatorially symmetric solutions follow the symmetry

$$(u_r, u_\theta, u_\phi)(r,\theta,\phi) = (u_r, -u_\theta, u_\phi)(r,\pi-\theta,\phi) \tag{2.40}$$

and the vector components of the solutions are written as

$$\begin{aligned}\hat{\boldsymbol{r}}\cdot\boldsymbol{u}^S_{mnk} = -\frac{i}{2}\sum_{i=0}^{k}\sum_{j=0}^{k-i}\frac{C^S_{mk;ij}}{r}\left[\sigma^2(m+2j)+m\sigma-2i\left(1-\sigma^2\right)\right]\\ \cdot\left[r^{m+2(i+j)}\sigma^{2i-1}\left(1-\sigma^2\right)^{j-1} sin^{m+2j}\theta cos^{2i}\theta\right]e^{im\phi}\end{aligned} \tag{2.41a}$$

$$\hat{\boldsymbol{\theta}} \cdot \boldsymbol{u}^S_{mnk} = -\frac{i}{2} \sum_{i=0}^{k} \sum_{j=0}^{k-i} \frac{C^S_{mk;ij}}{r} \left\{ \left[\sigma^2 \left(m + 2j \right) + m\sigma \right] cos^2\theta + 2i \left(1 - \sigma^2 \right) sin^2\theta \right\}$$
$$\cdot \left[r^{m+2(i+j)} \sigma^{2i-1} \left(1 - \sigma^2 \right)^{j-1} sin^{m+2j-1}\theta cos^{2i-1}\theta \right] e^{im\phi} \tag{2.41b}$$

$$\hat{\boldsymbol{\phi}} \cdot \boldsymbol{u}^S_{mnk} = -\frac{1}{2} \sum_{i=0}^{k} \sum_{j=0}^{k-i} \frac{C^S_{mk;ij}}{r} \left[\left(m + 2j \right) + m\sigma \right]$$
$$\cdot \left[r^{m+2(i+j)} \sigma^{2i} \left(1 - \sigma^2 \right)^{j-1} sin^{m+2j-1}\theta cos^{2i}\theta \right] e^{im\phi} \tag{2.41c}$$

where the coefficients $C^S_{mk;ij}$ are given by

$$C^S_{mk;ij} = \frac{\left(-1 \right)^{i+j} \left[2 \left(m + k + i + j \right) - 1 \right]!!}{2^{j+1} \left(2i - 1 \right)!! \left(k - i - j \right)! i! j! \left(m + j \right)!} \tag{2.42}$$

and the half-frequencies, σ, are the roots of one of two polynomials, depending on whether or not the mode is axisymmetric. For the axisymmetric modes ($m = 0$), the half-frequencies of equation 2.41 are given as the roots of

$$0 = \sum_{j=0}^{k-1} \left\{ \frac{\left(-1 \right)^j \left[2 \left(2k - j \right) \right]!}{j! \left[2 \left(k - j \right) - 1 \right]! \left(2k - j \right)!} \right\} \sigma^{2(k-j)} \tag{2.43}$$

where k varies over all positive integers ≥ 2. The above polynomial has $k - 1$ positive solutions. For a given k, we thus have $k - 1$ axisymmetric modes. Note that each mode, $\boldsymbol{u}_{mnk}$, then contains a unique half-frequency. The subscript denoting the indices, σ_{mnk} has been dropped for easier readability, however. Each root of the polynomial corresponds to a separate n in this subscript, such that

$0 < |\sigma_{01k}| < |\sigma_{02k}| < |\sigma_{03k}| < ... < |\sigma_{0nk}| < ... < |\sigma_{0(k-1)k}| < 1$

For non-axisymmetric modes ($m \geq 1$), the half-frequencies are given as the roots of

$$0 = \sum_{j=0}^{k-1} \left\{ \frac{\left(-1 \right)^{j+k} \left[2 \left(2k + m - j \right) \right]!}{j! \left[2 \left(k - j \right) \right]! \left(2k + m - j \right)!} \left[\left(2k + m - 2j \right) \sigma - 2 \left(k - j \right) \right] \right\} \sigma^{2(k-j)-1}$$
$$+ \frac{m \left[2 \left(k + m \right) \right]!}{k! \left(k + m \right)!} \tag{2.44}$$

where k varies over all positive integers ≥ 1. The above polynomial has $2k$ real solutions, meaning we have $2k$ non-axisymmetric modes for a given k, where

$0 < |\sigma_{m1k}| < |\sigma_{m2k}| < |\sigma_{m3k}| < ... < |\sigma_{mnk}| < ... < |\sigma_{m(2k)k}| < 1$

The special subset of symmetric modes with σ_{m1k} ($n = 1$) have the slowest periods and are referred to as quasi-geostrophic (QG) because they are almost invariant along the rotation axis. This subset includes the axisymmetric mode with σ_{01k}.

2.5.2.2 Equatorially Anti-Symmetric Modes

The equatorially anti-symmetric solutions follow the symmetry

$$(u_r, u_\theta, u_\phi)(r, \theta, \phi) = (-u_r, u_\theta, -u_\phi)(r, \pi - \theta, \phi) \tag{2.45}$$

and the vector components of the solutions are written as

$$\begin{aligned} \hat{\boldsymbol{r}} \cdot \boldsymbol{u}^A_{mnk} = &-\frac{i}{2} \sum_{i=0}^{k} \sum_{j=0}^{k-i} C^A_{mk;ij} \left[\sigma^2 (m + 2j) + m\sigma - (2i+1)\left(1 - \sigma^2\right)\right] \\ &\cdot \left[\sigma^{2i-1}\left(1-\sigma^2\right)^{j-1} sin^{m+2j}\theta cos^{2i+1}\theta\right] r^{m+2(i+j)} e^{im\phi} \end{aligned} \tag{2.46a}$$

$$\begin{aligned} \hat{\boldsymbol{\theta}} \cdot \boldsymbol{u}^A_{mnk} = &-\frac{i}{2} \sum_{i=0}^{k} \sum_{j=0}^{k-i} C^A_{mk;ij} \left\{\left[\sigma^2 (m + 2j) + m\sigma\right] cos^2\theta + (2i+1)\left(1 - \sigma^2\right) sin^2\theta\right\} \\ &\cdot \left[\sigma^{2i-1}\left(1-\sigma^2\right)^{j-1} sin^{m+2j-1}\theta cos^{2i}\theta\right] r^{m+2(i+j)} e^{im\phi} \end{aligned} \tag{2.46b}$$

$$\begin{aligned} \hat{\boldsymbol{\phi}} \cdot \boldsymbol{u}^A_{mnk} = &-\frac{1}{2} \sum_{i=0}^{k} \sum_{j=0}^{k-i} C^A_{mk;ij} \left[(m + 2j) + m\sigma\right] \\ &\cdot \left[\sigma^{2i}\left(1-\sigma^2\right)^{j-1} sin^{m+2j-1}\theta cos^{2i+1}\theta\right] r^{m+2(i+j)} e^{im\phi} \end{aligned} \tag{2.46c}$$

where the coefficients $C^A_{mk;ij}$ are given by

$$C^A_{mk;ij} = \frac{(-1)^{i+j}\left[2(m+k+i+j)+1\right]!!}{2^{j+1}(2i+1)!!\,(k-i-j)!i!j!\,(m+j)!} \tag{2.47}$$

As was the case for the symmetric modes, the half-frequencies of equation 2.46 are also given by one of two polynomials depending on whether or not the mode is axisymmetric.

For the axisymmetric modes ($m = 0$), the half-frequencies are the roots of the polynomial

$$0 = \sum_{j=0}^{k} \left\{\frac{(-1)^j\left[2(2k-j+1)\right]!}{j!\left[2(k-j)\right]!\,(2k-j+1)!}\right\} \sigma^{2(k-j)} \tag{2.48}$$

where k varies over all positive integers. The above polynomial has $2k$ real solutions, meaning we have $2k$ axisymmetric modes, where

$0 < |\sigma_{01k}| < |\sigma_{02k}| < |\sigma_{03k}| < ... < |\sigma_{0nk}| < ... < |\sigma_{0(2k)k}| < 1$

For non-axisymmetric modes ($m \geq 1$), the half-frequencies are the roots of

$$0 = \sum_{j=0}^{k} \frac{(-1)^j \left[2\left(2k+m-j+1\right)\right]!}{j!\left[2\left(k-j\right)+1\right]!\left(2k+m-j+1\right)!} \cdot \left[\left(2k-2j+m+1\right)\sigma - \left(2k-2j+1\right)\right]\sigma^{2(k-j)} \tag{2.49}$$

where k varies over all integers ≥ 0. The above polynomial has $2k+1$ solutions, meaning we have $2k+1$ non-axisymmetric modes, where

$0 < |\sigma_{m1k}| < |\sigma_{m2k}| < |\sigma_{m3k}| < ... < |\sigma_{mnk}| < ... < |\sigma_{m(2k+1)k}| < 1$

2.5.2.3 Summary of Inertial Mode Computation

The half-frequencies, σ_{mnk}, of the corresponding inertial modes, $\boldsymbol{u}_{mnk}$, are found by calculating the roots of the polynomials in equations 2.43-2.44 and 2.48-2.49 (K. Zhang and Liao 2017). The half-frequencies are then respectively inserted into equations 2.41 and 2.46 to calculate the corresponding mode.

2.5.3 Visualization

Figure 2.1 shows examples of each type of mode, i.e. a geostrophic mode, $\boldsymbol{u}_4^G$ (a), an equatorially symmetric mode, $\boldsymbol{u}_{312}^S$ (b), and an equatorially anti-symmetric mode, $\boldsymbol{u}_{312}^A$ (c). The figure shows the general structure of each type. Geostrophic modes are entirely azimuthal and thus invariant along the axis of rotation. The equatorially symmetric mode has a non-zero flow in the azimuthal, radial, and longitudinal directions. Note that the particular one shown here is a QG mode ($n = 1$) and is thus nearly invariant along the rotation axis. The equatorially anti-symmetric mode also has a non-zero flow in all three directions but is inverted across the equator. Equatorially anti-symmetric modes therefore tend to produce gyres centered on the equator, which necessitates equator crossings of the flow. With a linear combination of the different types of modes it is possible to describe unique flows with distinct features on a spherical surface.

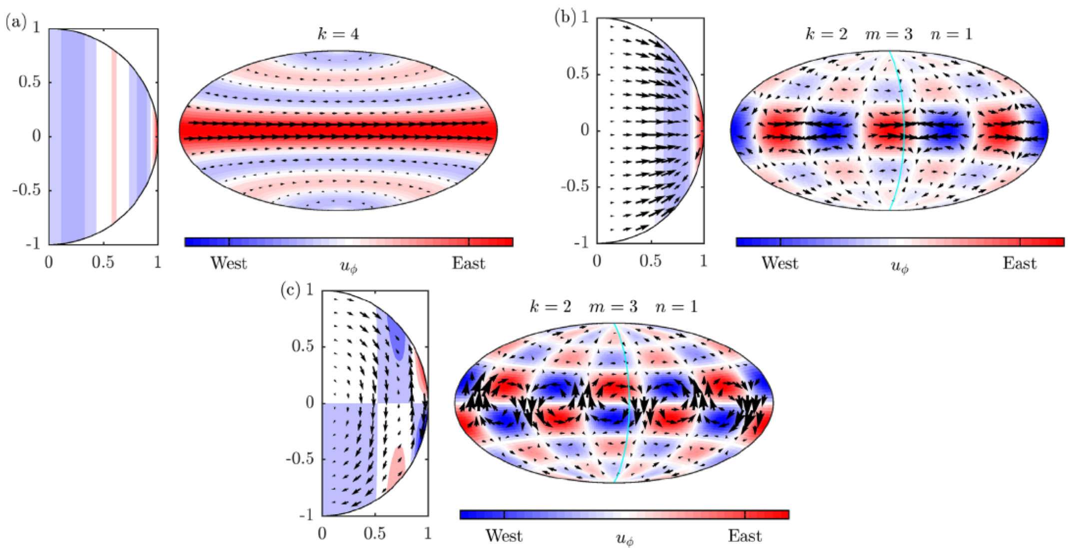


Figure 2.1: Examples of a geostrophic mode (a), an equatorially symmetric mode (b), and an equatorially anti-symmetric mode (c), as seen in the meridional plane (left panel) and the surface (right panel) of a sphere. The line defining the meridional plane is shown as a cyan line on the surface. Figure taken from Kloss and Finlay 2019.

2.5.4 Normalization on the CMB

For use in the inversion scheme, toroidal-poloidal expansions of the modes are necessary. The estimation of these involves evaluating and normalizing the horizontal components of the modes on a grid of points on the CMB with locations $\boldsymbol{r} = (1, \theta, \phi)$, where $r = 1$ is the radius of the CMB.

According to K. Zhang and Liao 2017, the mean over the full sphere of the squared absolute values of the geostrophic polynomials is given by

$$\frac{3}{4\pi}\int_{\nu} |G_{2k-1}|^2 \, d\nu = \frac{3\,(2k+1)!!\,(2k-1)!!}{(4k+1)\,(2k)!!\,(2k-2)!!} \tag{2.50}$$

which can be used for normalization. Here,

$$\int_{\nu} d\nu \equiv \int_0^{2\pi} d\phi \int_0^{\pi} sin\theta d\theta \int_0^1 r^2 dr \tag{2.51}$$

The symmetric inertial modes can likewise be normalized with

$$\begin{aligned}\frac{3}{4\pi}\int_{\nu} \left|\boldsymbol{u}^S_{mnk}\right|^2 d\nu = &\sum_{i=0}^{k}\sum_{j=0}^{k-i}\sum_{q=0}^{k}\sum_{l=0}^{k-q} C^S_{mk;ij} C^S_{mk;ql} \frac{3\cdot 2^{m+j+l-3}}{[2\,(m+i+j+q+l)+1]!!}\sigma^{2(i+q)}\left(1-\sigma^2\right)^{j+l} \\ &\cdot \Bigg([(m+m\sigma+2j)\,(m+m\sigma+2l) + (m+m\sigma+2j\sigma)\,(m+m\sigma+2l\sigma)] \\ &\cdot (m+j+l-1)!\frac{[2\,(i+q)-1]!!}{(1-\sigma^2)^2} + 8iq\,(m+j+l)!\frac{[2\,(i+q)-3]!!}{\sigma^2}\Bigg)\end{aligned} \tag{2.52}$$

and the anti-symmetric inertial modes can be normalized with

$$\begin{aligned}\frac{3}{4\pi}\int_{\nu} \left|\boldsymbol{u}^A_{mnk}\right|^2 d\nu = &\sum_{i=0}^{k}\sum_{j=0}^{k-i}\sum_{q=0}^{k}\sum_{l=0}^{k-q} C^A_{mk;ij} C^A_{mk;ql} \frac{3\cdot 2^{m+j+l-3}}{[2\,(m+i+j+q+l)+3]!!}\sigma^{2(i+q)}\left(1-\sigma^2\right)^{j+l} \\ &\cdot \Bigg([(m+m\sigma+2j)\,(m+m\sigma+2l) + (m+m\sigma+2j\sigma)\,(m+m\sigma+2l\sigma)] \\ &\cdot (m+j+l-1)!\frac{[2\,(i+q)+1]!!}{(1-\sigma^2)^2} + 2\,(2i+1)\,(2q+1)\,(m+j+l)! \\ &\cdot \frac{[2\,(i+q)-1]!!}{\sigma^2}\Bigg)\end{aligned} \tag{2.53}$$

2.5.5 Mode Enstrophies

The inversion scheme developed by Kloss and Finlay 2019 also requires explicit expressions of the enstrophy of the inertial modes and the geostrophic polynomials. The general expression for the enstrophy is

$$Q^2\left(\boldsymbol{u}\right)=\frac{3}{4\pi}\int_{\nu}\left|\nabla\times\boldsymbol{u}\right|^2d\nu \tag{2.54}$$

As in Kloss and Finlay 2019, the enstrophy is here defined as Q^2 because we will later use the square root of this quantity, Q, in the inversion scheme. More on this in section 2.11. According to K. Zhang and Liao 2017, the enstrophy of the unnormalized geostrophic polynomials from equation 2.37 can be expressed as

$$Q^2\left(G_{2k-1}\hat{\boldsymbol{\phi}}\right)=3\sum_{jl}^{k}C_{k;j}^{G}C_{k;j}^{G}2^{j+l}jl\frac{[j+l-2]!}{[2\left(j+l\right)-1]!!} \tag{2.55}$$

The enstrophy of the symmetric inertial modes is found by taking the inner product of the vorticity of two unnormalized symmetric inertial modes, $\boldsymbol{u}_{\alpha}^{S}$ and $\boldsymbol{u}_{\beta}^{S}$ from equation 2.41

$$\begin{aligned}
&\frac{3}{4\pi}\int_{\nu}\left(\nabla\times\boldsymbol{u}_{\alpha}^{S}\right)\cdot\left(\nabla\times\boldsymbol{u}_{\beta}^{S}\right)^{*}d\nu=\\
&\sum_{i=0}^{k_\alpha}\sum_{j=0}^{k_\alpha-i}\sum_{q=0}^{k_\beta}\sum_{l=0}^{k_\beta-q}C_{mk_\alpha;ij}^{S}C_{mk_\beta;ql}^{S}\frac{3\cdot2^{j+l+m-1}iq}{[2\left(i+j+q+l+m\right)-1]!!}\\
&\cdot\sigma_\alpha^{2i-2}\sigma_\beta^{2q-2}\left(1-\sigma_\alpha^2\right)^{j-1}\left(1-\sigma_\beta^2\right)^{l-1}\{\sigma_\alpha\sigma_\beta\left[l+j+m-1\right]!\left[2\left(i+q\right)-3\right]!\\
&\cdot\left[\left(2j\sigma_\alpha+m+m\sigma_\alpha\right)\left(2l\sigma_\beta+m+m\sigma_\beta\right)+\left(2j+m+m\sigma_\alpha\right)\left(2l+m+m\sigma_\beta\right)\right]\\
&+2\left(2i-1\right)\left(2q-1\right)\left(1-\sigma_\alpha^2\right)\left(1-\sigma_\beta^2\right)\left[l+j+m\right]!\left[2\left(i+q\right)-5\right]!!\}
\end{aligned} \tag{2.56}$$

where $*$ denotes the complex conjugate. Similarly for unnormalized anti-symmetric inertial modes from equation 2.46

$$\begin{aligned}
&\frac{3}{4\pi}\int_{\nu}\left(\nabla\times\boldsymbol{u}_{\alpha}^{A}\right)\cdot\left(\nabla\times\boldsymbol{u}_{\beta}^{A}\right)^{*}d\nu=\\
&\sum_{i=0}^{k_\alpha}\sum_{j=0}^{k_\alpha-i}\sum_{q=0}^{k_\beta}\sum_{l=0}^{k_\beta-q}C_{mk_\alpha;ij}^{A}C_{mk_\beta;ql}^{A}\frac{3\cdot2^{j+l+m-3}\left(2i+1\right)\left(2q+1\right)}{[2\left(i+j+q+l+m\right)+1]!!}\\
&\cdot\sigma_\alpha^{2i-2}\sigma_\beta^{2q-2}\left(1-\sigma_\alpha^2\right)^{j-1}\left(1-\sigma_\beta^2\right)^{l-1}\\
&\cdot\{\sigma_\alpha\sigma_\beta\left[l+j+m-1\right]!\left[2\left(i+q\right)-1\right]!\\
&\cdot\left[\left(2j\sigma_\alpha+m+m\sigma_\alpha\right)\left(2l\sigma_\beta+m+m\sigma_\beta\right)+\left(2j+m+m\sigma_\alpha\right)\left(2l+m+m\sigma_\beta\right)\right]\\
&+8iq\left(1-\sigma_\alpha^2\right)\left(1-\sigma_\beta^2\right)\left[l+j+m\right]!\left[2\left(i+q\right)-3\right]!!\}
\end{aligned} \tag{2.57}$$

Both equations 2.56 and 2.57 only apply when $m_\alpha=m_\beta=m$. Otherwise, the right-hand sides are zero. As in Kloss and Finlay 2019, for the computation of the regularization norms, for simplicity, we only use terms with $\alpha=\beta$.

2.6 Steady and Time-Dependent Flow Decomposition

In theory, modes can be calculated for infinite variations of the indices m and k. A computable solution to the total inertial flow is constructed as a linear combination of geostrophic and inertial modes with all combinations of the indices m and k up to some level of truncation, M and K. Recall that the index n is already bounded by the finite number of solutions to the polynomials in equations 2.43-2.44 and 2.48-2.49.

This study continues to follow Kloss and Finlay 2019 by decomposing the flow into a steady and a time-dependent part, such that

$$\boldsymbol{u}\left(\boldsymbol{r},t\right) = \boldsymbol{u}_0\left(\boldsymbol{r}\right) + \boldsymbol{u}_t\left(\boldsymbol{r},t\right) \tag{2.58}$$

where the time-dependent flow, $\boldsymbol{u}_t$, is made up of all the geostrophic modes and all inertial modes with $n=1$, i.e. the QG modes and the anti-symmetric modes with the smallest half-frequencies, as seen in equation 2.59. This choice is made because in the core, driving by convection is expected to occur on slow timescales, much longer than the rotation timescale.

$$\begin{aligned}\boldsymbol{u}_t\left(\boldsymbol{r},t\right) = &\sum_{k=1}^{K} a_k^G\left(t\right) G_{2k-1}\left(\boldsymbol{r}\right)\hat{\boldsymbol{\phi}} + \sum_{k=1}^{K}\sum_{m=1}^{M} a_{m1k}^S\left(t\right) Re\left[\boldsymbol{u}_{m1k}^S\left(\boldsymbol{r}\right)\right] \\ &+ \sum_{k=0}^{K}\sum_{m=1}^{M} a_{m1k}^A\left(t\right) Re\left[\boldsymbol{u}_{m1k}^A\left(\boldsymbol{r}\right)\right] + \text{imaginary parts}\end{aligned} \tag{2.59}$$

The steady flow, $\boldsymbol{u}_0$, is then made up of all the remaining inertial modes

$$\boldsymbol{u}_0\left(\boldsymbol{r}\right) = \sum_{k=1}^{K}\sum_{m=0}^{M}\sum_{n\neq 1} a_{mnk}^S Re\left[\boldsymbol{u}_{mnk}^S\left(\boldsymbol{r}\right)\right] + \sum_{k=0}^{K}\sum_{m=0}^{M}\sum_{n\neq 1} a_{mnk}^A Re\left[\boldsymbol{u}_{mnk}^A\left(\boldsymbol{r}\right)\right] + \text{imaginary parts} \tag{2.60}$$

where $Re[\,\cdot\,]$ takes the real part. The mode coefficients, a_{mnk}, determine the amplitude of the mode $\boldsymbol{u}_{mnk}$. It is these coefficients we will eventually solve for in the inverse problem. The imaginary parts in equations 2.59 and 2.60 are necessary because the inertial modes are sinusoidal in azimuth. Each inertial mode is thus expressed with two mode coefficients, in contrast to the geostrophic modes which are expressed with only one.

Although the inertial modes are time-dependent, their time-dependency is not explicitly used to parametrize the time-dependency of the core flow. As in Kloss and Finlay 2019, the time-dependence of the flow is instead parametrized by solving the inverse problem for individual epochs of data (see chapter 3). The combination of the geostrophic and inertial modes at each epoch is used to provide a basis for efficiently representing flow in a rapidly rotating sphere. For all models produced in this study, we use the truncation degrees $K=10$ and $M=20$.

2.7 Toroidal-Poloidal Expansion

Recall that equation 2.13 implied that we can use knowledge of the radial core field and the radial SV to estimate core flow at the CMB. This can be efficiently expressed using a toroidal-poloidal expansion of the core flow and SH expansion of the poloidal and toroidal scalars as well as the magnetic field and SV. The CMB flow is then expressed as

$$\boldsymbol{u}_H = \nabla \times (T\boldsymbol{r}) + \nabla_H (rS) \tag{2.61}$$

where $\boldsymbol{r}$ is a position vector. T and S are then the so-called toroidal and poloidal potentials, respectively, which are expressed with the SH expansions

$$T = \sum_{n=1}^{N_{tp}} \sum_{m=0}^{n} \left(t_n^{mc} cos\,(m\phi) + t_n^{ms} sin\,(m\phi)\right) P_n^m\,(cos\theta) \tag{2.62}$$

$$S = \sum_{n=1}^{N_{tp}} \sum_{m=0}^{n} \left(s_n^{mc} cos\,(m\phi) + s_n^{ms} sin\,(m\phi)\right) P_n^m\,(cos\theta) \tag{2.63}$$

where t_n^{mc}, t_n^{ms}, s_n^{mc}, and s_n^{ms} are SH coefficients for the flow. In this study we use the truncation degree $N_{tp} = 60$ for all models. Applying the steady and time-dependent flow decomposition in equation 2.58, the toroidal-poloidal flow can more accurately be expressed as

$$\boldsymbol{u}_H = [\nabla \times (T_0\boldsymbol{r}) + \nabla_H (rS_0)] + [\nabla \times (T_t\boldsymbol{r}) + \nabla_H (rS_t)] \tag{2.64}$$

In the following we describe the link between this toroidal-poloidal representation and the geostrophic and the inertial modes described previously.

2.8 Forward Problem

The first objective of this study was to produce a model using an inversion scheme that is identical to the one previously used in Kloss and Finlay 2019, but with the newer data set, which we will present in chapter 3. Although this study makes use of newer magnetic SV data from the SWARM mission, the formulation is identical to that of Kloss and Finlay 2019. We will therefore follow their notation closely, and preserve symbolic representations, when possible, for the sake of easy comparison. Later sections will present the modifications that were made in this study to allow for stronger anti-symmetric flows on short length scales and contributions from magnetic diffusion, which were assumed to be zero by Kloss and Finlay 2019.

Assuming the amplitudes of all modes are known for a given epoch, a toroidal-poloidal expansion of the flow on the CMB can be determined for that epoch via a linear set of equations. This is done by arranging the mode amplitudes into two vectors. As described in section 2.6, the time-dependent flow is made up of the geostrophic modes, the QG modes ($n = 1$), and the anti-symmetric modes with the smallest half-frequencies (also $n = 1$). These are represented by the amplitudes a_k^G, a_{m1k}^S, and a_{m1k}^A, as seen in equation 2.59. These amplitudes are thus arranged in the column vector

$$\boldsymbol{a}_p = \left[\boldsymbol{a}^G_{k,p};\ \boldsymbol{a}^S_{m1k,p};\ \boldsymbol{a}^A_{m1k,p}\right] \tag{2.65}$$

with the subscript p denoting the given epoch. The remaining inertial modes make up the steady flow and are arranged in the column vector

$$\boldsymbol{a}_0 = \left[\boldsymbol{a}^S_{mnk};\ \boldsymbol{a}^A_{mnk}\right] \tag{2.66}$$

The number of mode amplitudes stored in these vectors depends on the chosen truncation degrees, K and M, in equations 2.59 and 2.60. The SH coefficients of the toroidal-poloidal expansion at the CMB, listed in $\boldsymbol{x}_p$, can be determined with the linear equation system

$$\boldsymbol{x}_p = \boldsymbol{M}_0\boldsymbol{a}_0 + \boldsymbol{M}_t\boldsymbol{a}_p \tag{2.67}$$

in accordance with equation 2.58. The matrices $\boldsymbol{M}_0$ and $\boldsymbol{M}_t$ calculated by Kloss and Finlay 2019 were again used in this study. They relate the mode amplitudes to the toroidal-poloidal SH coefficients by computing the modes on a Gauss-Legendre grid of evaluation points on the CMB. An SH expansion can then be computed for the radial vorticity and horizontal divergence of the modes, corresponding to the first and second terms of equation 2.61.

Once the toroidal-poloidal expansion of the core flow is known, the SV coefficients can then be computed through

$$\dot{\boldsymbol{b}}_p = \boldsymbol{H}_b\left(\boldsymbol{b}_p\right)\boldsymbol{x}_p + \boldsymbol{e}_p = \boldsymbol{H}_b\left(\boldsymbol{b}_p\right)\left[\boldsymbol{M}_0\boldsymbol{a}_0 + \boldsymbol{M}_t\boldsymbol{a}_p\right] + \boldsymbol{e}_p \tag{2.68}$$

corresponding to equation 2.14. Here the matrix $\boldsymbol{H}_b$ is computed with a MATLAB function 'SV_synthesis' provided by C. Kloss, based on a FORTRAN code written by D. Lloyd in 1987 (Lloyd and Gubbins 1990; Jackson 1997). It contains the frozen flux induction equation (equation 2.10) and takes as input the truncation degrees of the SH expansions of the main field, the SV, and the toroidal-poloidal potentials. These are denoted in equations 2.23, 2.25, and 2.62-2.63 as N_{mf}, N_{sv}, and N_{tp}, respectively. The induction equation also requires knowledge of the main field, $\boldsymbol{b}_p$. 'SV_synthesis' therefore also takes as input the main field gauss coefficients from CHAOS-7.2 corresponding to the time of the given epoch. The vector $\boldsymbol{e}_p$ contains the unknown small-scale error coefficients which are co-estimated in the inversion, along with the mode amplitudes.

SV data predictions can be made at the locations corresponding to the ground observatories and virtual observatories used in this study through

$$\boldsymbol{d}_p = \boldsymbol{Y}_p\dot{\boldsymbol{b}}_p = \boldsymbol{Y}_p\boldsymbol{H}_b\left(\boldsymbol{b}_p\right)\left[\boldsymbol{M}_0\boldsymbol{a}_0 + \boldsymbol{M}_t\boldsymbol{a}_p\right] + \boldsymbol{Y}_p\boldsymbol{e}_p \tag{2.69}$$

where $\boldsymbol{d}_p$ contains SV predictions for all three vector components at each observatory location. The matrix $\boldsymbol{Y}_p$ is computed with another MATLAB function, 'design_SHA', which is written by N. Olsen in 2003. The function takes as input the (r,θ,ϕ) coordinates of each observatory and the truncation degree of the SV. It contains the details of the potential field representation of the internal field (equations 2.21-2.25). The output matrix then contains the appropriate spatial derivatives. At the CMB, SV is generally most prominent at lower latitudes (e.g. Aubert et al. 2013). Furthermore, the data is more contaminated by noise due to ionospheric sources at high latitudes (e.g. Kloss and Finlay 2019). We therefore exclusively use data from observatories outside the Tangent Cylinder (TC) of the core, i.e. observatories with co-latitudes $30° < \theta < 150°$ and focus our attention on low latitudes.

In the inversion scheme we solve for the mode amplitudes and small-scale error at all epochs simultaneously. SV predictions for the entire data period are thus found with

$$\boldsymbol{d} = \boldsymbol{G}\boldsymbol{m} \tag{2.70}$$

where the model vector (column) initially consists of the mode amplitudes and the small-scale error

$$\boldsymbol{m} = [\boldsymbol{a}_0;\ \boldsymbol{a}_1;\ \boldsymbol{a}_2;\ \cdots;\ \boldsymbol{a}_P;\ \boldsymbol{e}_1;\ \boldsymbol{e}_2;\ \cdots;\ \boldsymbol{e}_P] \tag{2.71}$$

with the subscript P denoting the final epoch. The associated design matrix for the forward problem is constructed as

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{F}_1\boldsymbol{M}_0 & \boldsymbol{F}_1\boldsymbol{M}_t & & & \boldsymbol{Y}_1 & & \\ \vdots & & \ddots & & & \ddots & \\ \boldsymbol{F}_P\boldsymbol{M}_0 & & & \boldsymbol{F}_P\boldsymbol{M}_t & & & \boldsymbol{Y}_P \end{bmatrix} \tag{2.72}$$

where $\boldsymbol{F}_p = \boldsymbol{Y}_p\boldsymbol{H}_b\left(\boldsymbol{b}_p\right)$. The matrices $\boldsymbol{Y}_p$ are adjusted for each epoch, such that predictions are only made at observatories that provided data for that epoch.

2.9 Inversion

As in Kloss and Finlay 2019, the first model is produced by minimizing the cost function

$$\begin{aligned}\boldsymbol{\Phi}\left(\boldsymbol{m}\right) = \left(\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}^{obs}\right)^T \boldsymbol{W}_d \left(\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}^{obs}\right) + \lambda_0||\boldsymbol{Q}_0\boldsymbol{a}_0||_1 + \lambda_t^S||\boldsymbol{Q}_t^S\boldsymbol{a}_t||_1 \\ + \lambda_t^A||\boldsymbol{Q}_t^A\boldsymbol{a}_t||_1 + \lambda_a||\boldsymbol{D}\boldsymbol{a}_t||_2^2 + \boldsymbol{e}_t^T\boldsymbol{C}_e^{-1}\boldsymbol{e}_t\end{aligned} \tag{2.73}$$

where $||\cdot||_p$ takes the l_p norm. The $\boldsymbol{a}_t$ and $\boldsymbol{e}_t$ column vectors respectively contain all the $\boldsymbol{a}_p$ and $\boldsymbol{e}_p$ vectors in chronological order. The diagonal matrices $\boldsymbol{Q}$ hold the square roots of the mode enstrophies, described in section 2.5.5, corresponding to the order of the mode amplitudes in the vectors $\boldsymbol{a}$. The spatial structure of the first flow model is thus regularized with an l_1 norm of all the mode enstrophies. The degree to which the steady, equatorially symmetric, and equatorially anti-symmetric mode enstrophies are penalized can then be adjusted with the regularization parameters λ_0, λ_t^S, and λ_t^A, respectively.

The inversion also includes an l_2 norm temporal regularization on the first difference of the flow using the matrix

$$\boldsymbol{D} = \frac{1}{\Delta t} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \tag{2.74}$$

with the time step $\Delta t = t_{p+1} - t_p$ taking the first difference, i.e. the difference between a given epoch and the previous one. The matrices $\boldsymbol{I}$ are then identity matrices, with dimensions compatible with $\boldsymbol{a}_p$. The degree to which changes to the flow are penalized in equation 2.73 is then adjusted with the regularization parameter λ_a.

The first term on the right-hand side of equation 2.73 regularizes the misfit between the model predictions and actual observations of SV. This is done with the diagonal matrix $\boldsymbol{W}_d$ containing the data error variances which are modified with a Tukey biweight scheme. The last term in equation 2.73 regularizes the small-scale error, by computing the small-scale error covariance matrix, $\boldsymbol{C}_e$, which depends on the model vector $\boldsymbol{m}$. Neither the regularization of data residuals nor the regularization of the small-scale error are subject to change in this study. The reader is thus referred to Kloss and Finlay 2019 for full details on the construction of $\boldsymbol{W}_d$ and $\boldsymbol{C}_e$.

The non-linear inverse problem is solved for $\boldsymbol{m}$, iteratively, with

$$\boldsymbol{m}_{k+1} = \left(\boldsymbol{G}^T \boldsymbol{W}_d \boldsymbol{G} + \boldsymbol{R}(\boldsymbol{m}_k)\right)^{-1} \boldsymbol{G}^T \boldsymbol{W}_d \boldsymbol{d}^{obs} \tag{2.75}$$

The regularization matrix, which depends on the model from the previous iteration, is constructed as

$$\boldsymbol{R} = \begin{bmatrix} \lambda_0 \boldsymbol{W}_0 & & \\ & \lambda_t^S \boldsymbol{W}_t^S + \lambda_t^A \boldsymbol{W}_t^A + \lambda_a \boldsymbol{W}_a & \\ & & \boldsymbol{C}_e^{-1} \end{bmatrix} \tag{2.76}$$

where the matrices $\boldsymbol{W_0}$, $\boldsymbol{W}_t^S$, and $\boldsymbol{W}_t^A$ implement the l_1 norms in equation 2.73 and are constructed as

$$\boldsymbol{W}_0 = \boldsymbol{Q}_0^T \left(\frac{\delta_{ij}}{\sqrt{(\boldsymbol{Q}_0 \boldsymbol{a}_0)_i^2 + \epsilon^2}} \right) \boldsymbol{Q}_0 \tag{2.77}$$

and similarly for $\boldsymbol{W}_t^S$ and $\boldsymbol{W}_t^A$. Here δ_{ij} is the Kronecker delta, with i and j being row and column indices. $\epsilon = 10^{-8}$ is a constant preventing computational issues, should the denominator approach zero.

Finally, the temporal l_2 norm is implemented with

$$\boldsymbol{W}_a = \boldsymbol{D}^T \boldsymbol{D} \tag{2.78}$$

Convergence of the inversion is considered complete when the relative change of each norm listed in equation 2.73 is below 0.01 for subsequent iterations.

2.10 Relaxing Penalization of Equatorially Anti-Symmetric Flow

The method of Kloss and Finlay 2019 is next extended to produce a model that allows for more power at short length-scales for the anti-symmetric flow. One purpose of this is to investigate whether a higher tolerance for localized equator crossings of the flow can still explain the data well. In order to do this, we replace the matrix penalizing the enstrophy of the anti-symmetric modes, $\boldsymbol{Q}_t^A$, with a matrix that instead penalizes the mode energy

$$E = \frac{3}{4\pi} \int_{\nu} |\boldsymbol{u}|^2 \, d\nu \tag{2.79}$$

Comparing to equation 2.53, this quantity is equal to the squares of the amplitudes of the normalized equatorially anti-symmetric modes themselves. We thus simply replace the previously defined matrix $\boldsymbol{Q}_t^A$ with

$$\boldsymbol{Q}_t^A \equiv \boldsymbol{I} \tag{2.80}$$

where $\boldsymbol{I}$ is the identity matrix with dimensions corresponding to the length of $\boldsymbol{a}_t$. The matrix implementing the new regularization norm for anti-symmetric modes then becomes

$$\boldsymbol{W}_t^A = \boldsymbol{I}^T \left(\frac{\delta_{ij}}{\sqrt{(\boldsymbol{I}\boldsymbol{a}_t)_i^2 + \epsilon^2}} \right) \boldsymbol{I} \tag{2.81}$$

which is then used in the regularization matrix (equation 2.76). This way it is possible to penalize the anti-symmetric modes with the regularization parameter λ_t^A and not penalize the vorticity of those modes. This further allows for more power at smaller wavelengths and localized equatorial crossings of the flow when required by observations.

2.11 Accounting for Magnetic Field Diffusion at the CMB

Another objective of this study was to develop a method of accounting for any diffusion that may have influenced the SV observed by SWARM and ground observatories. To this end, SH coefficients describing the SV due to diffusion from the Coupled-Earth (CE) numerical dynamo model (Aubert et al. 2013) are used. These coefficients, henceforth referred to as the *dynamo realizations*, were obtained from "snapshots" of dynamo simulations of core flow.

Magnetic field measurements in a dynamo simulation can be used to obtain SH coefficients of the magnetic field at the boundary (corresponding to the CMB), r_{cmb}, and just below the boundary at r_{low}. There is thus a depth difference $\partial_r = r_{cmb} - r_{low}$. For an SH degree n of the magnetic field, the recurrence relations (Schaeffer 2015)

$$\partial_r \boldsymbol{B}_{cmb} = \frac{-(n+1)\,\boldsymbol{B}_{cmb}}{r} \quad \text{and} \quad \nabla_\perp^2 \boldsymbol{B} = \frac{-n\,(n+1)\,\boldsymbol{B}}{r^2} \tag{2.82}$$

can then be used in a second order Taylor expansion to compute $\nabla^2 \boldsymbol{B}_r$. Multiplying with a known magnetic diffusivity, the diffusion part of the induction equation is obtained (equation 2.8) for the radial part of the field. It is thus possible to compute the dynamo realizations explaining the radial field SV from magnetic diffusion.

In this study, a self-covariance matrix of 1505 dynamo realizations from the CE Model was computed for use in our inversion scheme. In figure 2.2 predictions of SV at the CMB are shown for three of these realizations. The realizations, listed in vectors $\boldsymbol{z}$, were first truncated to match the truncation degree used for induction SV in previous inversion schemes, N_{sv}. A vector representing the "background state", $\hat{\boldsymbol{z}}$, was then computed as the mean of all the 1505 truncated realizations. The self-covariance matrix of the realizations, $\boldsymbol{C}_z$, was then computed as

$$\boldsymbol{C}_z = (\boldsymbol{z} - \hat{\boldsymbol{z}})(\boldsymbol{z} - \hat{\boldsymbol{z}})^T \tag{2.83}$$

A bulk diagonal matrix, $\boldsymbol{C}_Z^{-1}$, was then built containing one inverse of these self-covariance matrices for every epoch

$$\boldsymbol{C}_Z^{-1} = \begin{bmatrix} \boldsymbol{C}_z^{-1} & & \\ & \ddots & \\ & & \boldsymbol{C}_z^{-1} \end{bmatrix} \tag{2.84}$$

This matrix was then used to augment the regularization matrix (equation 2.76) such that

$$\boldsymbol{R} = \begin{bmatrix} \lambda_0 \boldsymbol{W}_0 & & & \\ & \lambda_t^S \boldsymbol{W}_t^S + \lambda_t^A + \boldsymbol{W}_t^A + \lambda_a \boldsymbol{W}_a & & \\ & & \boldsymbol{C}_e^{-1} & \\ & & & \boldsymbol{C}_Z^{-1} + \lambda_z \boldsymbol{W}_a \end{bmatrix} \tag{2.85}$$

Here we also apply a temporal regularization to the diffusion, similar to that of the time-dependent modes (i.e. l_2 norm of the first difference), by adding another $\boldsymbol{W}_a$ matrix with dimensions equal to those of $\boldsymbol{C}_Z^{-1}$. It is then multiplied with a new regularization parameter, λ_z, to allow for penalization of secular acceleration (i.e. changes to the SV) caused by diffusion. The model vector produced with equation 2.75 then becomes

$$\boldsymbol{m} = [\boldsymbol{a}_0;\ \boldsymbol{a}_1;\ \boldsymbol{a}_2;\ \cdots;\ \boldsymbol{a}_P;\ \boldsymbol{e}_1;\ \boldsymbol{e}_2;\ \cdots;\ \boldsymbol{e}_P;\ \boldsymbol{z}_1;\ \boldsymbol{z}_2;\ \cdots;\ \boldsymbol{z}_P] \tag{2.86}$$

with the vectors $\boldsymbol{z}_p$ containing SH coefficients for SV due to diffusion in the p'th epoch. The design matrix $\boldsymbol{G}$ of equation 2.75 then requires an augmentation, to produce data predictions for the SV at ground and satellite observatories. This is done with an augmentation similar to that employed for the small-scale error (since the model parameters for both the small-scale error and the diffusion take the form of SH coefficients), such that

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{F}_1 \boldsymbol{M}_0 & \boldsymbol{F}_1 \boldsymbol{M}_t & & \boldsymbol{Y}_1 & & \boldsymbol{Y}_1 & \\ \vdots & & \ddots & & \ddots & & \ddots \\ \boldsymbol{F}_P \boldsymbol{M}_0 & & \boldsymbol{F}_P \boldsymbol{M}_t & & \boldsymbol{Y}_P & & \boldsymbol{Y}_P \end{bmatrix} \tag{2.87}$$

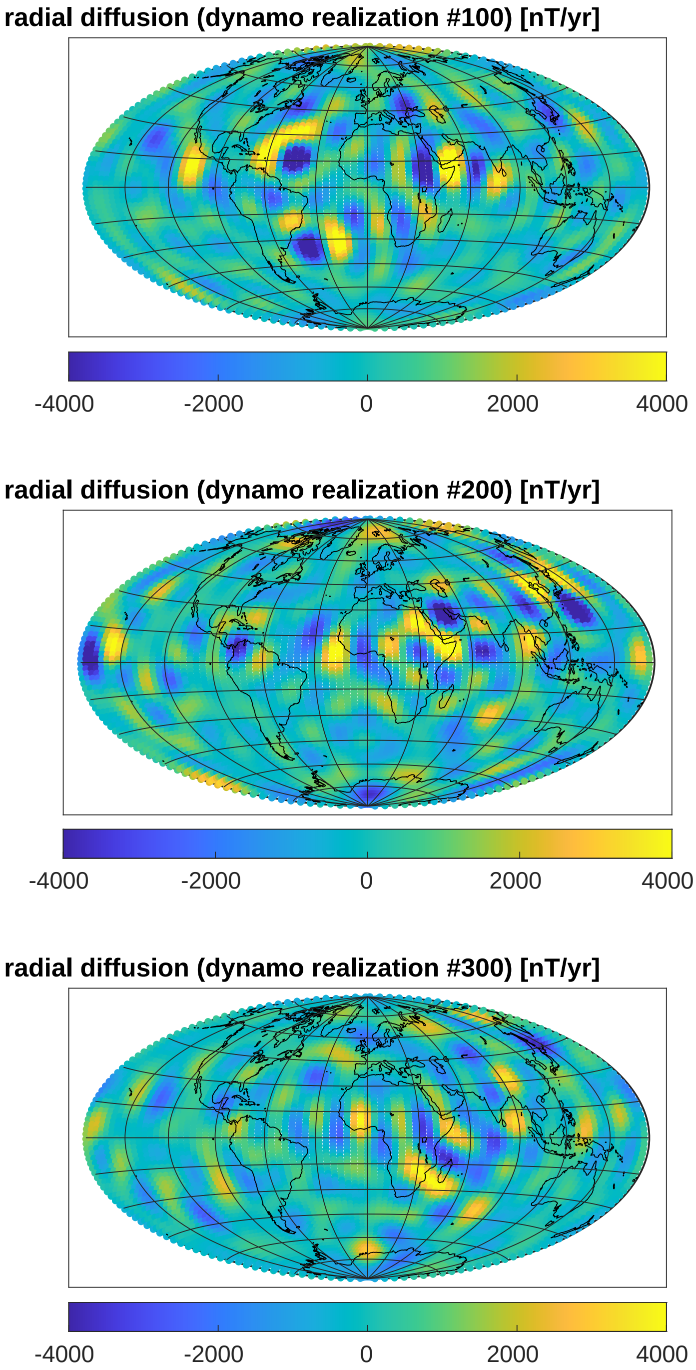


Figure 2.2: SV predictions at the CMB from three example realizations of the Coupled Earth Model.

3 Observations

This chapter presents the initial input data and details on how it was ingested for use in this study, as well as sections on data inspection prior to inversion. All input SV data is taken from the period September 2014 to September 2019, covering all available SWARM data at the time of this writing. The SV data is given in epochs covering an average period of 4 months. The SWARM period thus contains 16 epochs of data. The SV is defined by taking the yearly differences of the measured main field, i.e. the difference in main field strength for a given epoch plus/minus 6 months.

3.1 Data Types

3.1.1 Ground Observatories

Geomagnetic SV data from 205 INTERMAGNET ground observatories (GOs) were used as input. A detailed explanation on how the GO data set was produced and treated, prior to use in this study, is presented in Olsen et al. 2014. It is based on hourly means after corrections for external and ionospheric field and robust means. The data set contains SV data for three vector components for each observatory. The input file initially contained data covering Jan 1900 to May 2020 but data outside the SWARM period were removed for this study. Data from observatories inside the TC (colatitudes below 30 degrees and above 150 degrees) were also removed, resulting in a data set with data from 117 GOs. Not all of these observatories have provided data for every epoch in the SWARM period, since reporting from some of the observatories is delayed. This characteristic delay (Matzka et al. 2010) results in a general decline in the total number of GOs with available data from 2014 to 2019. The data set also includes sigmas (error estimates) for each ground observatory on each vector component. The sigmas are based on the variance of each series (each component at each observatory) with respect to the CHAOS field model.

Figure 3.1 shows an example of a 3-axis fluxgate magnetometer used for measurements at GOs (left) and the locations of GOs used in this study (right).

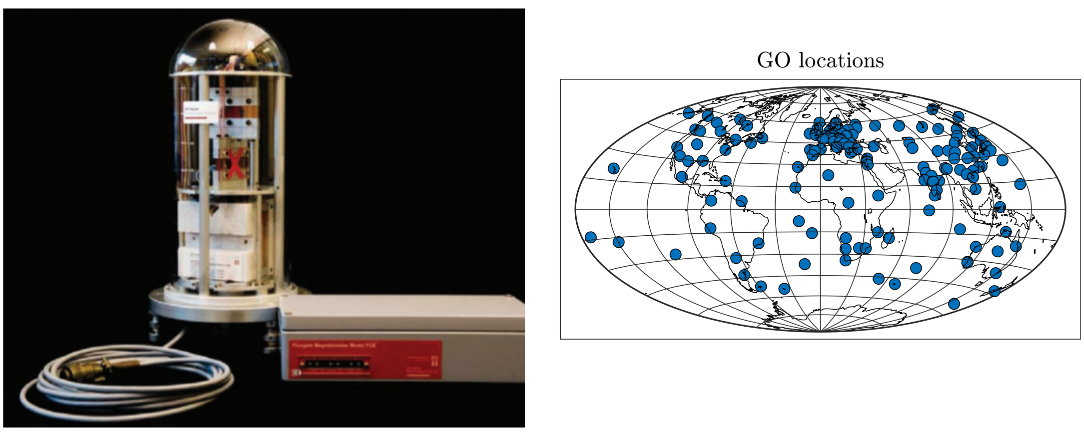


Figure 3.1: 3-axis fluxgate magnetometer (FGE model) developed by DTU Space (left) and locations of GOs used in this study (right).

The fluxgate magnetometers are calibrated with weekly absolute measurements of declination, inclination, and intensity of the geomagnetic field. In figure 3.2 we show SV measured throughout the data period at three example GOs in Hawaii (HON), French Guinea (KOU), and Japan (KAK) with associated error estimates as well as predictions by the CHAOS-7.2 field model. Timeseries are shown for each vector component, i.e. r (radial), theta (meridional), and phi (azimuthal).

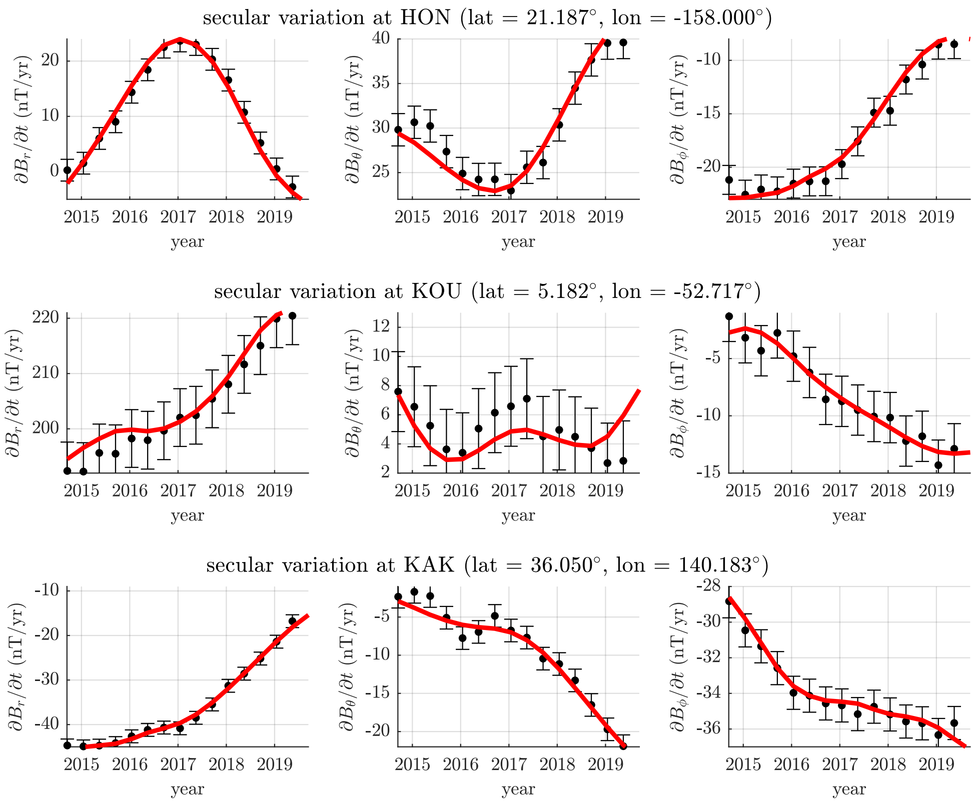


Figure 3.2: Example timeseries of SV data from three GOs (HON, KOU, and KAK) for all three vector components with associated error estimates. CHAOS-7.2 model predictions are also shown (red).

3.1.2 SWARM Satellite Data and Derived Virtual Observatory Series

Satellite altitude geomagnetic SV data were taken from the Geomagnetic Virtual Observatory (GVO) series, presented in Hammer et al. 2021. The GVO series is constructed using magnetic field measurements from the SWARM satellite trio (Alpha, Bravo, Charlie). In contrast to raw measurements from the constantly moving satellites, these series provide a convenient way of monitoring SV by constructing spatially fixed virtual observatories (VOs). The series thus contains data from 300 equal area distributed VOs, all with 490 km altitude. The series used in this study is obtained by robust fits of local Cartesian potential field models to along-track and east-west sums and differences of SWARM satellite data collected within a radius of 700 km of the VO locations and based on four monthly time windows. The basic concept is illustrated in figure 3.3.

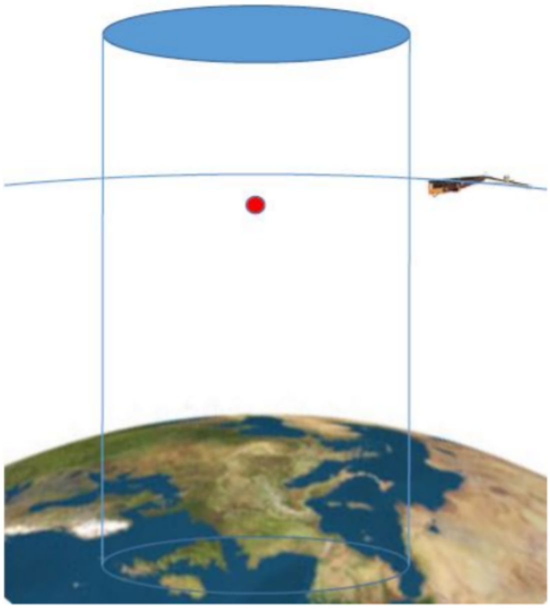


Figure 3.3: Illustration of the Geomagnetic Virtual Observatory concept. Satellite magnetic measurements from within a target cylinder are used to infer field time series at the VO location given by the red dot. Note the cylinder radius is not to scale. Figure taken from Hammer et al. 2021.

As with GO data, all VO data inside the core's TC were also removed in this study, resulting in a data set with 258 VOs. The data set includes sigmas for each VO on each vector component. These are again based on the variance of each series with respect to the CHAOS field model.

The 3-axis vector magnetometer carried by the SWARM satellites is shown in figure 3.4 (left) along with the star trackers used to determine orientation. In the figure, we also show the locations of VOs used in this study (right).

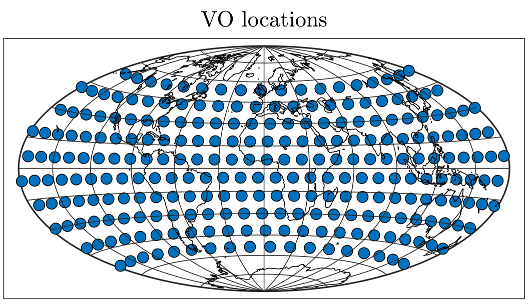


Figure 3.4: SWARM 3-axis vector magnetometer with star trackers, mounted on an optical bench (left) and locations of VOs used in this study (right).

The SWARM satellites' vector mangetometers are calibrated by making high frequency measurements of magnetic field intensity in many different orientations to calculate calibration parameters. In figure 3.5 we show SV measured throughout the data period at three example VOs with associated error estimates as well as predictions by the CHAOS-7.2 field model.

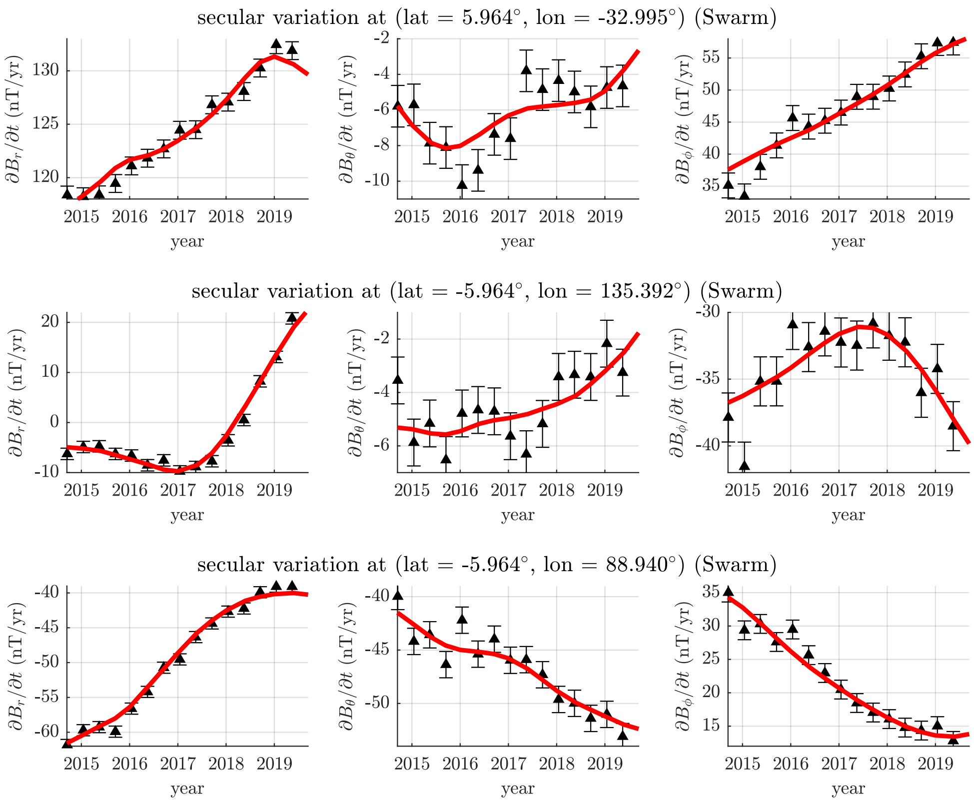


Figure 3.5: Example timeseries of SV data from three VOs for all three vector components with associated error estimates. CHAOS-7.2 model predictions are also shown (red).

3.2 Temporal Data Distribution

The figures presented in this section were produced to ensure that SV data were properly ingested before use in the inversion and show their temporal distribution throughout the data period. Figure 3.6 shows the total number of data vs. time, after removal of NaNs and data inside the TC. As expected, the number of VO data points remains constant, while the number of GO data points generally drops with time, due to delays in reporting.

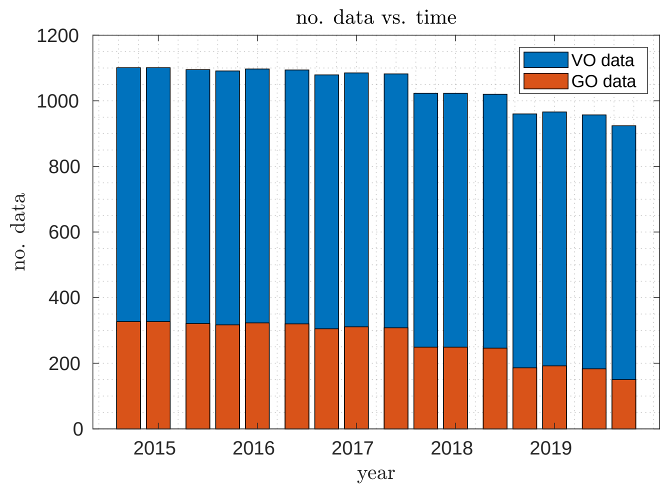


Figure 3.6: Total number of data points vs. time. Each column represents an epoch.

Figure 3.7 correspondingly shows the number of data points in each vector component vs. time. As expected, the number of data points for each vector component is equal for most epochs, although six subsequent epochs (September 2015 to June 2016) of GO data (right) contain one less data point for the phi component. Investigation of the data revealed that the missing data point is linked to a single GO in Western Samoa, codenamed API.

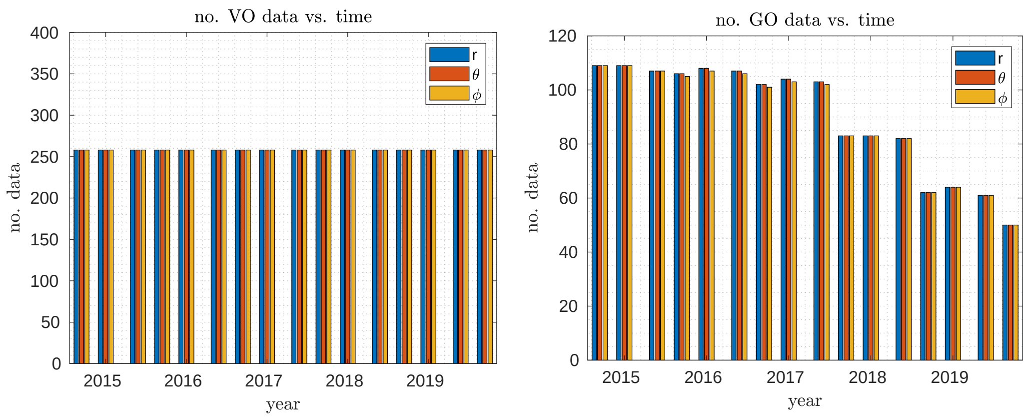


Figure 3.7: Number of data points for each vector component vs. time. Plots are shown for VOs (left) and GOs (right). Each trio of columns (blue, red, yellow) represents an epoch.

3.3 Data Error Estimates

The data error estimates used in the inversion were also plotted to document their distribution. Figure 3.8 shows the sigmas in histograms, while figure 3.9 plots the sigma values on a world map. Both figures reveal that GO data errors are on average larger than VO errors. This can partly be attributed to some of the ground observatories having been in operation since 1900 and thus having made use of less modern instruments. Figure 3.8 also shows that, among VO data (left), the average error on the phi component (bottom) is significantly larger than the other vector components.

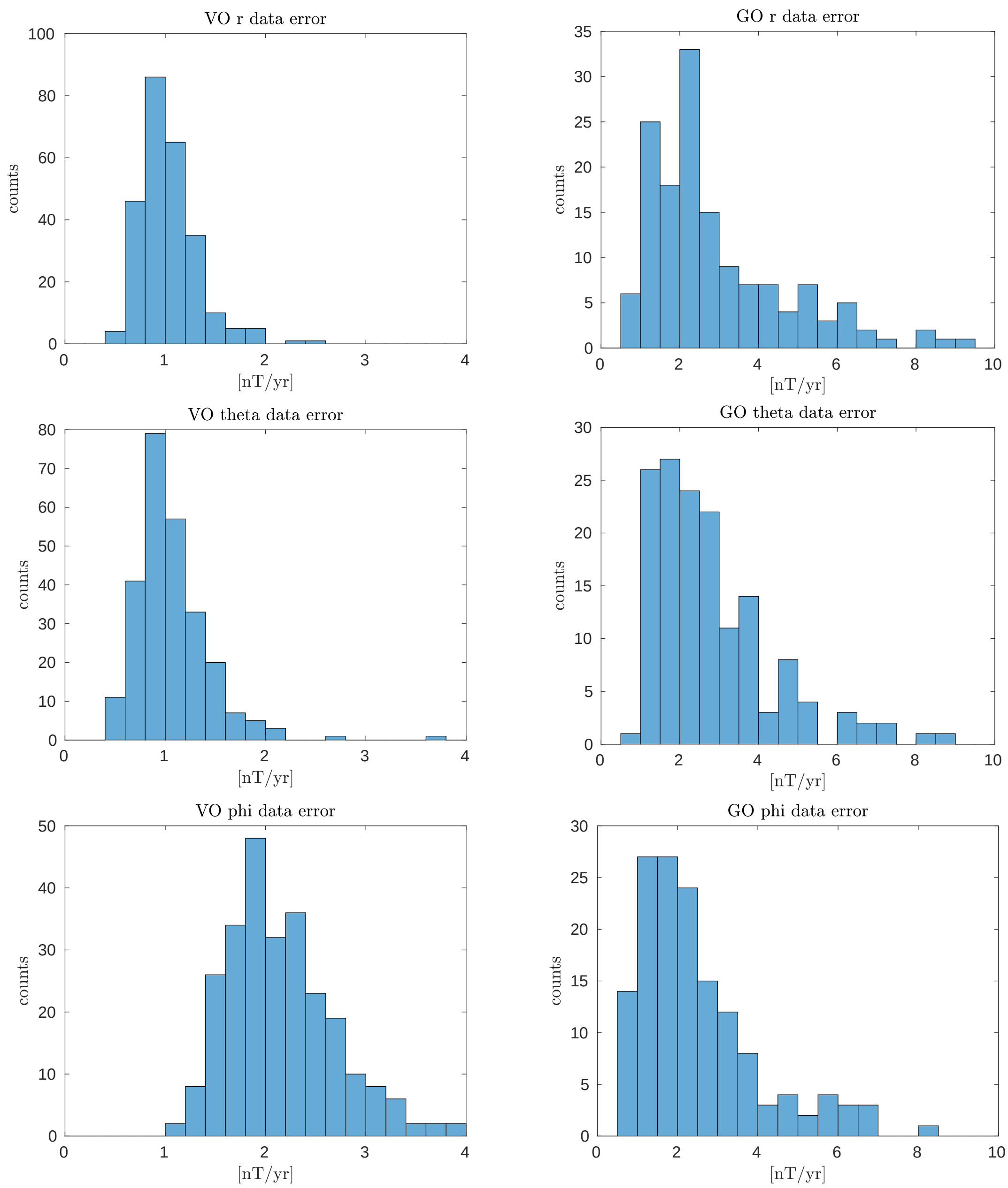


Figure 3.8: Histograms of sigma values for VOs (left) and GOs (right) shown for each vector component, r (top), theta (middle), phi (bottom).

Figure 3.9 shows that these relatively large errors in the VO phi data (bottom left) are somewhat evenly distributed across the VOs, but are largest at the highest northern latitude and above the North- and South American continents. The figure further shows, that the r and theta components of the VO data generally have higher error estimates at the southern-most latitude. The high error estimates on GO data can generally be attributed to individual GOs, such as VSS in Brazil.

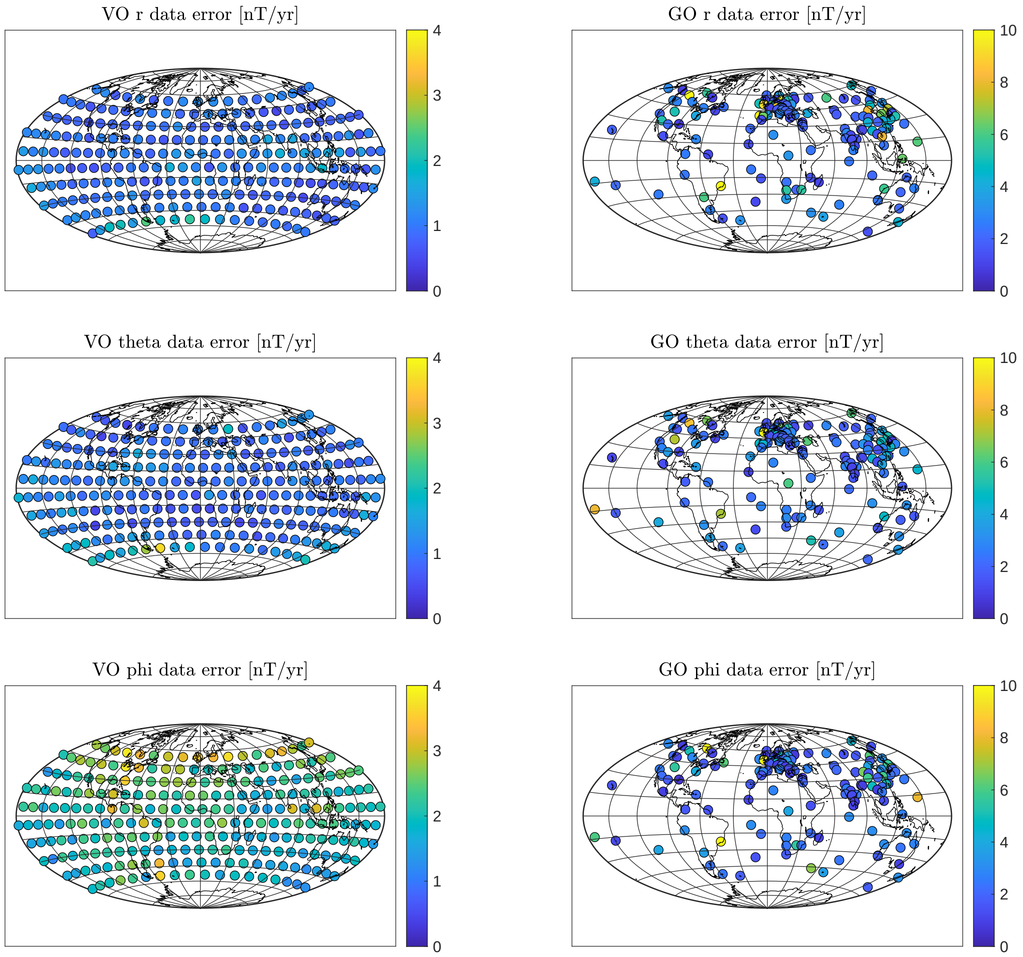


Figure 3.9: Sigma values on world map for VOs (left) and GOs (right) shown for each vector component, r (top), theta (middle), and phi (bottom).

Also note that figure 3.5 shows a CHAOS-7.2 misfit that is consistently greater than the error bars of the phi component for the epochs centered on January 2015 and January 2016. Further inspection revealed (see chapter 5) that most of the difference between VO data error in the phi component and the other vector components can be attributed to these two epochs. This feature is also found in the original SWARM data, which the CHAOS-7.2 model does not fit well for the magnetic field's phi component in summer 2015. This may indicate significant inter-hemispheric field aligned currents during these months which are currently under investigation (C. Finlay, pers. comm.).

4 Results

This chapter presents results, including diagnostic and visualization plots for the core flow models produced in this thesis, using the methods described in chapter 2. For presentation purposes, we collect the models into three ”generations”, with generation 1 being a model produced using the same regularization scheme as Kloss and Finlay 2019 (as described in section 4.1), generation 2 being the models incorporating the relaxed penalization of equatorially anti-symmetric modes (section 2.10), and generation 3 being the models that, along with the modification of generation 2, also take diffusion into account (section 2.11). All model generations presented are derived from the new GO and SWARM VO data sets, presented in chapter 3. Results for the preferred model in each generation (Model 1, Model 2a, and Model 3c) are presented in the following sections. The reasoning behind the choice of preferred models is discussed in chapter 5.

Table 4.1 lists all the models along with values of the regularization parameters used in the inversion. The table also presents the values of some norms used for diagnostics. The first norm, Φ_{sv}, is the normalized misfit between observed SV data and the SV data predicted by the models

$$\Phi_{sv} = \frac{1}{N_d}\left(\boldsymbol{d} - \boldsymbol{d}^{obs}\right)^T \boldsymbol{W}_d \left(\boldsymbol{d} - \boldsymbol{d}^{obs}\right) \tag{4.1}$$

where $\boldsymbol{W}_d$ is the matrix from equation 2.73 containing the estimated data variances, modified with a Tukey biweight scheme. The geomagnetic field model CHAOS-7.2 used for predictions of the magnetic field and SV at the CMB produces a normalized misfit of 0.6742 (this suggests that the adopted error estimates may have been too conservative). This misfit was used as a benchmark aimed for in the other models produced in this study, i.e. we derived flow models that predict the SV to the same level as the well established CHAOS field model. Regularization parameters were thus adjusted until a similar misfit was achieved (see Table 4.1). Another norm used for diagnostics, f_S, is a measure of the relative power of the geostrophic and equatorially symmetric modes (the equatorially symmetric part of the flow) versus the equatorially anti-symmetric modes (the equatorially anti-symmetric part of the flow), integrated over the core surface. Both the steady and the time-dependent parts of the flow are included.

Model	λ_0	λ_t^S	λ_t^A	λ_a	λ_z	Φ_{sv}	f$_S$	f$_t$
CHAOS-7.2	-	-	-	-	-	0.6742	-	-
1	6.3	0.76	3.8	1.5$\cdot 10^3$	-	0.6956	0.6963	0.0569
2a	6.3	0.95	47	1.5$\cdot 10^3$	-	0.6828	0.6738	0.0697
2b	6.3	1.5	30	1.5$\cdot 10^3$	-	0.6806	0.5550	0.0635
2c	6.3	3.8	9.1	1.5$\cdot 10^3$	-	0.6817	0.4861	0.1059
2d	6.3	7.6	6.8	1.5$\cdot 10^3$	-	0.6818	0.4509	0.1540
3a	44	6.7	3.3$\cdot 10^2$	1.1$\cdot 10^4$	0	0.6822	0.5349	2.545$\cdot 10^{-5}$
3b	6.3	0.95	47	1.5$\cdot 10^3$	7.5$\cdot 10^4$	0.6792	0.6873	0.1016
3c	9.5	1.4	60	2.3$\cdot 10^3$	1.1$\cdot 10^4$	0.6823	0.6394	0.1072

Table 4.1

$$f_S = \frac{\left\langle \int_\Omega \left|\boldsymbol{u}^S\right|^2 d\Omega \right\rangle}{\left\langle \int_\Omega \left|\boldsymbol{u}\right|^2 d\Omega \right\rangle} \tag{4.2}$$

where $\langle \cdot \rangle$ takes the time average, $\boldsymbol{u}^S$ denotes the equatorially symmetric flow and $\boldsymbol{u}$ denotes all flow. This is a useful tool for measuring the influence of varying the regularization parameters, in particular those penalizing symmetric and anti-symmetric flow (λ_t^S and λ_t^A). A similar norm is used to measure the relative power of the time-dependent flow versus the steady flow.

$$f_t = \frac{\left\langle \int_\Omega \left|\boldsymbol{u}_t\right|^2 d\Omega \right\rangle}{\left\langle \int_\Omega \left|\boldsymbol{u}\right|^2 d\Omega \right\rangle} \tag{4.3}$$

which is particularly useful for measuring the influence of varying the penalization of temporal flow changes (λ_a).

4.1 Model 1: Previous Regularization Scheme

This section presents results for Model 1, which is produced by using the previous inversion scheme of Kloss and Finlay 2019, but with the new data presented in chapter 3. The purpose of producing this model was, in part, to test the scheme's ability to produce reasonable models with newer data, before modifications were applied to the regularization. Another purpose was to find reasonable values for the regularization parameters, which were then used as a starting point for determining the regularization parameters of later generation models. Lastly, this previous regularization scheme model was useful as a control model for evaluating the effects of the modifications that were later applied.

Model 1 has a time-dependent part consisting of a single geostrophic mode of 10 geostrophic polynomials, 200 QG modes, and 220 equatorially anti-symmetric modes with $n = 1$, as well as 288 SH coefficients describing the SV due to the small-scale error. These numbers arise from the truncation degrees $K = 10$ and $M = 20$ in equation 2.59 and $N_{sv} = 16$ in equation 2.25. Recall that the inertial modes are represented by two mode amplitudes. For the 16 epochs of data, the total number of parameters estimated for the time-dependent part of the model is therefore $16 \cdot (10 + 2 \cdot 200 + 2 \cdot 220 + 288) = 18208$.

The steady part of the model is represented by 2045 equatorially symmetric modes and 2255 equatorially anti-symmetric modes. These numbers arise from the truncation degrees $K = 10$ and $M = 20$ in equation 2.60, along with the finite number of solutions to the polynomials in equations 2.43-2.44 and 2.48-2.49. The total number of parameters estimated for the steady part of the model is therefore $2 \cdot 2045 + 2 \cdot 2255 = 8600$. The full model vector (equation 2.71) then contains $18208 + 8600 = 26808$ parameters. By comparison, the full data vector contains 16698 SV data points, meaning we are dealing with an under-determined inverse problem. The regularization scheme is therefore crucial to the obtained results.

4.1.1 Estimated Flow

Figure 4.1 shows the total flow (steady + time-dependent) at the CMB, averaged over all 16 epochs. This time-averaged flow is very similar to that produced by Kloss and Finlay

2019, although their model covered the period from September 2000 to January 2018. One characteristic of the flow is a large anti-cyclonic gyre in the northern hemisphere. The gyre flows south at around $100°E$ before turning into a strong equatorial westward flow under Africa and the Atlantic Ocean and finally turning back north under the north-western Atlantic. The gyre is somewhat mirrored in the southern hemisphere, where the flow is northward at around $100°E$ before turning west at the equator and finally turning south under South America. Overall, the flow is fastest south of the equator, under the Atlantic. Our time-averaged flow deviates somewhat from that produced by Kloss and Finlay 2019, as it has a clear eastward flow under the Pacific Ocean. The time-averaged flow is mostly symmetrical with about 70% of the flow power being produced by symmetric flow. It is also mostly steady with only about 6% of the power being produced by the time-dependent part of the flow (see f_S and f_t in table 4.1).

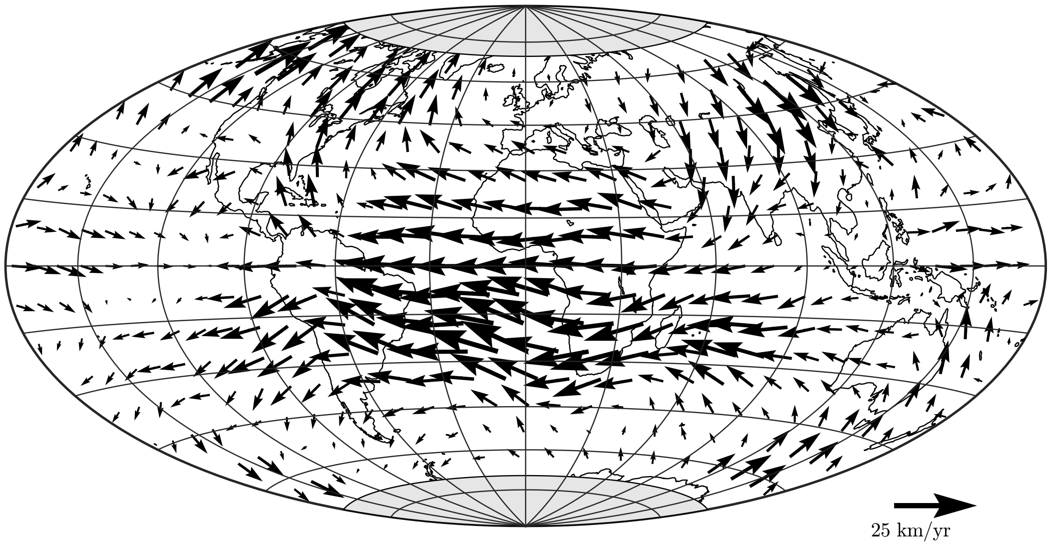


Figure 4.1: Model 1: Time-average of the total flow on the CMB.

The time-dependent part of the flow is shown for every September of the data period in figure 4.2. It is characterized by relatively strong flows in the beginning and towards the end of the data period, with a relatively quiet period in between. Overall, the time-dependent flow is strongest at lower latitudes with alternating directions along the equator, seperated by regions of diverging or converging flow, indicating up- and downwelling of the core fluid. This low-latitude flow is mainly azimuthal, although a slight equator crossing is seen under the Pacific Ocean from the first epoch (September 2014) until 2018. The direction of the time-dependent flow also seems to alternate with time. For example we see a reversal of flow from westward to eastward under the Eastern Pacific and Indian Ocean and vice versa under the Atlantic. Under the western Pacific, flow is initially quite slow, but has, in 2019, evolved into a strong eastward flow.

Figure 4.3 shows power spectra of the time-averaged total flow, as well as snapshots of the time-dependent flow. These power spectra are very consistent with the model produced by Kloss and Finlay 2019 with the flow gradually losing power with increasing

SH degree up to degree 15, which is a consequence of the imposed enstrophy-based regularization. This indicates that large-scale flows are dominant, while small-scale flows are increasingly insignificant. Especially after SH degree 15, where the flow starts to rapidly lose all of its power. The time-dependent flow alone is 92% symmetric.

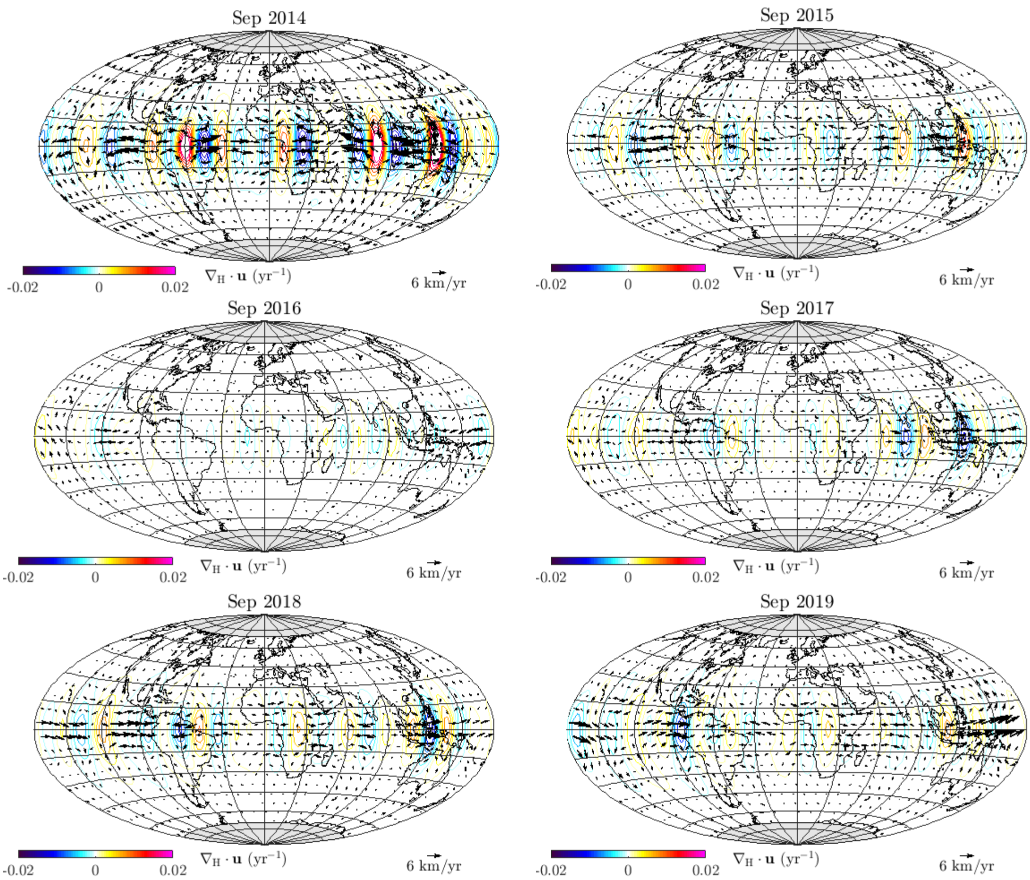


Figure 4.2: Model 1: Time-dependent flow at the CMB for September of every year in the data period. Arrows represent flow and contours represent divergence/convergence, corresponding to upwelling/downwelling of the core fluid.

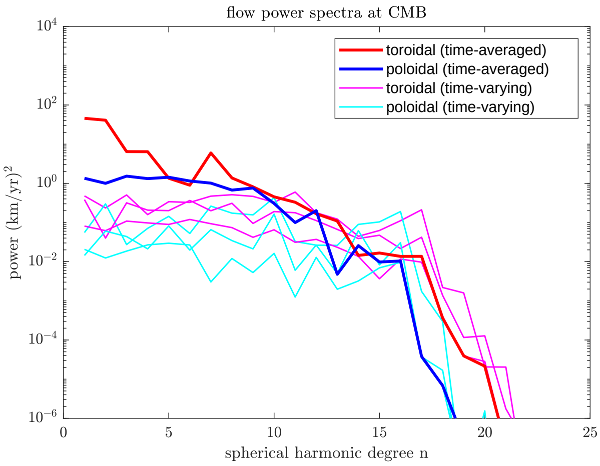


Figure 4.3: Model 1: Power spectra of time-averaged flow (thick lines) and snapshots of time-dependent flow (thin lines) taken every 2 years of the data period, starting September 2014.

The alternating nature of the azimuthal part of the time-dependent flow seen at low latitudes in figure 4.2 is also visualized in figure 4.4. White regions represent areas and times of zero azimuthal flow. Along the horizontal axis, these white regions thus represent the areas of diverging and converging flow. Along the vertical axis, the white regions correspond to times where the flow changes direction from westward to eastward or vice versa. We thus see that the azimuthal flow at low latitudes is subject to a change in sign at almost all longitudes within the data period, although the specific time of this inversion varies at different longitudes. For example, we see a reversal of the time-dependent flow starting to occur under the Eastern Pacific ($\phi \approx 210°$) in early 2017. This inversion then continues westward until the flow under the rest of the Pacific has been inverted in 2018. Interestingly, the time span of this flow inversion somewhat corresponds to the period in which the last La Niña was active in the Pacific Ocean (e.g. C. Zhang et al. 2019). This observation led to the question of whether SV caused by ocean currents could perhaps be significant enough to be picked up by SWARM observations. This is investigated in chapter 5.

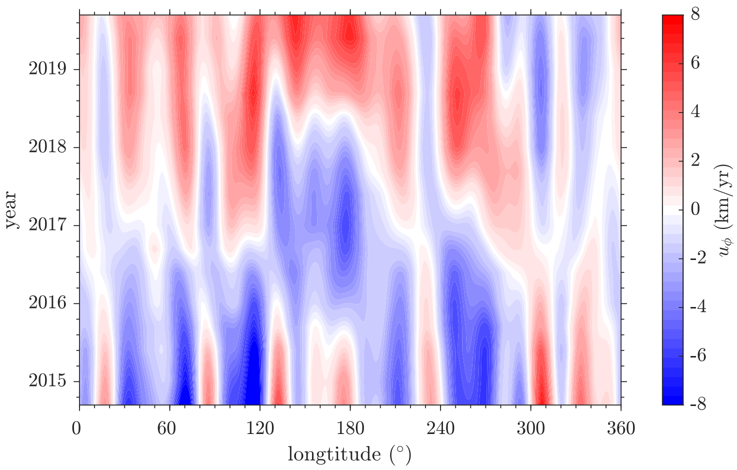


Figure 4.4: Model 1: Azimuthal, time-dependent flow, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The figure is centered in the Pacific.

Similar to figure 4.4, the azimuthal flow *acceleration* is shown in figure 4.5. For longitudes $0° - 120°E$ (i.e. roughly from western Europe to China's east coast) we mostly see eastward (positive) acceleration of core flows throughout the data period. From roughly $120° - 200°E$ (western Pacific), however, we see a change in sign from an initial westward acceleration to an eastward acceleration in early 2017. At $300° - 360°E$ (Atlantic Ocean) flow acceleration is mostly westward throughout the data period. The acceleration patterns correspond well with the flow reversals seen under the Pacific, Atlantic, and Indian Ocean in figure 4.2.

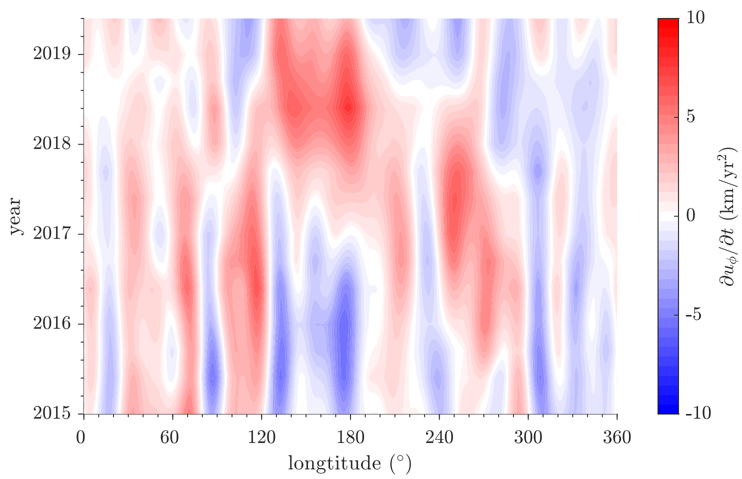


Figure 4.5: Model 1: Azimuthal flow acceleration, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The acceleration is estimated for a given epoch as the average acceleration explaining the difference in flow velocities between the prior and subsequent epoch. The figure is centered in the Pacific.

4.1.2 Predicted Secular Variation

Figure 4.6 shows the power spectra of SV predictions at the CMB by Model 1 as well as CHAOS-7.2. These power spectra are well in agreement with the model produced by Kloss and Finlay 2019. The total SV predicted by Model 1 (green) is very similar to the SV predicted by CHAOS-7.2 (red) up to SH degree 11, where they start to deviate. The power of the SV generated by the small-scale error (interaction of the flow with unresolved lengthscales of the magnetic field) grows as the SH degree increases and is greater than the power of the large-scale field SV at degree 14 and higher. Up to that point, the total SV is explained well by the large-scale field alone.

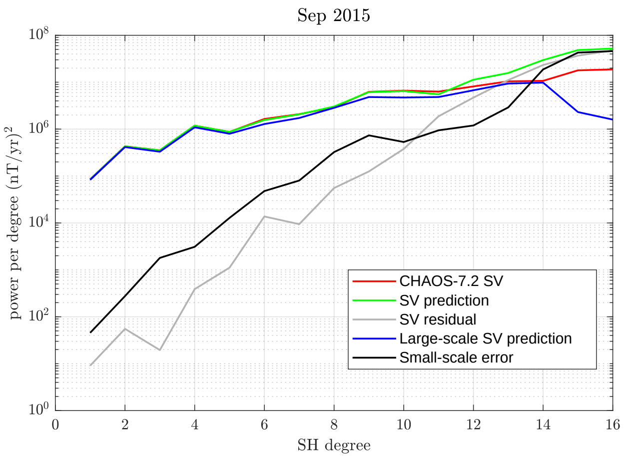


Figure 4.6: Model 1: Power spectra of SV predictions at the CMB generated by CHAOS-7.2 (red), the model (green), the SH coefficient residuals between the model and CHAOS-7.2 (grey), the large-scale part of the model only (blue), and the small-scale error part of the model only (black).

Figure 4.7 shows residuals between observed SV and SV predicted by the model. The distribution is close to Laplacian. Note that the long tails of the phi residuals for VO data (top left) have successfully been accounted for by the associated error estimates in the normalization by the expected data errors (top right). Figure 4.8 shows timeseries of SV predicted by the model and CHAOS-7.2 at example observatories. The model predictions are very similar to those of CHAOS-7.2, which was our aim. Notice in particular the reversal in trend of observed SV at the HON observatory in Hawaii (top) for the radial and meridional vector components. According to our model, this SV trend reversal in the Pacific seems to be explained by the reversal of core flow underneath.

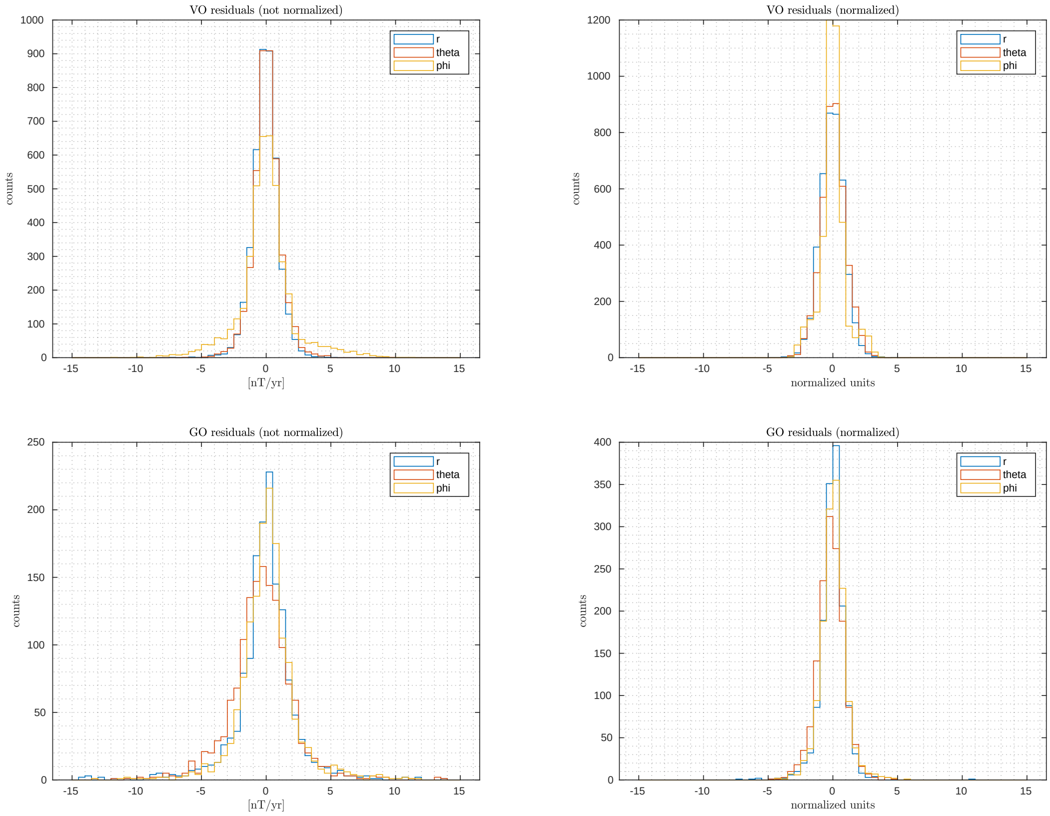


Figure 4.7: Model 1: Histograms of residuals between SV observations and model SV predictions. Histograms are shown for both VOs (top) and GOs (bottom), both non-normalized (left) and normalized with associated error estimates (right).

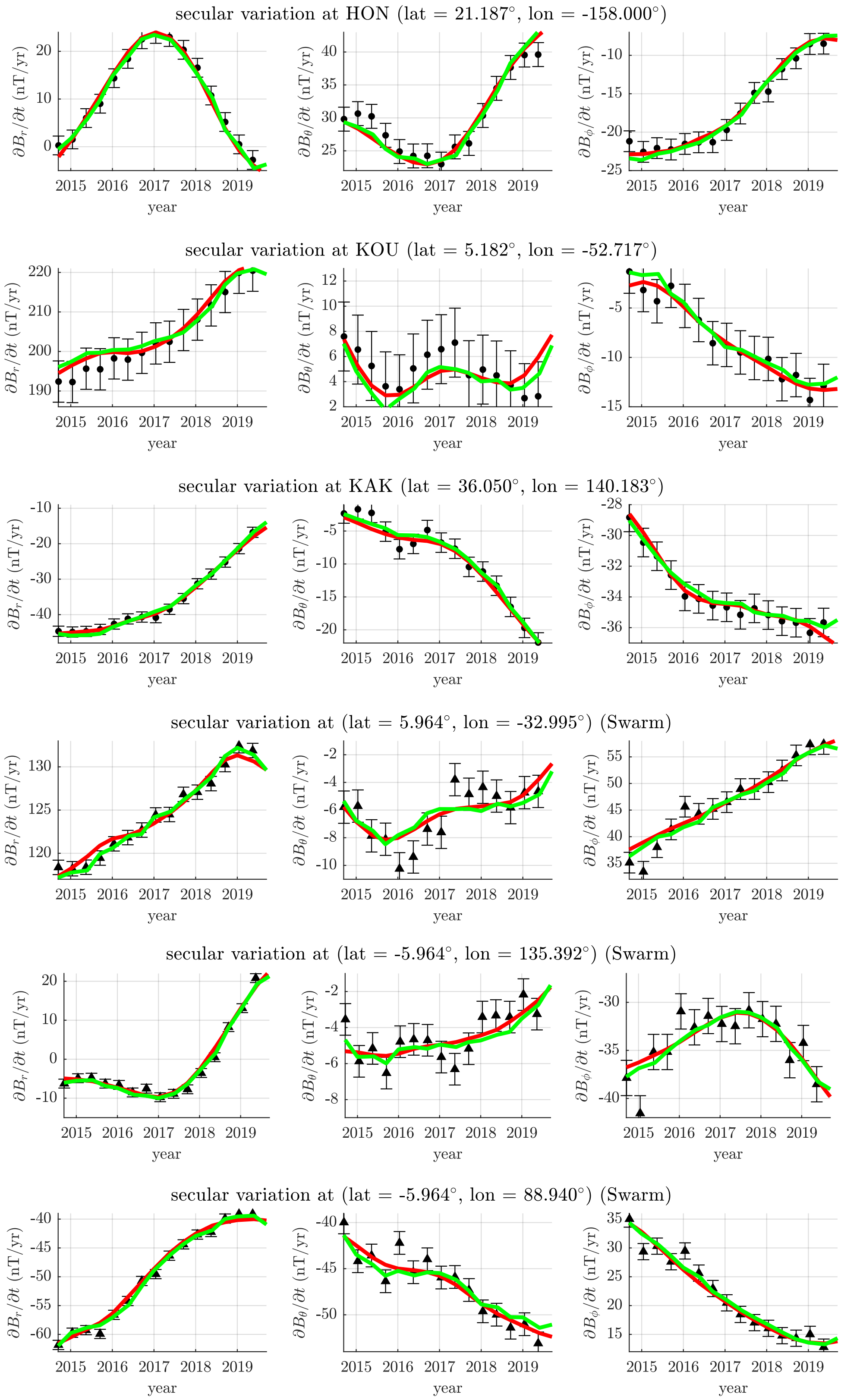


Figure 4.8: Model 1: Example timeseries of SV data from 3 GOs (HON, KOU, and KAK) and 3 VOs for all three vector components with associated error estimates. Model predictions (green) are shown along with CHAOS-7.2 predictions (red).

4.2 Model 2a: Relaxed Penalization of Equatorially Anti-Symmetric Flow

This section presents results for Model 2a, which was the preferred model from generation 2. Here, we applied the modifications to the regularization scheme presented in section 2.10, i.e. weaker penalization of small-scale, equatorially anti-symmetric flow. Model 2a and all other generation 2 models contain the same number of mode amplitudes and small-scale error coefficients as Model 1, i.e. a total of 26808 estimated parameters.

4.2.1 Estimated Flow

The time-average of Model 2a's total flow, seen in figure 4.9, remains very similar to that of Model 1. We see that the flow is dominated by gyres in both the northern and southern hemisphere, contributing to a strong westward flow under Africa and the Atlantic Ocean. We also see that the equatorial flow under the Pacific remains mostly eastward.

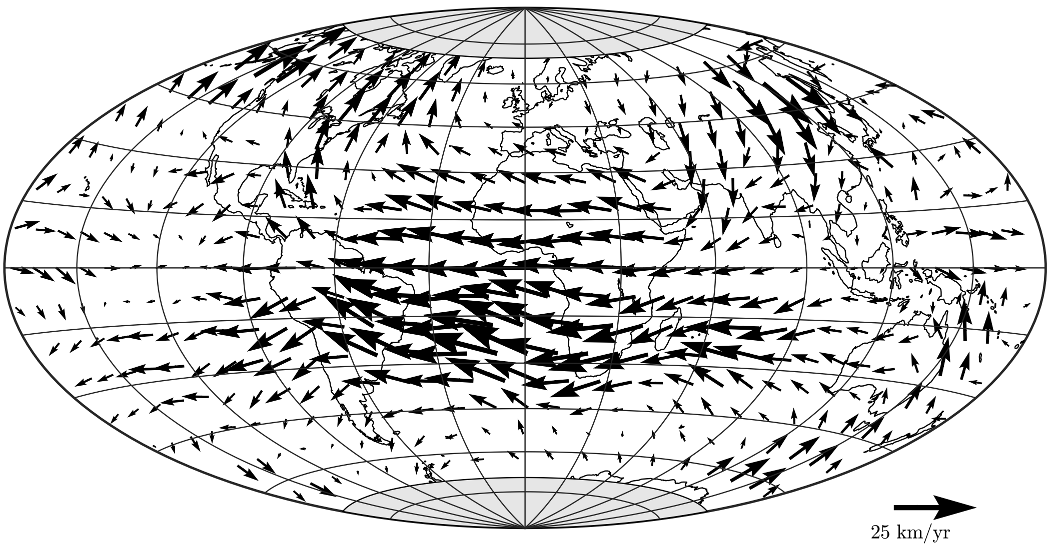


Figure 4.9: Model 2a: Time-average of the total flow on the CMB.

Effects of the new regularization scheme are more apparent in the time-dependent flow alone, seen in figure 4.10. One significant feature is a strong, longitudinally widespread equator crossing of the flow under the Pacific. The equator crossing is southward in the early data period, i.e. 2014 to 2015. From then on, it gradually transitions into a westward, azimuthal flow. In the middle of the data period (2016 to 2017), this clockwise rotation of the flow direction has continued to the point where we instead see a northward equator crossing under the Pacific. Towards the end of the data period (2018 to 2019), the sub-Pacific flow is mostly eastward, although it still retains some velocity in the northward direction, particularly under the western Pacific near Indonesia. The clockwise rotation of the flow direction is in fact seen to some extent all along the equator. Weaker equator crossings are also seen under the mid Atlantic and the Indian Ocean. An equator crossing under the Indian Ocean forms the end of a staircase-like pattern, starting under the Sahara Desert, with two ”steps” of alternating azimuthal and meridional flow going across Africa. This pattern is particularly visible in the September epochs of 2018 and 2019, but seems to be a somewhat persistent feature, varying in strength and flow direction. Similar to Model 1, we see a general reversal of azimuthal flow at most longitudes throughout the data period. Another feature from Model 1 that is preserved is that of relatively weak flows in the mid data period and stronger flows in the early and late epochs. For Model 2a, the power of the symmetric flow explains about 67% of the total flow power (down from %70 for Model 1), while the time-dependent flow accounts for about 7% of the total power (up from 6% for Model 1).

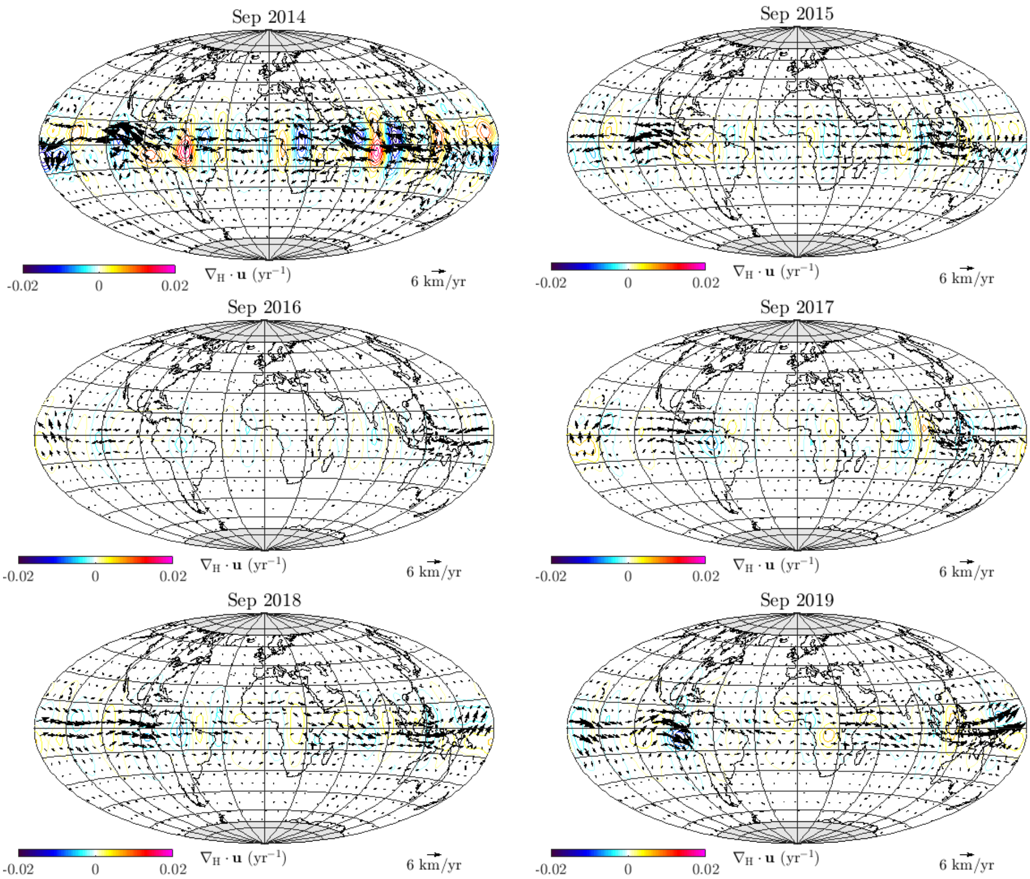


Figure 4.10: Model 2a: Time-dependent flow at the CMB for September of every year in the data period. Arrows represent flow and contours represent divergence/convergence, corresponding to upwelling/downwelling of the core fluid.

The intended effects of the new regularization scheme are clearly visible in the power spectra of the flow, seen in figure 4.11. Significant power is retained at much higher SH degrees. This is due to the relaxed penalization of anti-symmetric flow on small length scales. Above SH degree 18, the power of the time-dependent flow (purple and cyan lines) is entirely explained by anti-symmetric flow (dashed lines).

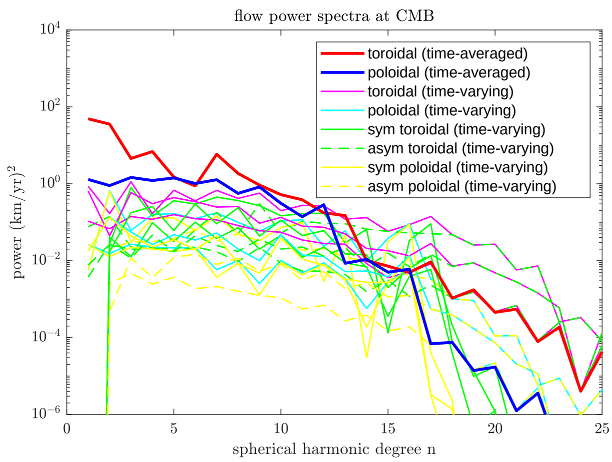


Figure 4.11: Model 2a: Power spectra of time-averaged flow (red, blue) and snapshots of time-dependent flow (purple, cyan) taken every 2 years of the data period, starting September 2014. Power spectra for the equatorially symmetric (solid green, solid yellow) and anti-symmetric (dashed green, dashed yellow) parts of the time-dependent flow snapshots are also shown. Two of the snapshots (corresponding to September 2014 and September 2018) hold noticeably more power.

The azimuthal, time-dependent flow at the equator is shown in figure 4.12. Note that the flow reversals (e.g. under the Pacific in the period 2017-18), previously seen in Model 1, are still present.

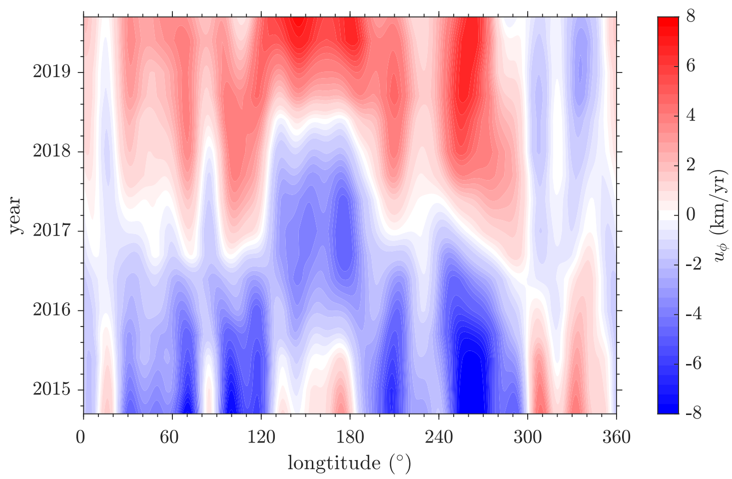


Figure 4.12: Model 2a: Azimuthal, time-dependent flow, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The figure is centered in the Pacific.

The main features of the flow acceleration pattern seen in figure 4.13 are also very similar to Model 1. Flow acceleration is mostly eastward under the Indian Ocean and mostly westward under the Atlantic, throughout the data period. Under the Pacific, we again see a change in acceleration from westward to eastward.

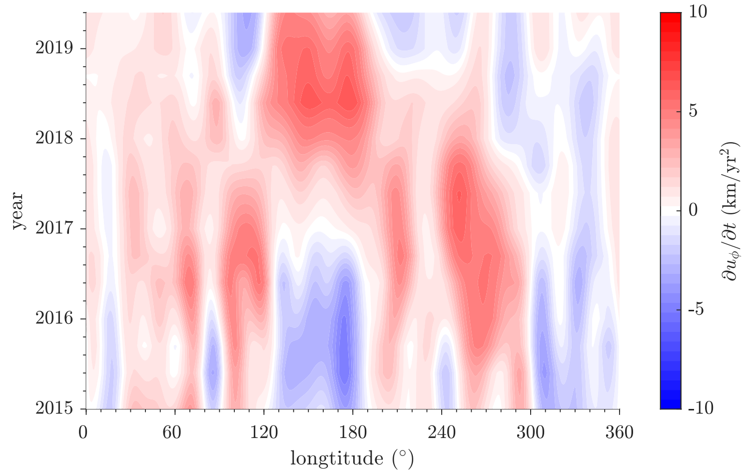


Figure 4.13: Model 2a: Azimuthal flow acceleration, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The acceleration is estimated for a given epoch as the average acceleration explaining the difference in flow velocities between the prior and subsequent epoch. The figure is centered in the Pacific.

4.2.2 Predicted Secular Variation

Figures 4.14-4.16 show that the new regularization scheme has not had any significant effect on the model's predictions of SV. The power spectrum of the SV is practically identical to that of Model 1, the residual distribution is still Laplacian, and timeseries of SV at observatories are still generally predicted equally well by the model and CHAOS-7.2. Note that this is mainly because regularization parameters are adjusted to achieve roughly the same misfit as CHAOS-7.2 (see Φ_{sv} in table 4.1). This clearly illustrates the non-uniqueness in the core flow inversion problem, as several solutions can fit the data to the same level.

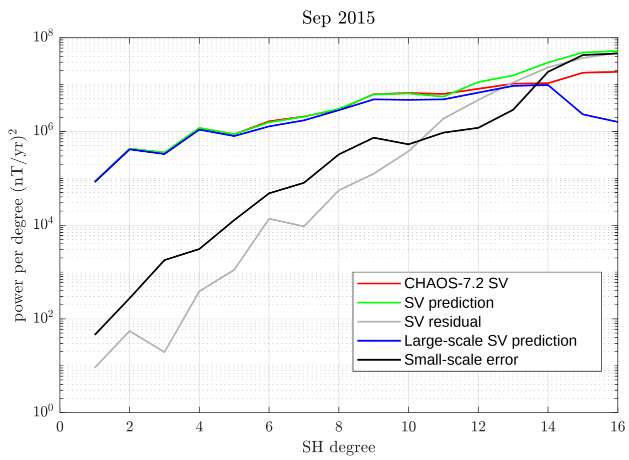


Figure 4.14: Model 2a: Power spectra of SV predictions at the CMB generated by CHAOS-7.2 (red), the model (green), the SH coefficient residuals between the model and CHAOS-7.2 (grey), the large-scale part of the model only (blue), and the small-scale error part of the model only (black).

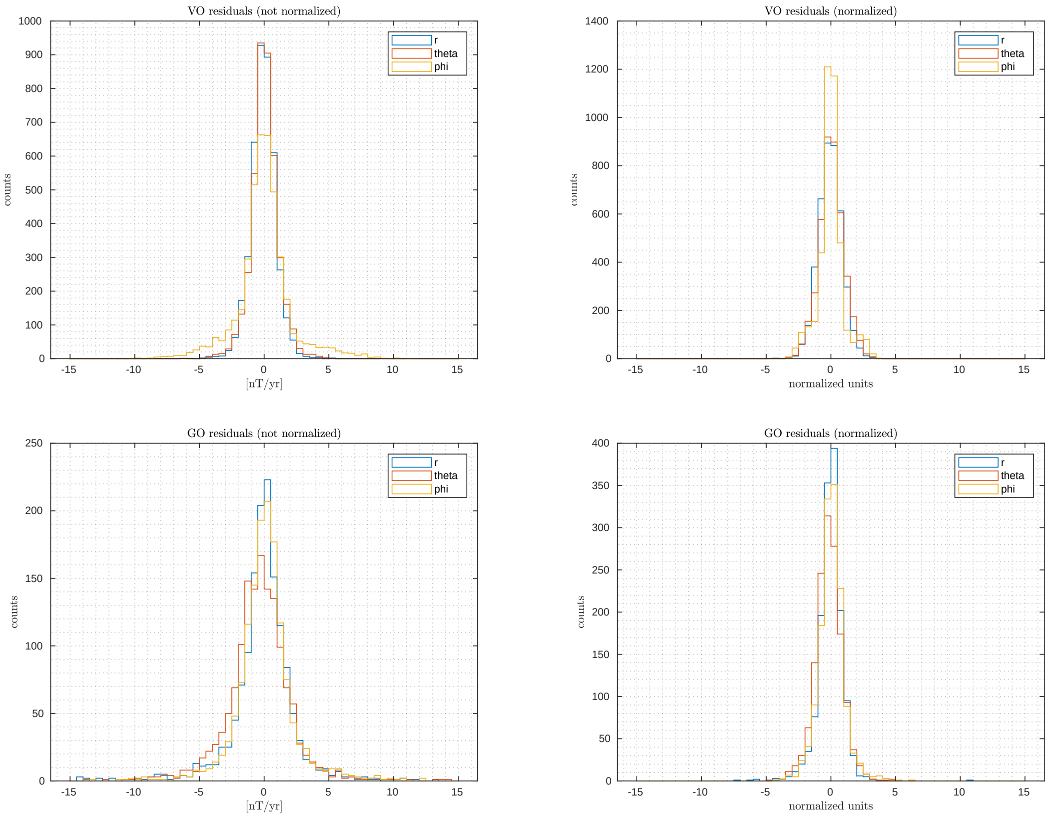


Figure 4.15: Model 2a: Histograms of residuals between SV observations and model SV predictions. Histograms are shown for both VOs (top) and GOs (bottom), both non-normalized (left) and normalized with associated error estimates (right).

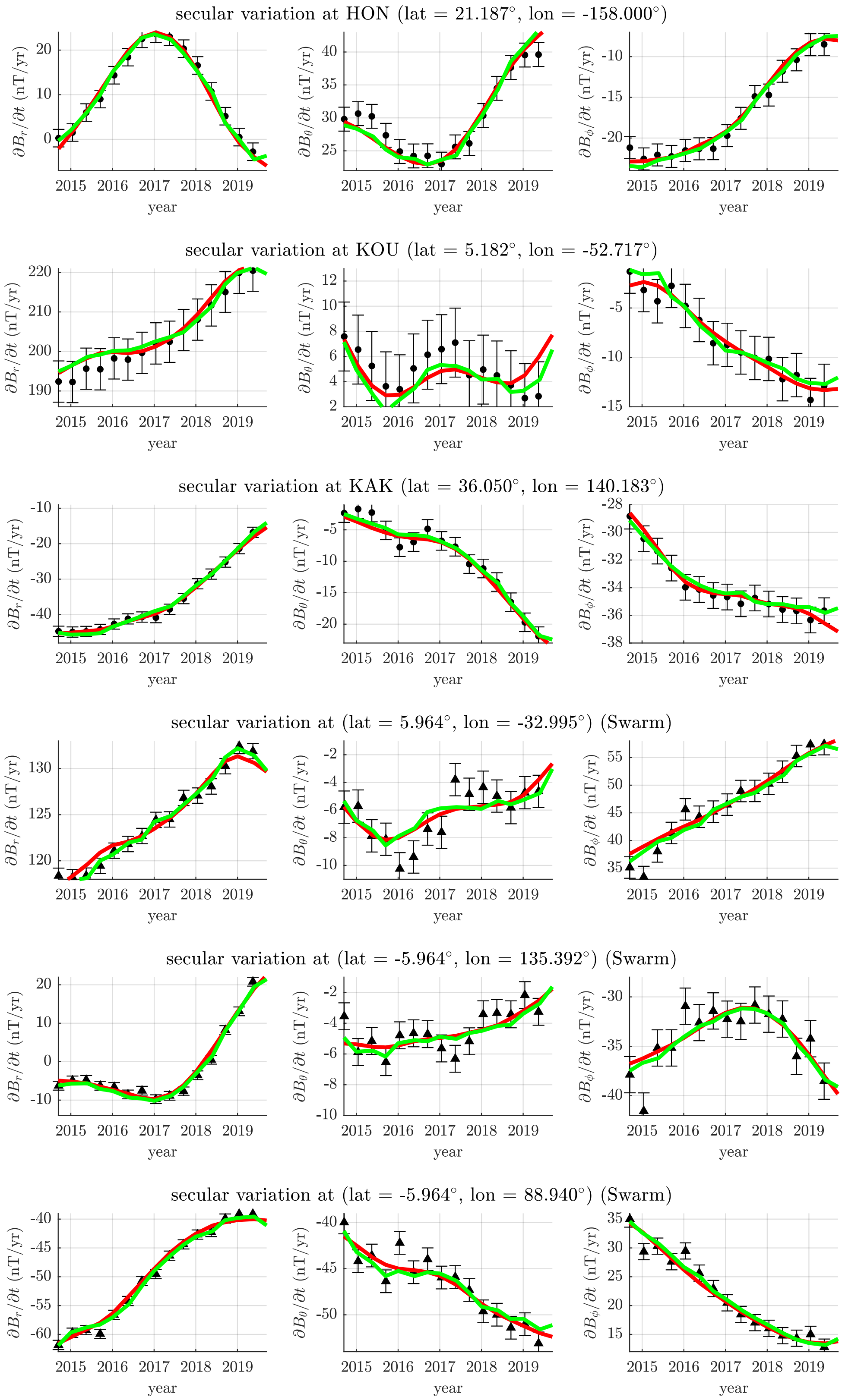


Figure 4.16: Model 2a: Example timeseries of SV data from 3 GOs (HON, KOU, and KAK) and 3 VOs for all three vector components with associated error estimates. Model predictions (green) are shown along with CHAOS-7.2 predictions (red).

4.3 Model 3c: Diffusion

This section presents the results for Model 3c, which was the preferred model for generation 3. Here, we applied the modifications to the regularization scheme presented in both section 2.10 and 2.11, i.e. relaxed penalization of small-scale, equatorially anti-symmetric flow *and* accounting for diffusion. On top of the 26808 parameters estimated in previous generation models, generation 3 models also estimate 288 SH coefficients explaining the diffusion generated SV for every epoch. The total number of estimated parameters for generation 3 models is therefore $26808 + 16 \cdot 288 = 31416$.

4.3.1 Estimated Flow

Figure 4.17 shows that the general structure of the time-averaged total flow is still preserved with diffusion taken into account, i.e. gyres in the northern and southern hemispheres with a strong westward flow under Africa and the Atlantic. The flow does differ from previous generation models in some important aspects, however, namely the flow speeds in certain areas. For example, we see significantly slower flows under the Pacific as well as under Australia and Indonesia. This decrease in flow speeds is also seen to a lesser extent all over the CMB. Furthermore, the flow under the Pacific is no longer noticably dominated by eastward motion, except for the western part, close to Indonesia. Instead, the averaged flow includes a southward equator crossing in the mid Pacific.

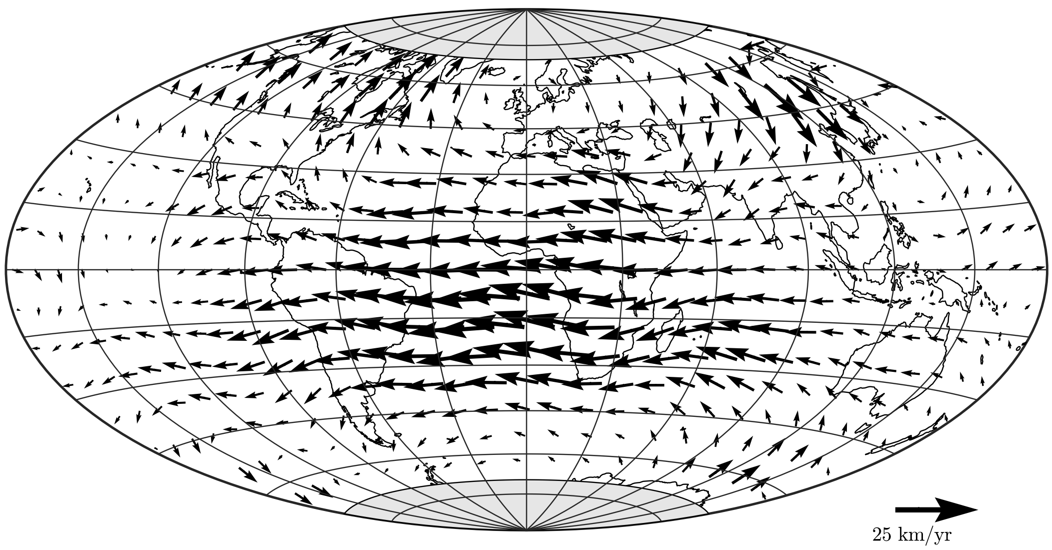


Figure 4.17: Model 3c: Time-average of the total flow on the CMB.

Figure 4.18 shows that the time-dependent flow is still strongest in the early and late epochs with a quieter period in between. Other features of Model 2a are also still present. For example the wide equator crossing under the western Pacific, and the staircase pattern between the Sahara Desert and the Indian Ocean. The speeds of the time-dependent flow are not as significantly reduced as the steady flow, suggesting that the diffusion generated SV is mostly taking power from the steady flow. The f_t norm in table 4.1, also indicates this, as the flow power is explained by time-dependent flow to a greater extent, i.e. about 11%. The flow is also a bit less symmetric, with about 64% of the flow power being explained by symmetric flow.

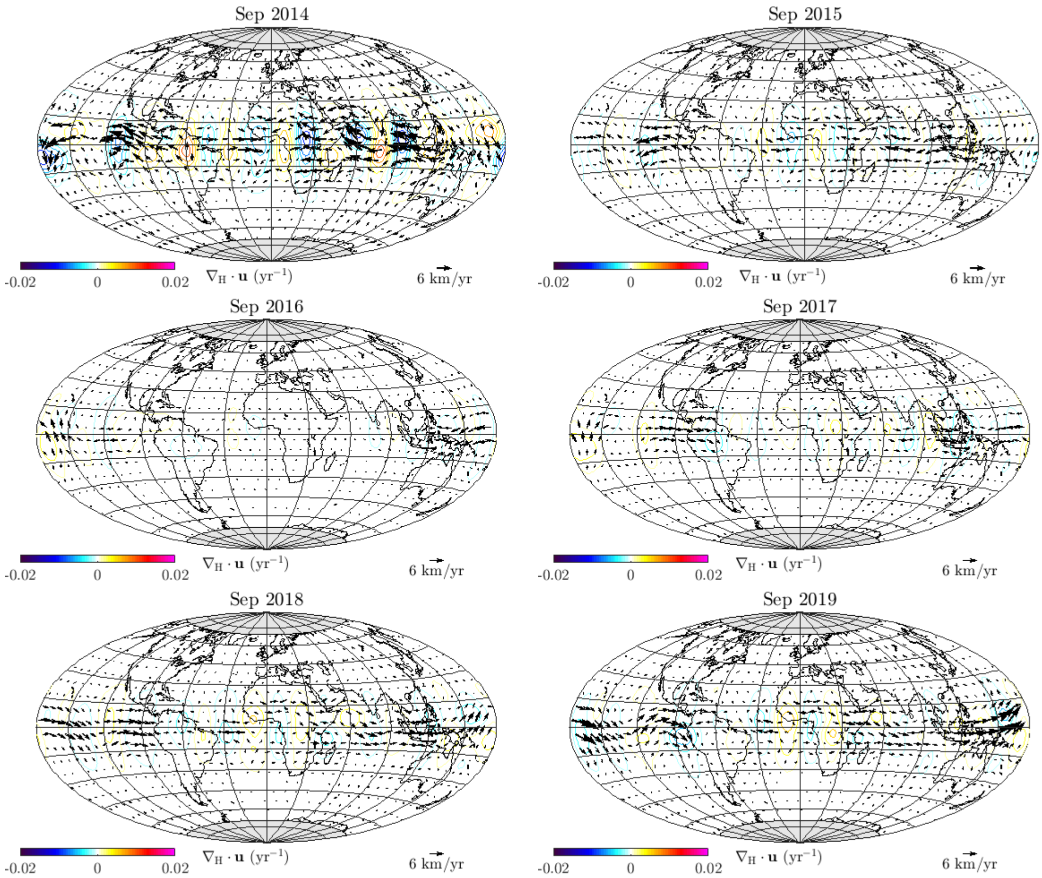


Figure 4.18: Model 3c: Time-dependent flow at the CMB for September of every year in the data period. Arrows represent flow and contours represent divergence/convergence, corresponding to upwelling/downwelling of the core fluid.

As seen in figure 4.19, the total time-averaged flow (red and blue lines) loses power at a faster rate up to SH degree 15, than it did in Model 2a. This is especially true for the poloidal part (blue). At SH degrees above 18, the time-dependent flow power is still entirely explained by anti-symmetric flow. In figure 4.20, we again see that the equatorial azimuthal flow undergoes a reversal at all latitudes, due to the acceleration pattern in figure 4.21, which is also very similar to Model 2a.

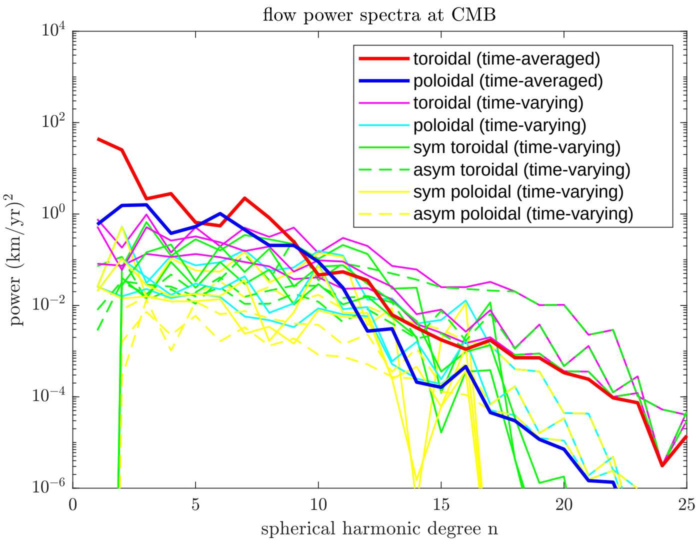


Figure 4.19: Model 3c: Power spectra of time-averaged flow (red, blue) and snapshots of time-dependent flow (purple, cyan) taken every 2 years of the data period, starting September 2014. Power spectra for the equatorially symmetric (solid green, solid yellow) and anti-symmetric (dashed green, dashed yellow) parts of the time-dependent flow snapshots are also shown.

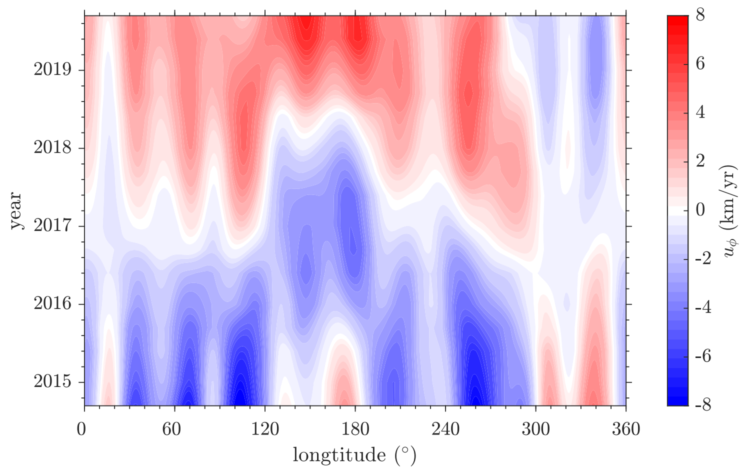


Figure 4.20: Model 3c: Azimuthal, time-dependent flow, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The figure is centered in the Pacific.

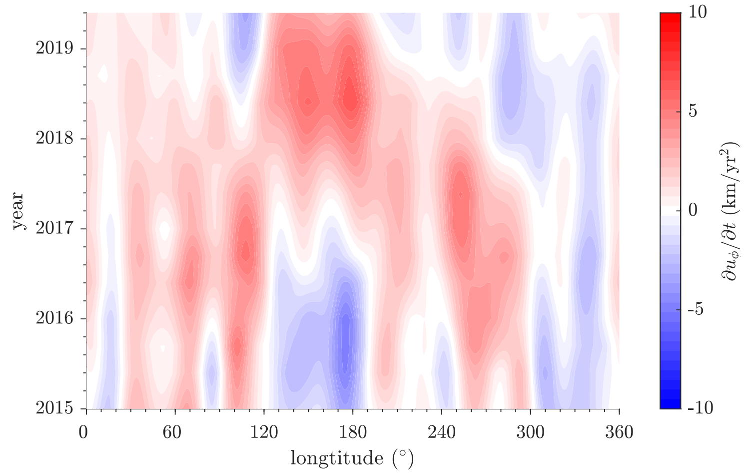


Figure 4.21: Model 3c: Azimuthal flow acceleration, averaged over latitudes between $15°$N and S in bins of $1°$ longitude, throughout the data period. The acceleration is estimated for a given epoch as the average acceleration explaining the difference in flow velocities between the prior and subsequent epoch. The figure is centered in the Pacific.

4.3.2 Predicted Secular Variation

A new aspect of the generation 3 models is that it is possible to examine the estimated signature in the SV due to magnetic diffusion. The diffusion generated SV at the CMB is shown throughout the data period in figure 4.22. Here we see a roughly equal amount of negative and positive patches of SV. Pairs of a negative and a positive patch may correspond to magnetic field line loops being pushed out through the surface, similar to those seen on the Sun's surface prior to a coronal mass ejection. One such pair is seen under Indonesia for all epochs. This is the strongest SV feature predicted by the diffusion part of the model. We also see that the diffusion generated SV is relatively static in both intensity and geographical location. There is some variation, however. For example we see that the negative patch under Indonesia is initially more elongated, covering areas beneath China as well. This northern part of the patch fades and is practically gone in 2016. In 2014 we also see two, small, positive patches under northern Africa, that seem to merge into a horseshoe shape as time progresses. A similar merger is also seen of the two patches under northern South America. Apart from Indonesia, Africa, and northern South America, significant patches are also seen at the Pacific, India, and east of the Caspian Sea and Madagascar.

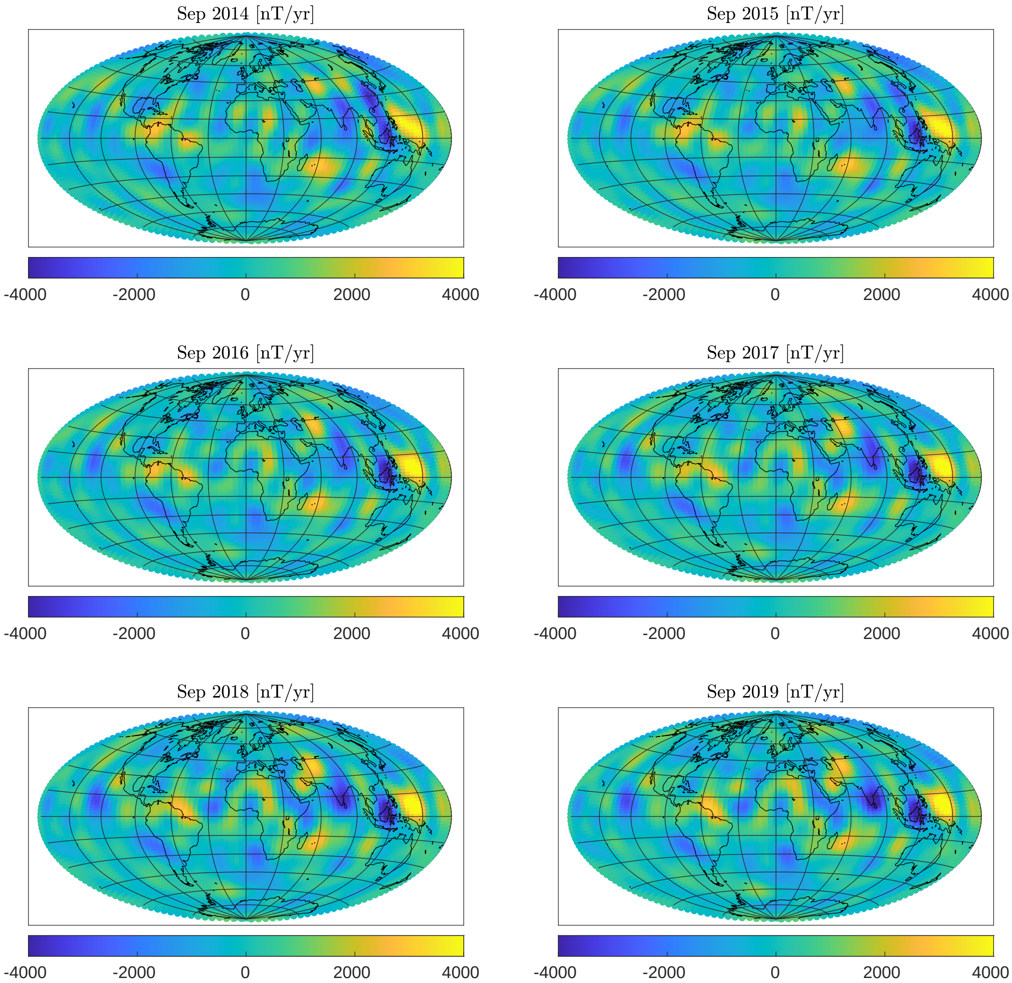


Figure 4.22: Model 3c: Prediction of diffusion generated SV at the CMB for September of every year within the data period.

In this model, some of the SV power that was explained by the small-scale error in previous generation models, seems to be explained by the diffusion instead. This is seen in figure 4.23 where the power of the small-scale error SV (black) is significantly smaller at large length scales than it was for previous generation models. The diffusion SV (yellow) is significant at these large length scales. Its power rises steadily until SH degree 4, and then remains roughly constant. The diffusion SV also seems to explain practically all the difference (grey) between the flow model and CHAOS-7.2 for SH degrees 7 and below.

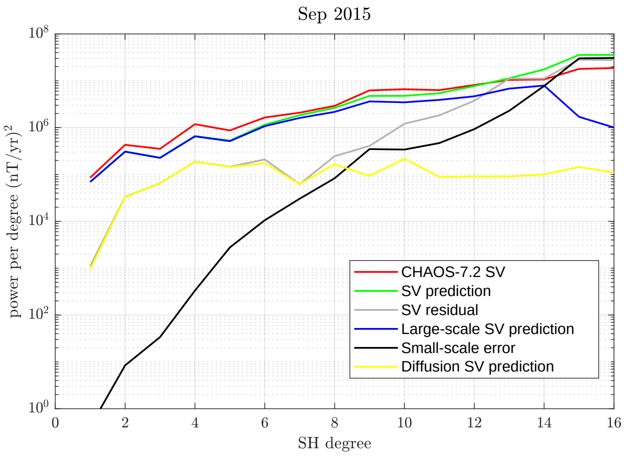


Figure 4.23: Model 3c: Power spectra of SV predictions at the CMB generated by CHAOS-7.2 (red), the model (green), the SH coefficient residuals between the model and CHAOS-7.2 (grey), the large-scale part of the model only (blue), the small-scale error part of the model only (black), and diffusion (yellow).

In figure 4.24 we again see a Laplacian distribution of residuals, similar to previous models, while figure 4.25 demonstrates the ability of the flow model to predict SV timeseries at observatories with roughly the same accuracy as CHAOS-7.2. In some locations, the timeseries fits are slightly different, however. For example, the flow model (green) seems to fit the theta component (middle) at KOU (second from the top) a bit better than CHAOS (red).

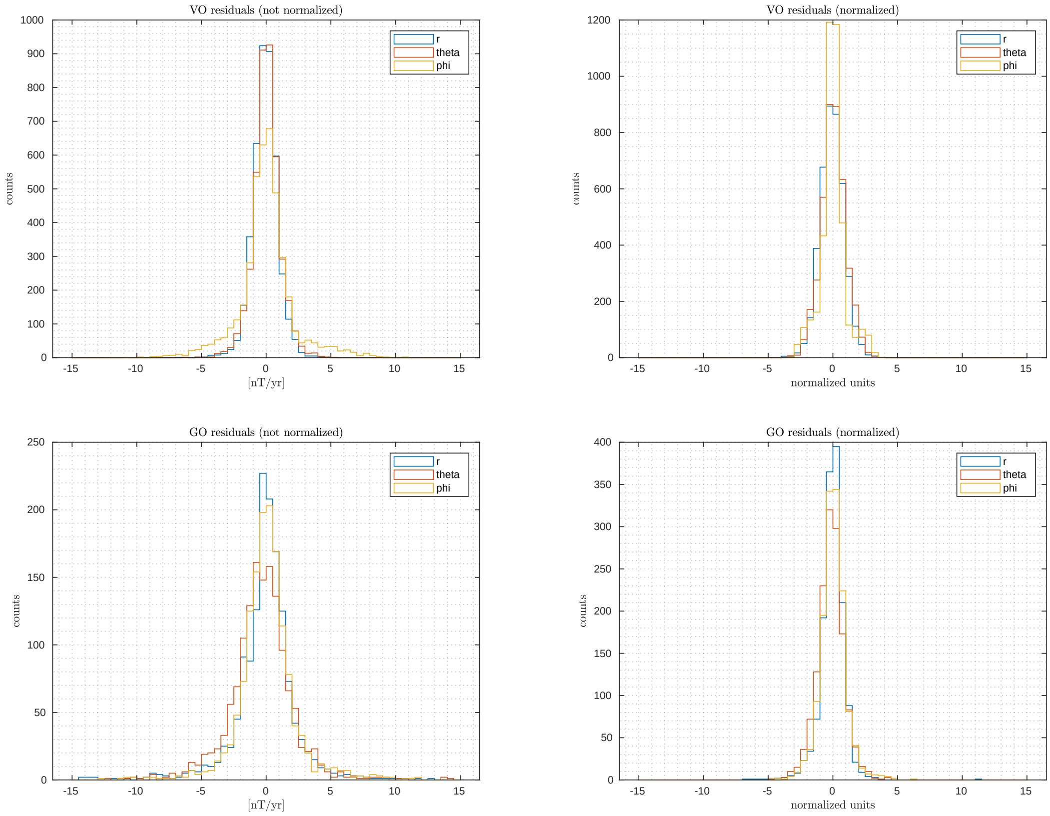


Figure 4.24: Model 3c: Histograms of residuals between SV observations and model SV predictions. Histograms are shown for both VOs (top) and GOs (bottom), both non-normalized (left) and normalized with associated error estimates (right).

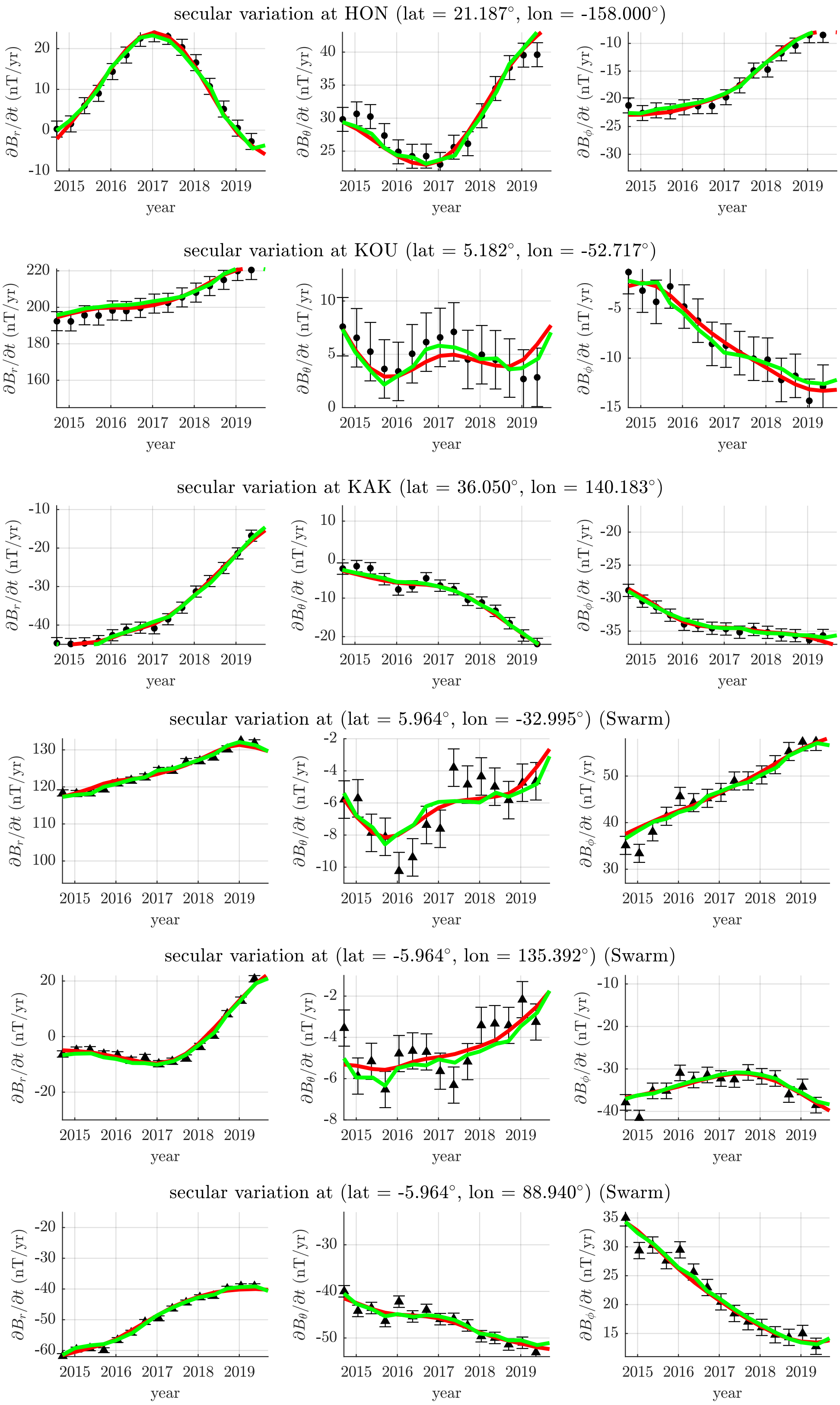


Figure 4.25: Model 3c: Example timeseries of SV data from 3 ground ground observatories (HON, KOU, and KAK) and 3 virtual observatories for all three vector components with associated error estimates. Model predictions (green) are shown along with CHAOS-7.2 predictions (red).

5 Discussion

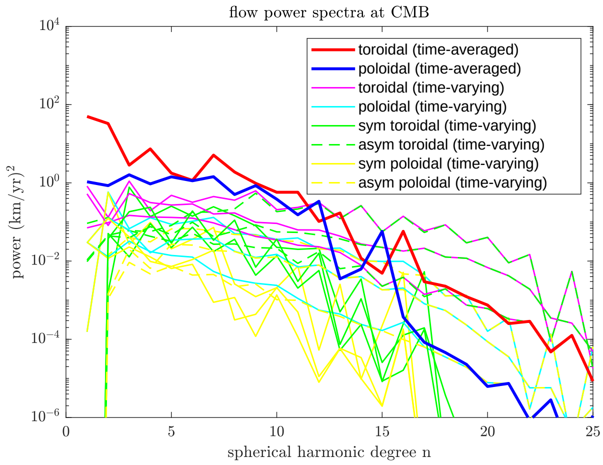
5.1 Non-Uniqueness of the Presented Models

The preferred models presented in the previous chapter were chosen after producing a large number of test models including, but not limited to, the ones shown in table 4.1. Even with the various assumptions about core flow, presented in chapter 2, the inverse problem is still highly non-unique. Achieving a model that is able to explain the SV observations well is therefore not sufficient to determine with certainty that the flows of that model are accurate representations of the actual flow in the Earth's core. This is demonstrated in table 4.1 where we see that models produced with different regularization schemes and parameters all manage to roughly achieve the same benchmark misfit as the CHAOS-7.2 model ($\Phi_{sv} = 0.6742$). As we saw for the preferred models of the previous chapter, each of these models have flows that are unique in some aspects. We will now further demonstrate the non-uniqueness of the inverse problem by illustrating the differences between models of the same generation, i.e. with the same regularization scheme, but varying regularization parameters and explain the choice of the preferred models.

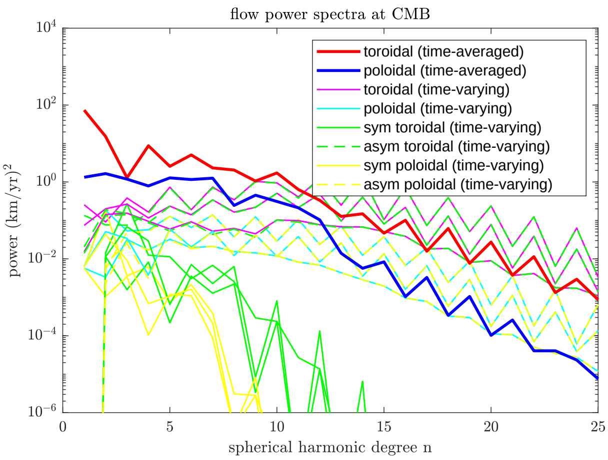
5.1.1 Generation 2

In generation 2 models, the regularization scheme modification described in section 2.10 was applied. The models of generation 2 differ from one another by varying the regularization parameters penalizing equatorially symmetric and anti-symmetric flow, i.e. λ_t^S and λ_t^A. The four generation 2 test models listed in table 4.1 thus have increasingly anti-symmetric flows with Model 2d being most anti-symmetric. This is demonstrated in figure 5.1, which shows the power spectrum of the toroidal-poloidal flow for Model 2b, 2c, and 2d (see figure 4.11 for corresponding figure for Model 2a). We see that allowing for a more equatorially anti-symmetric flow results in more overall power (red and blue lines) at smaller length scales. The reduced penalization of anti-symmetric flows are compensated for by increasing the penalization of symmetric flow to maintain the CHAOS-7.2 SV data misfit benchmark. The power spectra of symmetric flow (solid yellow and green lines) thus drop more rapidly with increasing SH degree for the more anti-symmetric models.

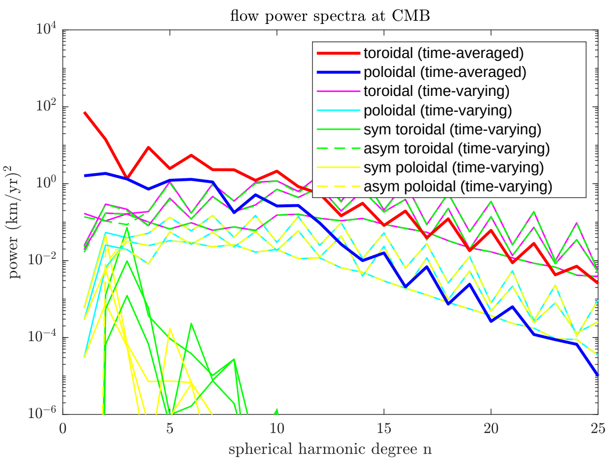
In figure 5.2 we see the effects of this on the flow. In choosing the preferred generation 2 model, Model 2c and 2d were deemed too extreme. In both models, the anti-symmetric flow explains the majority of the flow power and they exhibit very large-scale equator crossings of strong northward flows under South America and Indonesia. These dramatic features are not required to fit the observations, as shown by Model 2a (figure 4.9) and Model 2b. Although the aim of generation 2 models was to allow for equator crossings, we aim for more local crossings to maintain a mostly symmetric flow at the equator. Several core flow studies (e.g. Schaeffer and Pais 2011) indicate that breakdowns of symmetric flow at the equator are small-scale in nature, while the large-scale structure of the flow remains mostly symmetric. Schaeffer and Pais 2011 thus found that increasing the truncation degree of their flow decreased the relative power of the symmetric flow. For their highest truncation degree, the power of their flow, integrated over the CMB, was 66% symmetric in the period investigated, i.e. 1997-2010. In this study, Model 2a achieved a similar symmetric power ratio of 67% (see f_S in table 4.1). It should be noted, however, that the ratio of symmetric flow power is likely not constant in time. For example, Gillet et al. 2011 found that flow symmetry at the CMB equator generally increased in the period 1840-2010. Even so, Model 2b was also deemed too anti-symmetric, at only 56% symmetric power, leaving Model 2a as the preferred model of generation 2.



(a) Model 2b

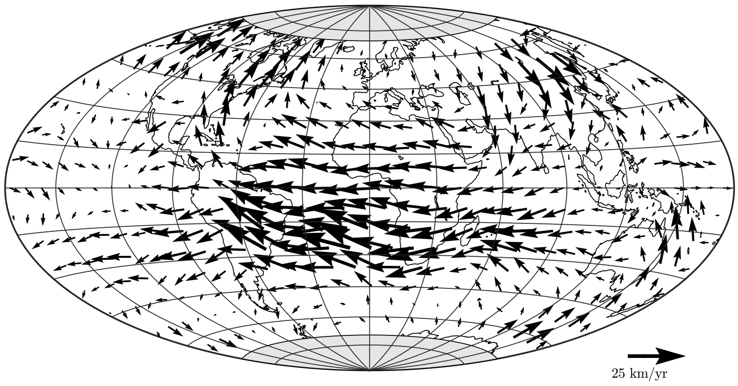


(b) Model 2c

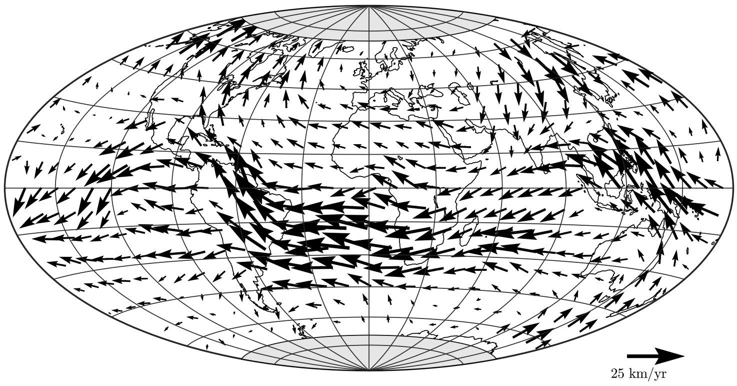


(c) Model 2d

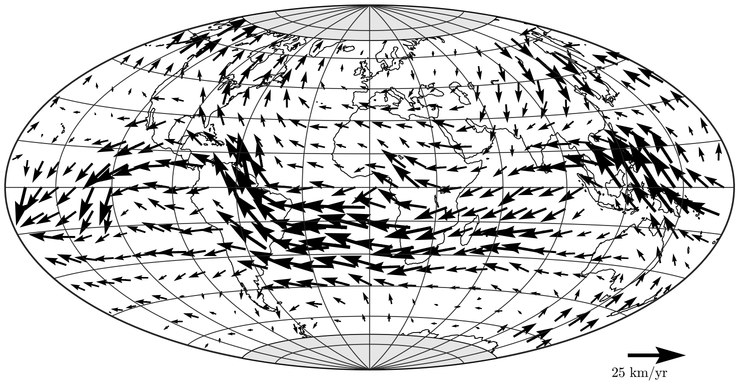
Figure 5.1: Power spectra of time-averaged flow (red, blue) and snapshots of time-dependent flow (purple, cyan) taken every 2 years of the data period, starting September 2014. Power spectra for the equatorially symmetric (solid green, solid yellow) and anti-symmetric (dashed green, dashed yellow) parts of the time-dependent flow snapshots are also shown.



(a) Model 2b



(b) Model 2c



(c) Model 2d

Figure 5.2: Time-average of the total flow on the CMB from September 2014 to September 2019.

5.1.2 Generation 3

In generation 3 models, the regularization scheme modifications of both section 2.10 and 2.11 were applied. The generation 3 models differ from one another by varying the regularization parameter penalizing diffusion generated secular acceleration, λ_z. Model 3a features zero time regularization with $\lambda_z = 0$. In this model, the ratios of regularization parameters in Model 2a were adopted, but had to be increased by a factor of 7 to achieve the CHAOS-7.2 misfit benchmark. The result of this was almost zero core flow, because of the harsh flow penalization, with the magnetic diffusion explaining most of the observed SV, but changing dramatically, and unphysically, between subsequent epochs, which are only seperated by four months. This is unrealistic, since flow in the core is required for dynamo action to be maintained. Apart from having almost zero flow, we also reason that this flow model is unrealistic, because diffusion is estimated for each epoch completely independently. There should be some temporal correlation between subsequent epochs, which can only be achieved with a non-zero λ_z. Diffusion generated SV predictions at the CMB for Model 3a are shown for every September of the data period in figure 5.3. Note that the magnitude of the SV patches is significantly higher than for Model 3c (figure 4.22). Variations of the SV patches are also more obvious.

Conversely, Model 3b features a very strong time regularization of the magnetic diffusion, which makes it almost steady throughout the studied period. It was constructed by maintaining all the flow regularization parameters of Model 2a and instead increasing λ_z until the CHAOS-7.2 misfit benchmark was achieved. This model was deemed unrealistic because it almost nullified the steady part of the flow, resulting in an almost entirely time-dependent flow. The strong time regularization also means that changes to the diffusion generated SV will be strongly linked to changes of the flow. It should ideally be independent (the diffusion term of the induction equation does not depend on a moving conductor). The predicted diffusion generated SV at the CMB is shown for Model 3b in figure 5.4.

The preferred Model 3c is a compromise between the two extremes of Model 3a and 3b, and features a moderate time regularization. It was produced by reducing λ_z in Model 3b and increasing the other regularization parameters by a factor of 1.5. This resulted in a moderate reduction of the flow, as intended. Also, the magnitudes of SV patches (see figure 4.22) correspond well with those of the dynamo realizations (figure 2.2).

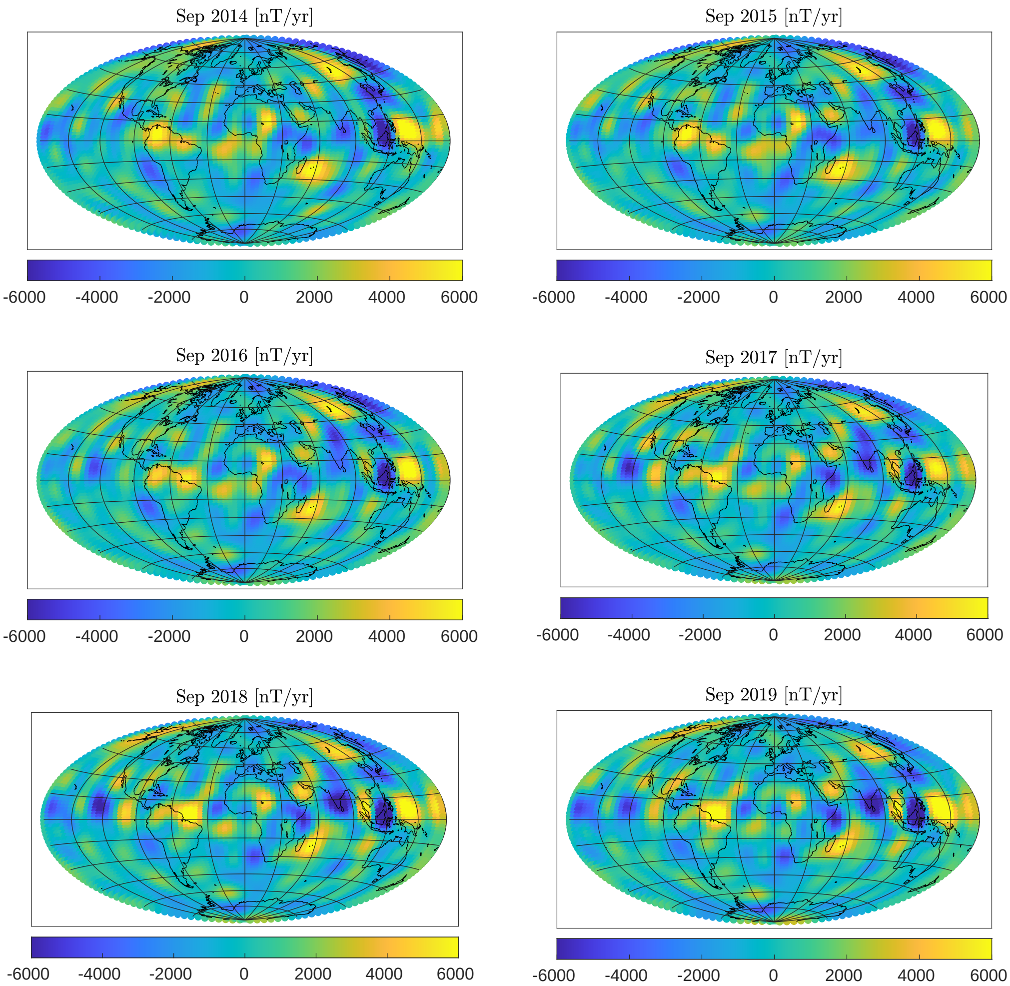


Figure 5.3: Model 3a: Prediction of diffusion generated SV at the CMB for September of every year within the data period.

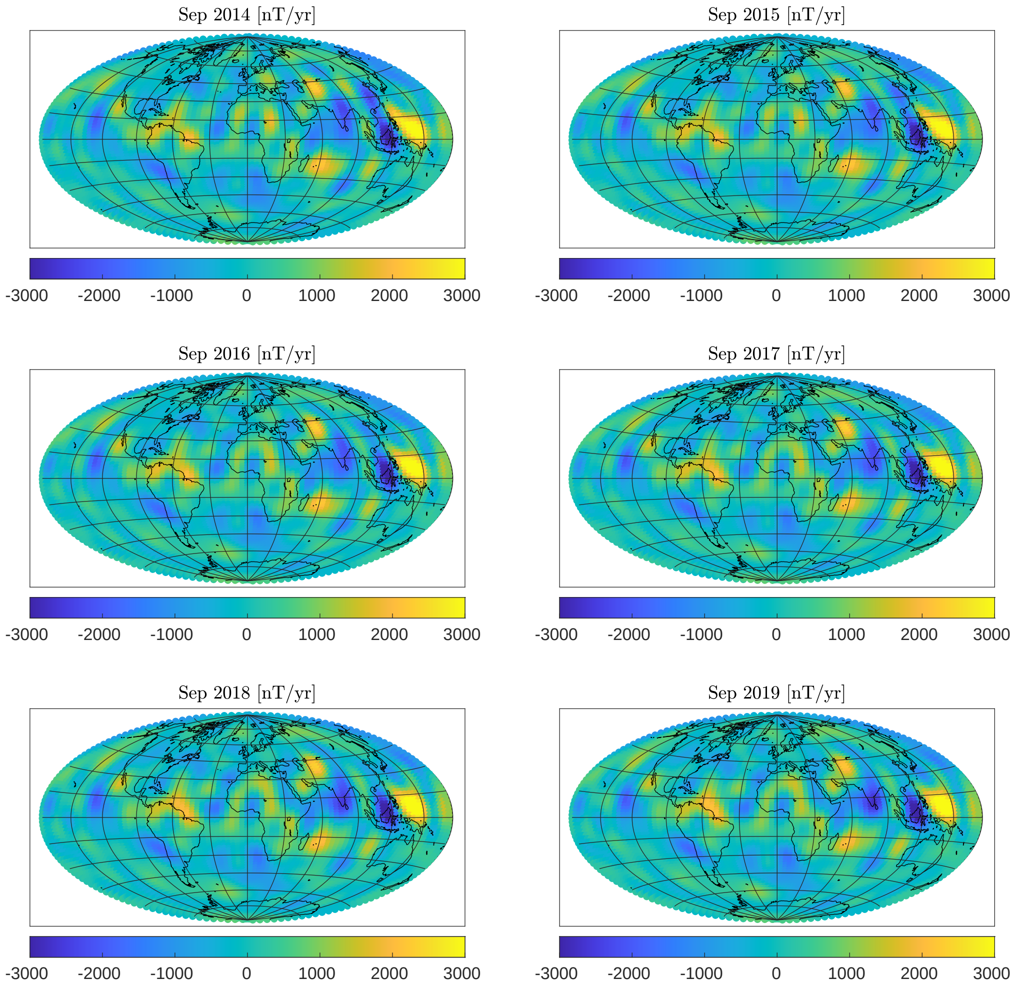


Figure 5.4: Model 3b: Prediction of diffusion generated SV at the CMB for September of every year within the data period.

5.2 Error Sources

5.2.1 Epochs of Displaced Phi Data

Figure 3.8 demonstrated that estimated errors of the phi (azimuthal) component of SV observations at the VOs were significantly larger than the r and theta component errors. This is also apparent in figures 4.7, 4.15, and 4.24, where the distributions of phi residuals are characterized by relatively long tails. Investigation of SV residuals for individual epochs revealed that the larger phi residuals were not caused by an error source that persists throughout the data period. It is rather caused by individual epochs of displaced phi residuals (not peaking at 0 nT/yr). One example of this is shown for the January 2015 epoch in figure 5.5. Notice that normalization is not sufficient to completely bring the peak to zero. A similar peak displacement was seen for January 2016

The displacements were also reproducible with CHAOS predictions, indicating that the issue was not caused by a bug in the inversion. As mentioned in chapter 3, the displacements are in fact present in the original SWARM data. The fact that they appear exclusively in January epochs links them to times of Summer in the northern hemisphere (recall that the SV is defined by measuring the difference in main field strength for a given epoch plus/minus 6 months) and the exact cause is still being investigated, but might be linked to enhanced field aligned current signatures following large magnetic storm events, e.g. April-August, 2015 (C. Finlay, pers. comm.).

Since the inversion includes a time regularization of the flow, subsequent epochs will be correlated. All epochs could therefore be somewhat affected by this error source. Normalization of residuals based on estimated SV error levels for the entire data period reasonably accounts for the displacements however (e.g. figure 4.24) and we find that our flows, like the reference CHAOS model, are unable to fit the SV data during these disturbed times. The effects on the final models are expected to be small. Even so, it is an error source that should ideally be accounted for in future studies when we know more about its origin.

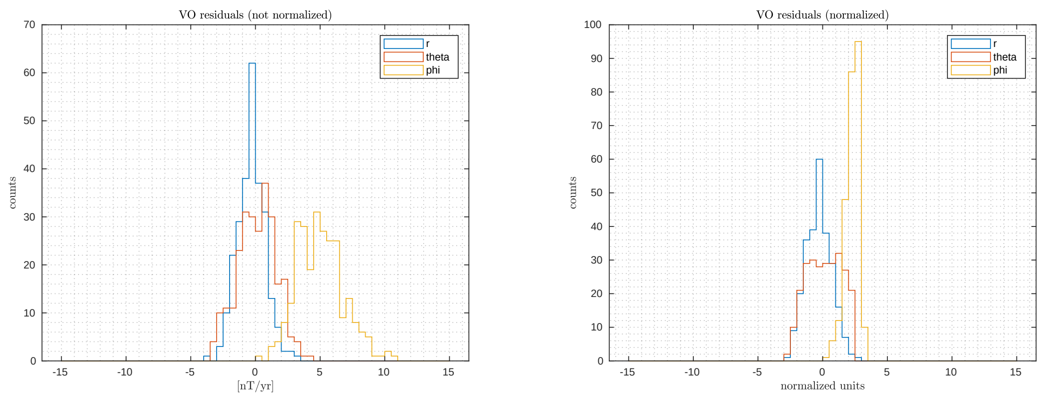


Figure 5.5: Histograms of residuals between SV observations and model SV predictions for the January 2015 epoch. Histograms are shown for both non-normalized residuals (left) and residuals normalized with associated error estimates (right).

5.2.2 Ocean Currents

In the following we investigate the possible contribution from ocean currents (e.g. La Niña, El Niño) to observed SV, and the possible influence of this signal on the estimated core flows. The ocean flow model used for the tests described here was derived at daily time resolution by J. Velimsky (Schnepf et al. 2020, pers. comm.) based on the ECCO ocean flow model version 4 (Marshall et al. 1997; Forget et al. 2015) and solving the magnetic induction equation including seasonal ocean conductivity variations (Tyler et al. 2017), the conductivity of oceanic, coastal, and continental sediments and an electrically conducting mantle (Grayver et al. 2017). The resulting poloidal magnetic field was then averaged using 1 year time-windows, and spherical harmonics up to degree 20 were fit using a 6th order B-spline model with 0.1yr knot spacing. The ocean model covers the period September 1997 to January 2000. We compute SV predictions from this oceanic model at the locations of the GOs and VOs and subsequently use these predictions as input data in the inversion scheme of Model 2a with all regularization parameters reduced to compensate for the shorter data period. This allows us to attempt computing a hypothetical core flow that produces the ocean generated SV to assess the size of possible error related to this source. The magnitude of this core flow then corresponds to the possible flow error imposed by ocean currents.

Figure 5.6 illustrates the amplitudes of the ocean generated SV at example GOs and VOs in the Pacific. The amplitudes are only a tiny fraction of actual observed SV (e.g. figure 4.16). The corresponding average core flow is shown in figure 5.7. These are on the order of a few m/yr, which again is an insignificant fraction of the flows estimated by our models (e.g. figure 4.9). Flow acceleration is also insignificant compared to our models, as seen in figure 5.8. These results suggest that the influence of ocean generated SV on core flow models is negligible, at least when considering annual SV data and the temporal regularization scheme we imposed here.

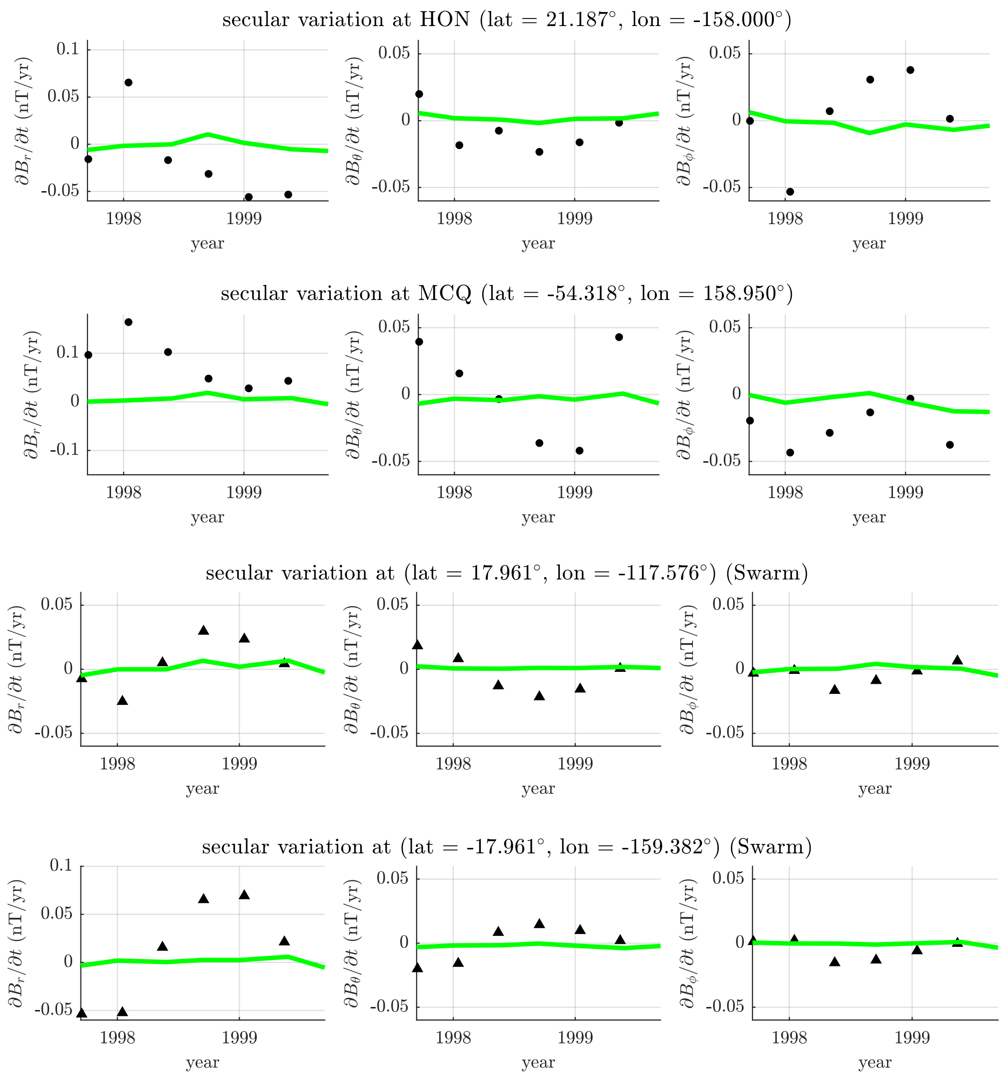


Figure 5.6: Examples of ocean model SV predictions at GOs and VOs in the Pacific (black dots). The fit by the inversion is also shown (green line).

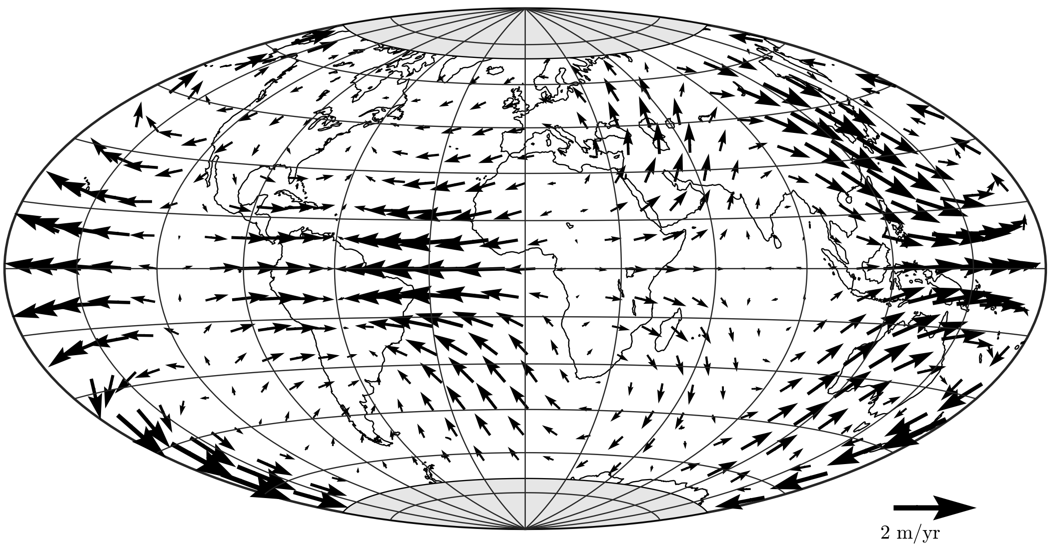


Figure 5.7: Hypothetical CMB flow explaining ocean generated SV.

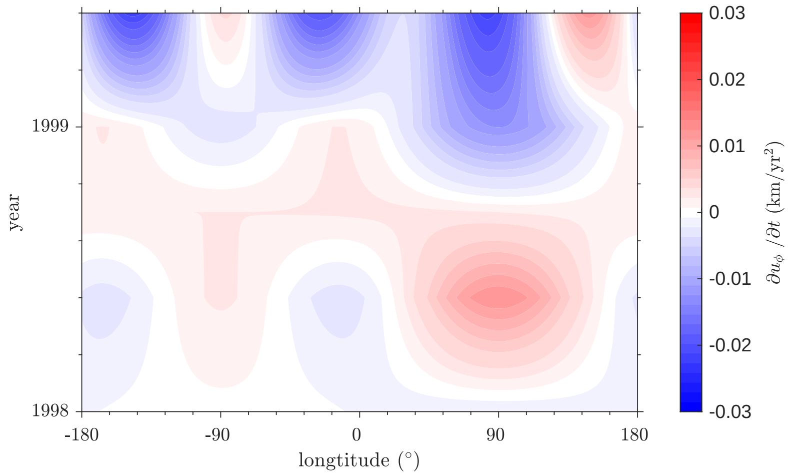


Figure 5.8: Acceleration of the azimuthal CMB flow at the equator explaining ocean generated SV.

5.3 Robust Features and Comparisons to Other Research

Considering the non-uniqueness of the inverse problem, making definitive conclusions about the core flow to a high degree of detail is a challenge. Judgements can however be made as to how likely various flow features are, based on their robustness, i.e. the degree to which they appear across different inversion methods. The various models produced in this and similar studies are investigated for this purpose.

In this study, three different regularization schemes were used to produce models with varying regularization parameters. Model 3a manifested almost zero flow, due to the high power of the non-regularized diffusion. Among the remaining models, recurring flow features include anti-cyclonic gyres in the northern and southern hemispheres that merge at low latitudes to create a strong westward flow under Africa and the Atlantic Ocean. Robust features for our models also include inter-annual reversals of equatorial, time-dependent, azimuthal flow at almost all longitudes, resulting in a period of relatively weak flows in the mid data period, i.e. 2016 to 2017. Time-dependent flows are generally strongest under the Pacific Ocean. Here, time-dependent azimuthal flow appears to reverse from initial westward flow to mostly eastward flow from 2017 onwards. The data seems to favor equator crossings under the Indian Ocean and regions between the mid Pacific and Indonesia, when equatorially anti-symmetric flow is allowed to retain power on small length scales (i.e. generation 2 and generation 3 models). All the preferred models (Model 1, 2a, 3c), manifest flows that are mostly symmetric and steady (f_S and f_t, table 4.1) and mostly toroidal for most length scales (e.g. figure 4.19). Among generation 3 models with varying temporal regularization, we consistently see that diffusion generated SV is strongest under Indonesia throughout the data period.

Overall, the robust features found in this study are in good agreement with the features previously identified by Kloss and Finlay 2019, but the new regularization schemes do produce some noticable differences. The relaxed penalization of the equatorially anti-symmetric modes on short length scales did have the intended effect of producing a more anti-symmetric flow with significant equator crossings. Kloss and Finlay 2019 also found a southward equator crossing under the Pacific in 2014, similar to e.g. figure 4.18, but did not find that it reversed to become a northward crossing in 2016, as suggested by Model 2a and 3c. Allowing diffusion to explain some of the observed SV (Model 3c) resulted in generally weaker flows, especially at the locations of strong diffusion patches (e.g. Indonesia), but the flow structure was generally preserved, apart from the elimination of the mostly eastward motion observed for the time-averaged flow under the Pacific in Model 1 and 2a. Amit and Christensen 2008 also compared inverted flow models with and without diffusion and found that diffusion did not dramatically affect CMB flow globally, but did cause some local changes. They also concluded that magnetic diffusion likely contributes to observed SV. Contrary to this study, however, they concluded that accounting for diffusion generally *increased* flow velocities. This further demonstrates the non-uniqueness of the inverse problem posed by the induction equation. Instead of using spherical harmonics, Amit and Christensen 2008 handled the non-uniqueness by making a helical flow assumption (Amit and Olson 2004), which relates the horizontal flow divergence to the radial vorticity, and they estimated magnetic diffusion using a simple correlation with the observed horizontal gradients of the flow, rather than using correlations obtained using statistics from numerical dynamo simulations, as we have done.

The global structure of hemispheric anticyclonic gyres and westward flows under Africa and the Atlantic Ocean is a classical structure observed in many studies, (e.g. Barrois et al. 2018; Pais and Jault 2008; Gillet et al. 2015). This CMB flow structure is thought to be the surface expression of a planetary eccentric gyre, i.e. a column of anti-cyclonic flow surrounding the solid inner core. Barrois et al. 2018 also found a patch of intense diffusion generated SV under Indonesia and a mostly equatorially symmetric flow outside the TC. The planetary eccentric gyre and mostly symmetric CMB flows are also found by Gillet et al. 2019 who studied the period 1880 to 2015, also using statistics from geodynamo simulations to mitigate the non-uniqueness.

6 Conclusions

We have succesfully developed and implemented modifications to the inversion scheme of Kloss and Finlay 2019 for core flow estimation, based on normal modes of core flow, described by K. Zhang and Liao 2017. These modifications were two-fold: First, we introduced a relaxed penalization of equatorially anti-symmetric flow on small length scales, to further allow for flows crossing the equator. Second, we introduced a means of accounting for magnetic diffusion, by using covariances in the SV due to magnetic diffusion, obtained from dynamo realizations from the Coupled Earth Model (Aubert et al. 2013). We conclude that the first modification results in significant changes to time-dependent flow structures on local scales, while global time-averaged flows are only slightly affected. We also conclude that the second modification sees a significant reduction in flow velocities, due to a significant fraction of SV power being accounted for by diffusion. At the same time, we recognize that the estimation of core flow from SV observations is a highly non-unique inverse problem and that stronger prior information or a more complete probabilistic inversion is needed to meaningfully constrain the regularization parameters. We have, however, demonstrated that core flows including diffusion and small-scale, equatorial anti-symmetry are plausible explanations of the inter-annual, azimuthal flow reversals at low latitudes also observed by e.g. Kloss and Finlay 2019 and Gillet et al. 2015.

Based on the preferred models presented in chapter 4, we argue that CMB flow in the period 2014 to 2019 is dominated by a largely steady, planetary-scale, eccentric gyre, consisting of anticyclonic gyres in the northern and southern hemispheres, that merge to form a strong westward flow under Africa and the Atlantic Ocean. We further suggest the existence of temporary equator crossings under the Indian and Pacific Ocean. The crossing under the Pacific is initially southward, but reverses into a northward flow in 2016. Similarly, we find inter-annual reversals of the time-dependent azimuthal flow all around the CMB equator, resulting in a period of relatively weak time-dependent flows in 2016 to 2017. Throughout the data period, time-dependent flows are most dominant under the Pacific. The total flow is estimated to be mostly toroidal and equatorially symmetric. Regarding diffusion, we suggest that it may explain some fraction of observed SV, and that constraining this fraction is important to accurately estimate the magnitude of flow velocities.

We also conclude that ocean generated SV picked up by satellites and ground observatories is small enough to be neglected when considering core flow on inter-annual and longer timescales, and that the SWARM data is subject to an unknown error source on the azimuthal SV measurements during some northern summers (most notably northern summer 2015). The error source is estimated to not have had any significant impact on the results of this study, however, as the derived flows are unable to fit this feature.

For future research, it may be interesting to investigate the impact of varying truncation degrees, especially those governing the number of modes used. It would also be useful to use our models to predict changes to the length of day; a classic method used to indicate the validity of core flow models. Lastly, it would be interesting to repeat this study, as longer time series of SWARM data become available. For example, we only observe one reversal of time-dependent azimuthal flow at most longitudes and have not yet observed a full ”cycle” of this phenomenon in the period covered by SWARM data.

References

Amit, Hagay and Ulrich Christensen (2008). “Accounting for magnetic diffusion in core flow inversions from geomagnetic secular variation”. In: *Geophysical Journal International* 175.3, pp. 913–924. DOI: https://doi.org/10.1111/j.1365-246X.2008.03948.x.

Amit, Hagay and Peter Olson (2004). “Helical core flow from geomagnetic secular variation”. In: *Physics of the Earth and Planetary Interiors* 147.1, pp. 1–25. DOI: https://doi.org/10.1016/j.pepi.2004.02.006.

Aubert, Julien et al. (2013). “Bottom-up control of geomagnetic secular variation by the Earth’s inner core”. In: *Nature* 502, pp. 219–223. DOI: https://doi.org/10.1038/nature12574.

Barrois, Olivier et al. (2018). “Assimilation of ground and satellite magnetic measurements: inference of core surface magnetic and velocity field changes”. In: *Geophysical Journal International* 215.1, pp. 695–712. DOI: https://doi.org/10.1093/gji/ggy297.

– (2019). “Contributions to the geomagnetic secular variation from a reanalysis of core surface dynamics”. In: *Geophysical Journal International* 211.1, pp. 50–68. DOI: https://doi.org/10.1093/gji/ggx280.

Bloxham, Jeremy and Andrew Jackson (1991). “Fluid flow near the surface of Earth’s outer core”. In: *Reviews of Geophysics*. DOI: https://doi.org/10.1029/90RG02470.

Finlay, Chris et al. (2010). “International Geomagnetic Reference Field: the eleventh generation”. In: *Geophysical Journal International* 183.3, pp. 1216–1230. DOI: https://academic.oup.com/gji/article/183/3/1216/637157?login=true.

– (2020). “The CHAOS-7 geomagnetic field model and observed changes in the South Atlantic Anomaly”. In: *Earth, Planets and Space* 72.156. DOI: https://doi.org/10.1186/s40623-020-01252-9.

Forget, G. et al. (2015). “ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation”. In: *Geoscientific Model Development* 8.10, pp. 3071–3104. DOI: https://doi.org/10.5194/gmd-8-3071-2015.

Gillet, Nicolas et al. (2019). “A reduced stochastic model of core surface dynamics based on geodynamo simulations”. In: *Geophysical Journal International* 219.1, pp. 522–539. DOI: https://doi.org/10.1093/gji/ggz313.

– (2011). “Rationale and geophysical evidence for quasi-geostrophic rapid dynamics within the Earth’s outer core”. In: *Physics of the Earth and Planetary Interiors* 187.3-4. DOI: https://doi.org/10.1016/j.pepi.2011.01.005.

– (2015). “Stochastic forecasting of the geomagnetic field from the COV-OBS.x1 geomagnetic field model, and candidate models for IGRF-12”. In: *Earth Planets and Space* 67.71. DOI: https://earth-planets-space.springeropen.com/articles/10.1186/s40623-015-0225-z.

Grayver, Alexander et al. (2017). “Joint inversion of satellite□detected tidal and magnetospheric signals constrains electrical conductivity and water content of the upper mantle and transition zone”. In: *Geophysical Research Letters* 44.12, pp. 6074–6081. DOI: https://doi.org/10.1002/2017GL073446.

Hammer, Magnus et al. (2021). “Geomagnetic Virtual Observatories: Monitoring geomagnetic secular variation with the Swarm satellites”. In: *Not yet published*.

Jackson, Andrew (1997). “Time-dependency of tangentially geostrophic core surface motions”. In: *Physics of the Earth and Planetary Interiors* 103.3-4, pp. 293–311. DOI: https://doi.org/10.1016/S0031-9201(97)00039-3.

Kloss, Clemens and Chris Finlay (2019). “Time-dependent low-latitude core flow and geomagnetic field acceleration pulses ”. In: *Geophysical Journal International* 217.1, pp. 140–168. DOI: https://academic.oup.com/gji/article/217/1/140/5270734.
Lloyd, D. and D. Gubbins (1990). “Toroidal fluid motion at the top of the Earth’s core”. In: *Geophysical Journal International* 100.3, pp. 455–467. DOI: https://doi.org/10.1111/j.1365-246X.1990.tb00698.x.
Marshall, John et al. (1997). “A finite□volume, incompressible Navier Stokes model for studies of the ocean on parallel computers”. In: *Journal of Geophysical Research* 102.C3, pp. 5753–5766. DOI: https://doi.org/10.1029/96JC02775.
Matzka, J. et al. (2010). “Geomagnetic Observations for Main Field Studies: From Ground to Space”. In: *Space Science Reviews* 155, pp. 29–64. DOI: https://doi.org/10.1007/s11214-010-9693-4.
Olsen, Nils et al. (2010). “Measuring the Earth’s Magnetic Field from Space: Concepts of Past, Present and Future Missions”. In: *Space Science Reviews* 155.1-4, pp. 65–93. DOI: https://orbit.dtu.dk/en/publications/measuring-the-earths-magnetic-field-from-space-concepts-of-past-p.
– (2014). “The CHAOS-4 geomagnetic field model”. In: *Geophysical Journal International* 197.2, pp. 815–827. DOI: https://doi.org/10.1093/gji/ggu033.
– (2013). “The Swarm Satellite Constellation Application and Research Facility (SCARF) and Swarm data products”. In: DOI: https://doi.org/10.5047/eps.2013.07.001.
Pais, Maria and Dominique Jault (2008). “Quasi□geostrophic flows responsible for the secular variation of the Earth’s magnetic field”. In: *Geophysical Journal International* 173.2, pp. 421–443. DOI: https://doi.org/10.1111/j.1365-246X.2008.03741.x.
Schaeffer, Nathanaël (2015). “Boundary conditions in spherical geometry”. In:
Schaeffer, Nathanaël and Maria Pais (2011). “On symmetry and anisotropy of Earth□core flows”. In: *Geophysical Research Letters* 38.10. DOI: https://academic.oup.com/gji/article/217/1/140/5270734.
Schnepf, Neesha et al. (2020). “Can seafloor voltage cables be used to study large-scale circulation? An investigation in the Pacific Ocean”. In: *Ocean Science (in review)*. DOI: https://doi.org/10.5194/os-2019-129.
Tyler, Robert et al. (2017). “Electrical conductivity of the global ocean”. In: *Earth, Planets and Space* 69.156. DOI: https://doi.org/10.1186/s40623-017-0739-7.
Zhang, Chao et al. (2019). “Impacts of Tropical Indian and Atlantic Ocean Warming on the Occurrence of the 2017/2018 La Niña”. In: *Geophysical Research Letters* 46.6, pp. 3435–3445. DOI: https://doi.org/10.1029/2019GL082280.
Zhang, Keke and Xinhao Liao (2017). *Theory and Modelling of Rotating Fluids; Convection, Inertial Waves and Precession*. Cambridge monographs on mechanics series. Cambridge University Press. ISBN: 978-0-521-85009-4.

A Main Script

The following is the main MATLAB script used for the forward and inverse problem in this study. Based on code by C. Kloss.

```
set(0,'defaulttextinterpreter','latex')
set(0,'defaultaxesfontsize',8); %10.8 corresponds roughly to small in Latex 12
    pt
set(0,'defaulttextfontsize',8);

%predefine global variables
global PATH_fig PATH_mov r_surf r_cmb n_sv n_mf n_b kmax Nmax_sym Mmax_sym
    Nmax_asym ...
    Mmax_asym idx_bg M_bg M_fg bg_modes fg_modes sym_modes asym_modes ...
    n_v tab_modes pp theta_tc Tor_coeff Pol_coeff vor

addpath('/r9/cfinl/m/CPT/')
addpath(genpath('/home/galjo/Documents/Thesis/tools'))
addpath(genpath('/home/galjo/Documents/Thesis/modes'))
addpath(genpath('/home/galjo/Documents/Thesis/data'))
PATH_fig = '/home/galjo/Documents/figs_gustav';
PATH_mov = './';

%define radii of Earth's surface and CMB
r_surf = 6371.2;
r_cmb = 3485;
theta_tc = 30;  % degree co-latitude of tangent cylinder
n_sv = 16;  % maximum degree of SV
n_mf = 14;  % maximum degree of large scale resolved MF
n_b = 30; % maximum degree of main field including unresolved small scales
rad = pi/180;

%load modes
tab_modes = [];
load('zon10_symN10M0M20_asymN10M0M20.mat');
% load('modes/zon30_symN10M0M20_asymN10M0M20.mat');
% load('modes/zon20_symN5M0M20_asymN5M0M20.mat');
M = [Tor_coeff;Pol_coeff];

%load vorticity
vor = [];
load('vorticity_diag_zon10_symN10M0M20_asymN10M0M20.mat');

%define set of inertial modes and load them in (incl. axisymmetric)
kmax = 10;
Nmax_sym = 10;
Mmax_sym = 20;
Nmax_asym = 10;
Mmax_asym = 20;
sym_modes = 2*Nmax_sym*Mmax_sym*(Nmax_sym+1)+(Nmax_sym-1)*(Nmax_sym);
asym_modes = 2*Mmax_asym*(Nmax_asym*(Nmax_asym+2)+1)+Nmax_asym*(Nmax_asym+1);
%mode_type = zeros(kmax+sym_modes+asym_modes,1);
%mode_type(1:kmax) = 'z'; mode_type(kmax+1:kmax+sym_modes) = 's'; mode_type(
    kmax+sym_modes+1:kmax+sym_modes+asym_modes) = 'a';

reply_bg = input('Choose time-dependent modes G+NG/[G+NG+EA]: ','s');
if isempty(reply_bg); reply_bg = 'G+NG+EA'; end;
```

```
switch reply_bg
    case 'G+NG'
        %extract background modes
        col_qg = repmat(4*(1:Nmax_sym),Mmax_sym,1);
        col_qg(end,:) = col_qg(end,:)+2*(1:Nmax_sym);
        col_qg = cumsum(col_qg(:));
        col_qg = [0;col_qg(1:end-1)]+kmax+1;
        idx_bg = true(size(M,2),1);
        idx_bg(1:kmax) = false(kmax,1); idx_bg(col_qg) = false(size(col_qg));
            idx_bg(col_qg+1) = false(size(col_qg));

    case 'G+NG+EA'
        col_qg = repmat(4*(1:Nmax_sym),Mmax_sym,1);
        col_qg(end,:) = col_qg(end,:)+2*(1:Nmax_sym);
        col_qg = cumsum(col_qg(:));
        col_qg = [0;col_qg(1:end-1)]+kmax+1;
        sym_tdep_modes = length(col_qg);

        col_asym = repmat(2*(2*(0:Nmax_asym)+1),Mmax_asym,1);
        col_asym(end,:) = col_asym(end,:)+2*(1:Nmax_asym+1);
        col_asym = cumsum(col_asym(:));
        col_asym = [0;col_asym(1:end-1)]+kmax+sym_modes+1;
        asym_tdep_modes = length(col_asym);

        idx_bg = true(size(M,2),1);
        idx_bg(1:kmax) = false(kmax,1);
        idx_bg(col_qg) = false(size(col_qg)); idx_bg(col_qg+1) = false(size(
            col_qg));
        idx_bg(col_asym) = false(size(col_asym)); idx_bg(col_asym+1) = false(
            size(col_asym));

end
clear reply_bg col_qg col_asym

M_bg = M(:,idx_bg);  % background (time static) modes
M_fg = M(:,~idx_bg);  % rest are time-dep modes
bg_modes = size(M_bg,2);
fg_modes = size(M_fg,2);

%load MF and SV from CHAOS6
pp = [];
CHAOS_filepath = '/home/galjo/Documents/coreflow-source/data/Gustav/Downloads/
    CHAOS-7.2.mat';
load(CHAOS_filepath);


%% load observation data

global  theta_GrObs phi_GrObs r_GrObs ...
    num_of_VO_sw r_VO_sw theta_VO_sw phi_VO_sw ...
    codes_GrObs time_GrObs dBr_GrObs dBt_GrObs dBp_GrObs time_VO_sw...
    dBr_VO_sw dBt_VO_sw dBp_VO_sw num_of_GrObs var_VO_sw var_GrObs

%load VO data
%data_VO_ch = load('/home/magdh/phd/2019/01_VO_timeseries/VO_ver0117/
    VO_CHAMP_SV.0117');
data_cdf = cdfread('/home/galjo/Documents/coreflow-source/data/Gustav/
    Downloads/SW_OPER_VOBS_4M_2__20140301T000000_20200301T000000_0101.cdf','
    CombineRecords', 1, 'variable', {'Timestamp','Latitude','Longitude','
    Radius','B_OB','sigma_OB','B_CF','sigma_CF','Timestamp_SV', 'B_SV','
    sigma_SV'});

```

```
Timestamp = [(todatenum(data_cdf{1})-730486)./365.25 + 2000];
Latitude = squeeze(double(data_cdf{2}));
Longitude = squeeze(double(data_cdf{3}));
Radius = squeeze(double(data_cdf{4}))./1000;
B_OB = squeeze(double(data_cdf{5}));
sigma_OB = squeeze(double(data_cdf{6}));
B_CF = squeeze(double(data_cdf{7}));
sigma_CF = squeeze(double(data_cdf{8}));
Timestamp_SV = [(todatenum(data_cdf{9})-730486)./365.25 + 2000];
B_SV = squeeze(double(data_cdf{10}));
sigma_SV = squeeze(double(data_cdf{11}));
index_nan = find(Timestamp_SV==Timestamp_SV(1));
Timestamp_SV(index_nan)=nan;
Colatitude = 90-Latitude;

data_VO_sw = [Colatitude Longitude Timestamp_SV Radius B_SV(:,1) B_SV(:,2)
    B_SV(:,3)];
%num_of_VO_ch = 300;  % number of CHAMP VOs
num_of_VO_sw = 300;  % number of Swarm VOs
%data_VO_ch(data_VO_ch==99999) = NaN;  % replace tags of invalid data
data_VO_sw(data_VO_sw==99999) = NaN;

%champ
%theta_VO_ch = data_VO_ch(:,1);  % 0:180 degree
%phi_VO_ch = data_VO_ch(:,2);  % -180:180 degree
%time_VO_ch = data_VO_ch(:,3);  % years
%r_VO_ch = data_VO_ch(:,4);  % km
%dBr_VO_ch = data_VO_ch(:,5);  % radial SV nT/yr
%dBt_VO_ch = data_VO_ch(:,6);  % theta SV nT/yr
%dBp_VO_ch = data_VO_ch(:,7);  % phi SV nT/yr
%cov_VO_ch = load('/home/magdh/phd/2019/01_VO_timeseries/VO_ver0117/
    VO_SV_CHAMP_COV_diag.0117');
%var_VO_ch = reshape(diag(cov_VO_ch),num_of_VO_ch,3); % error of dBr (col 1),
    dBt (col 2), dBp (col 3)

%swarm
theta_VO_sw = data_VO_sw(:,1);  % 0:180 degree
phi_VO_sw = data_VO_sw(:,2);  % -180:180 degree
time_VO_sw = data_VO_sw(:,3);  % years
r_VO_sw = data_VO_sw(:,4);  % km
dBr_VO_sw = data_VO_sw(:,5);  % radial SV nT/yr
dBt_VO_sw = data_VO_sw(:,6);  % theta SV nT/yr
dBp_VO_sw = data_VO_sw(:,7);  % phi SV nT/yr
%cov_VO_sw = load('/home/magdh/phd/2019/01_VO_timeseries/VO_ver0117/
    VO_SV_SWARM_COV_diag.0117');
var_VO_sw = sigma_SV(301:600,:).^2;  % error of dBr (col 1), dBt (col 2), dBp
    (col 3)

clear data_VO_sw cov_VO_sw

%load Ground Obs data
GrObs_data_cdf = cdfread('/home/galjo/Documents/coreflow-source/data/Gustav/
    Downloads/GO_CDF/GObs_4M_19970301T000000_20201101T000000_0101.cdf', '
    CombineRecords', 1, 'variable', {'Timestamp','Latitude', 'Longitude','
    Radius', 'B_OB','sigma_OB','B_CF','sigma_CF','Timestamp_SV', 'B_SV','
    sigma_SV','bias_crust','Obs'});
Timestamp    = [todatenum(GrObs_data_cdf{1}) - 730486];
Timestamp2    = [(todatenum(GrObs_data_cdf{1}) - 730486)]./365.25 +2000; % in
    years
Latitude     = squeeze(double(GrObs_data_cdf{2}));
Longitude    = squeeze(double(GrObs_data_cdf{3}));
Radius       = squeeze(double(GrObs_data_cdf{4}))./1000;
```

```
B_OB         = squeeze(double(GrObs_data_cdf{5}));
sigma_OB     = squeeze(double(GrObs_data_cdf{6}));
B_CF         = squeeze(double(GrObs_data_cdf{7}));
sigma_CF     = squeeze(double(GrObs_data_cdf{8}));
Timestamp_SV = [todatenum(GrObs_data_cdf{9}) - 730486];
Timestamp_SV2 = [todatenum(GrObs_data_cdf{9}) - 730486]./365.25 +2000; % in years
B_SV         = squeeze(double(GrObs_data_cdf{10}));
sigma_SV     = squeeze(double(GrObs_data_cdf{11}));
bias_crust   = squeeze(double(GrObs_data_cdf{12}));
Obs = GrObs_data_cdf{13};

Colatitude = 90-Latitude;

data_GrObs = [Colatitude Longitude Timestamp_SV2 Radius B_SV(:,1) B_SV(:,2) B_SV(:,3)];
data_GrObs(data_GrObs==99999)=NaN;

codes_GrObs = Obs;  % alphanumeric 4 digits
theta_GrObs = data_GrObs(:,1);  % 0:180 degree
phi_GrObs = data_GrObs(:,2);  % 0:360 degree
phi_GrObs(phi_GrObs > 180) = phi_GrObs(phi_GrObs > 180)-360;  % -180:180 degree
time_GrObs = data_GrObs(:,3);  % years
r_GrObs = data_GrObs(:,4);  % km
dBr_GrObs = data_GrObs(:,5);
dBt_GrObs = data_GrObs(:,6);
dBp_GrObs = data_GrObs(:,7);

num_of_GrObs = find(codes_GrObs(:,1) == codes_GrObs(1,1) & codes_GrObs(:,2) == codes_GrObs(1,2) & codes_GrObs(:,3) == codes_GrObs(1,3) & codes_GrObs(:,4) == codes_GrObs(1,4));
num_of_GrObs = num_of_GrObs(2)-1;  % number of ground observatories

%cov_GrObs = load('home/galjo/Documents/coreflow-source/data/Gustav/Downloads/GO_V33_4monthly/GR_OBS_RMM_SV_COV_V33_4month.10');
var_GrObs = sigma_SV(11071:11071+205,:).^2; %error of dBr (col 1), dBt (col 2), dBp (col 3)

clear fid temp_in cov_GrObs

%% Combine data sets




%set global variables
global time_step time theta_data phi_data r_data pred_chaos time_data ...
    d var_d comp_list obs_type timeLOD pred_chaos_sa ...
    G_LOD d_LOD obsLOD num_data_total num_data_VO num_data_GrObs ...
    num_data_r num_data_theta num_data_phi num_data_VO_r num_data_VO_theta ...
    num_data_VO_phi num_data_GrObs_r num_data_GrObs_theta num_data_GrObs_phi W_d ...
    codes_GrObs_final G_gauss2spacetime

time_step = 4/12; %4 month steps

time = [2014.7 2015 2015.4 2015.7 2016 2016.4 2016.7 2017 2017.4 2017.7 2018 2018.4 2018.7 2019 2019.4 2019.7]';
%time = ((2000+8/12):time_step:(2018+4/12))';

theta_data = [];
```

```
phi_data = [];
r_data = [];
pred_chaos = [];
pred_chaos_sa = [];
time_data = [];
d = [];
var_d = [];
comp_list = [];
obs_type = [];
codes_GrObs_final = [];
G_gauss2spacetime = [];

%rework_timestamps; %TEMPORARY: Rework GrObs timestamps to match sw timestamps

for k=1:length(time)

    fprintf('Data: working on time %s\n', my_datestr(time(k)));

    %idx_data_ch = round(time_VO_ch,3) == round(time(k),3);
    idx_data_sw = round(time_VO_sw,1) == round(time(k),1);
    idx_data_GrObs = round(time_GrObs,1) == round(time(k),1);

    theta_new = [theta_VO_sw(idx_data_sw);theta_GrObs(idx_data_GrObs)];
    phi_new = [phi_VO_sw(idx_data_sw);phi_GrObs(idx_data_GrObs)];
    r_new = [r_VO_sw(idx_data_sw);r_GrObs(idx_data_GrObs)];
    time_new = [time_VO_sw(idx_data_sw);time_GrObs(idx_data_GrObs)];
    obs_type_new = [repmat('v',sum(idx_data_sw),1);repmat('g',sum(
        idx_data_GrObs),1)];
    codes_GrObs_new = codes_GrObs(idx_data_GrObs,:);
    codes_GrObs_theta_new = theta_GrObs(idx_data_GrObs);

    d_new_r = [dBr_VO_sw(idx_data_sw);dBr_GrObs(idx_data_GrObs)];
    d_new_theta = [dBt_VO_sw(idx_data_sw);dBt_GrObs(idx_data_GrObs)];
    d_new_phi = [dBp_VO_sw(idx_data_sw);dBp_GrObs(idx_data_GrObs)];
    codes_GrObs_d_new_r = dBr_GrObs(idx_data_GrObs);
    codes_GrObs_d_new_theta = dBt_GrObs(idx_data_GrObs);
    codes_GrObs_d_new_phi = dBp_GrObs(idx_data_GrObs);


    %find line numbers that correspond to observatories with numbers
    %between first and last one (1:num_of_VO) or (1:num_of_GrObs)
    %line_ch = mod(find(idx_data_ch),num_of_VO_ch);
    %line_ch(line_ch==0) = num_of_VO_ch; %last one is num_of_VO instead 0
    line_sw = mod(find(idx_data_sw),num_of_VO_sw);
    line_sw(line_sw==0) = num_of_VO_sw;
    line_GrObs = mod(find(idx_data_GrObs),num_of_GrObs);
    line_GrObs(line_GrObs==0) = num_of_GrObs;

    %small repair to make script function with new variance data
    %var_VO_sw_temp = var_VO_sw(k*num_of_VO_sw-num_of_VO_sw+1:k*num_of_VO_sw
        ,:);

    var_d_new_r = [var_VO_sw(line_sw,1);var_GrObs(line_GrObs,1)];
    var_d_new_theta = [var_VO_sw(line_sw,2);var_GrObs(line_GrObs,2)];
    var_d_new_phi = [var_VO_sw(line_sw,3);var_GrObs(line_GrObs,3)];

    if isequal(size(theta_new,1),size(phi_new,1),size(r_new,1),size(d_new_r,1)
        ,size(d_new_theta,1),size(d_new_phi,1),size(var_d_new_r,1),size(
        var_d_new_theta,1),size(var_d_new_phi,1),size(time_new,1),size(
        obs_type_new,1))==0
        error(['Sizes don''t match in ',num2str(time(k)),', check input data
            format!'])
```

```
    end

    %exclude polar region, so only equatorial region
    idx_eq = (theta_new >= theta_tc & theta_new <= (180-theta_tc));
    phi_new = phi_new(idx_eq);
    theta_new = theta_new(idx_eq);
    r_new = r_new(idx_eq);
    time_new = time_new(idx_eq);
    d_new_r = d_new_r(idx_eq);
    var_d_new_r = var_d_new_r(idx_eq);
    d_new_theta = d_new_theta(idx_eq);
    var_d_new_theta = var_d_new_theta(idx_eq);
    d_new_phi = d_new_phi(idx_eq);
    var_d_new_phi = var_d_new_phi(idx_eq);
    obs_type_new = obs_type_new(idx_eq);
    codes_GrObs_idx_eq = (codes_GrObs_theta_new >= theta_tc &
        codes_GrObs_theta_new <= (180-theta_tc));
    codes_GrObs_new = codes_GrObs(codes_GrObs_idx_eq,:);
    codes_GrObs_d_new_r = codes_GrObs_d_new_r(codes_GrObs_idx_eq);
    codes_GrObs_d_new_theta = codes_GrObs_d_new_theta(codes_GrObs_idx_eq);
    codes_GrObs_d_new_phi = codes_GrObs_d_new_phi(codes_GrObs_idx_eq);

    %find not NaNs: useful data
    idx_nonan_r = ~isnan(d_new_r);
    idx_nonan_theta = ~isnan(d_new_theta);
    idx_nonan_phi = ~isnan(d_new_phi);
    codes_GrObs_idx_nonan_r = ~isnan(codes_GrObs_d_new_r);
    codes_GrObs_idx_nonan_theta = ~isnan(codes_GrObs_d_new_theta);
    codes_GrObs_idx_nonan_phi = ~isnan(codes_GrObs_d_new_phi);

    %codes_GrObs_idx_nonan_r = codes_GrObs_d_new_r==codes_GrObs_d_new_r;
    %codes_GrObs_idx_nonan_theta = codes_GrObs_d_new_theta==
        codes_GrObs_d_new_theta;
    %codes_GrObs_idx_nonan_phi = codes_GrObs_d_new_phi==codes_GrObs_d_new_phi;

    if sum([idx_nonan_r;idx_nonan_theta;idx_nonan_phi])==0
        time(k) = NaN;
        continue
    else
        %produce matrix to go from spherical harmonics to grid on CMB
        [G_gauss2grid_r,G_gauss2grid_theta,G_gauss2grid_phi] = design_SHA(
            r_new/r_surf,theta_new*rad,phi_new*rad,n_sv);
        G_gauss2spacetime_new = [G_gauss2grid_r(idx_nonan_r,:);
            G_gauss2grid_theta(idx_nonan_theta,:);G_gauss2grid_phi(
            idx_nonan_phi,:)];
        G_gauss2spacetime = blkdiag(G_gauss2spacetime,G_gauss2spacetime_new);
    end

    g_sv_chaos = fnval(jd2000(time(k), 1, 1), fnder(pp, 1))*365.25; %SV
    g_sv_chaos = g_sv_chaos(1:n_sv*(n_sv+2),:);
    g_sa_chaos = fnval(jd2000(time(k), 1, 1), fnder(pp, 2))*365.25^2; %SV
    g_sa_chaos = g_sa_chaos(1:n_sv*(n_sv+2),:);
    g_sa_chaos(100:end) = 0;  % set coefficient of n>=10 to zero

    %extend vectors
    comp_list = [comp_list;repmat('r',sum(idx_nonan_r),1);repmat('t',sum(
        idx_nonan_theta),1);repmat('p',sum(idx_nonan_phi),1)];
    theta_data = [theta_data;theta_new(idx_nonan_r);theta_new(idx_nonan_theta)
        ;theta_new(idx_nonan_phi)];
    phi_data = [phi_data;phi_new(idx_nonan_r);phi_new(idx_nonan_theta);phi_new
        (idx_nonan_phi)];
    r_data = [r_data;r_new(idx_nonan_r);r_new(idx_nonan_theta);r_new(
```

```
        idx_nonan_phi)];
    time_data = [time_data;time_new(idx_nonan_r);time_new(idx_nonan_theta);
        time_new(idx_nonan_phi)];
    d = [d;d_new_r(idx_nonan_r);d_new_theta(idx_nonan_theta);d_new_phi(
        idx_nonan_phi)];
    pred_chaos = [pred_chaos;G_gauss2spacetime_new*g_sv_chaos];
    pred_chaos_sa = [pred_chaos_sa;G_gauss2spacetime_new*g_sa_chaos];
    var_d = [var_d;var_d_new_r(idx_nonan_r);var_d_new_theta(idx_nonan_theta);
        var_d_new_phi(idx_nonan_phi)];
    obs_type = [obs_type;obs_type_new(idx_nonan_r);obs_type_new(
        idx_nonan_theta);obs_type_new(idx_nonan_phi)];
    codes_GrObs_final = [codes_GrObs_final;codes_GrObs_new(
        codes_GrObs_idx_nonan_r,:);codes_GrObs_new(codes_GrObs_idx_nonan_theta
        ,:);codes_GrObs_new(codes_GrObs_idx_nonan_phi,:)];

    %save number of sw and GrObs data as well as r, theta, and phi for data
    %inspection
    comp_list_temp = [repmat('r',sum(idx_nonan_r),1);repmat('t',sum(
        idx_nonan_theta),1);repmat('p',sum(idx_nonan_phi),1)];
    obs_type_temp = [obs_type_new(idx_nonan_r);obs_type_new(idx_nonan_theta);
        obs_type_new(idx_nonan_phi)];

    codes_GrObs_unique1 = codes_GrObs_new(codes_GrObs_idx_nonan_phi,:);
    codes_GrObs_unique2 = codes_GrObs_new(codes_GrObs_idx_nonan_r,:);
    num_data_total(k) = length(comp_list_temp);
    num_data_r(k) = sum(comp_list_temp == 'r');
    num_data_theta(k) = sum(comp_list_temp == 't');
    num_data_phi(k) = sum(comp_list_temp == 'p');
    num_data_VO(k) = sum(obs_type_temp == 'v');
    num_data_VO_temp = obs_type_temp == 'v';
    num_data_VO_r(k) = sum(num_data_VO_temp(comp_list_temp == 'r'));
    num_data_VO_theta(k) = sum(num_data_VO_temp(comp_list_temp == 't'));
    num_data_VO_phi(k) = sum(num_data_VO_temp(comp_list_temp == 'p'));
    num_data_GrObs(k) = sum(obs_type_temp == 'g');
    num_data_GrObs_temp = obs_type_temp == 'g';
    num_data_GrObs_r(k) = sum(num_data_GrObs_temp(comp_list_temp == 'r'));
    num_data_GrObs_theta(k) = sum(num_data_GrObs_temp(comp_list_temp == 't'));
    num_data_GrObs_phi(k) = sum(num_data_GrObs_temp(comp_list_temp == 'p'));


end
clear G_gauss2grid_r G_gauss2grid_theta G_gauss2grid_phi time_new r_new ...
    phi_new theta_new d_new_r d_new_theta d_new_phi var_d_new_r ...
    var_d_new_theta var_d_new_phi idx_nonan_r idx_nonan_theta ...
    idx_nonan_phi idx_eq line_ch line_sw line_GrObs idx_data_ch ...
    idx_data_sw idx_data_GrObs G_gauss2spacetime_new g_sv_chaos obs_type_new;

fprintf('Done.\n')

time(isnan(time)) = [];

idx_time_nlap = nan(size(d));
idx_time_plap = nan(size(d));
i = 1;
while i<=length(d)
   idx = find(comp_list==comp_list(i) & theta_data==theta_data(i) & r_data==
       r_data(i) & phi_data==phi_data(i) & obs_type(:,1)==obs_type(i,1) & (
       round(time_data,1)==round(time_data(i)-time_step,1) | round(time_data
       ,1)==round(time_data(i)+time_step,1)));
   if numel(idx)==2
       idx = sort(idx,'ascend');
       idx_time_nlap(i) = idx(1);
```

```
        idx_time_plap(i) = idx(2);
    elseif numel(idx)==4  % two stations at one location
        idx = sort(idx,'ascend');
        idx_time_nlap(i) = idx(1);
        idx_time_plap(i) = idx(3);
        i = i+1;
        idx_time_nlap(i) = idx(2);
        idx_time_plap(i) = idx(4);
    elseif numel(idx)>4
        error('Too many values found')
    end
    i = i+1;
end
clear idx

weight_choice = input('Weight? huber/[tukey]: ','s');
if isempty(weight_choice); weight_choice = 'tukey'; end;

switch weight_choice
    case 'huber'
        %data weight matrix, fixed with Huber weights from CHAOS-6
        res = abs(pred_chaos-d)./sqrt(var_d);
        huber_factor = input('Huber weights factor? [1.5]: ');
        if isempty(huber_factor); huber_factor = 1.5; end;
        W_d = diag(min(ones(size(res)),huber_factor./res)./var_d); %incl.
            Huber weights
        clear res
    case 'tukey'
        %data weight matrix using fixed Tukey's biweight
        res = abs(pred_chaos-d)./sqrt(var_d);
        W_d = 1./var_d;
        W_d(res<=4.685) = (1-res(res<=4.685).^2/4.685^2).^2.*W_d(res<=4.685);
        W_d(res>4.685) = 0;
        W_d = diag(W_d);

        d2 = zeros(size(d));
        for i=1:length(d2)
            if isnan(idx_time_plap(i))
                d2(i) = 999999;
            else
                d2(i) = (d(idx_time_plap(i))-d(idx_time_nlap(i)))/(2*time_step
                    );
            end
        end

        var_sa = 2*var_d/(2*time_step)^2;
        res = abs(pred_chaos_sa-d2)./sqrt(var_sa);
        W_d2 = 1./var_sa;
        W_d2(res<=4.685) = (1-res(res<=4.685).^2/4.685^2).^2.*W_d2(res<=4.685)
            ;
        W_d2(res>4.685) = 0;
        W_d2 = diag(W_d2);  % needed for statistics and plots

        clear res
end

%read in LOD data and interpolate
data = dlmread('LOD_noAAM_notides_1yr_1962.5-2016.5.dat');
timeLOD = data(:,1);
obsLOD = data(:,2); %subtract trend (very small)?: -1.4e-2*(1:length(timeLOD))
    ';
d_LOD = spline(timeLOD,obsLOD,time);
```

```
d_LOD(time>timeLOD(end) | time<timeLOD(1)) = NaN;
d_LOD = d_LOD-mean(d_LOD,'omitnan');
sigma2_LOD = 0.24^2; % used 1e-2 before
G_LOD = [sparse(length(time),bg_modes),1.138*M_fg(1,1)*kron(eye(length(time))
    -1/length(time)*ones(length(time),length(time)),[1 sparse(1,fg_modes-1)]),
    sparse(length(time),length(time)*n_sv*(n_sv+2))];
clear data

%% Define covariance of the small-scale magnetic field

para = fit_spectrum(CHAOS_filepath);

%correlation function
tau0 = @(n) para(3)*exp(para(4)*n); %characteristic time in yrs
rho = @(t1,t2,n) (1+sqrt(3)*abs(t1-t2)./tau0(n)).*exp(-sqrt(3)*abs(t1-t2)./
    tau0(n));

%mf variance fit to spectrum
sigma2 = @(n) para(1)*exp(para(2)*n)./((2*n+1).*(n+1));

%degrees to be used for extending spectrum
num_of_extspec = n_b*(n_b+2)-n_mf*(n_mf+2);
n_extspec = zeros(num_of_extspec,1);
mm = 0;
for n=(n_mf+1):n_b
    n_extspec(mm+(1:2*n+1)) = n*ones(2*n+1,1);
    mm = mm+2*n+1;
end
clear mm

%auto-covariance of unresolved magnetic field at single epoch (t1-t2 = 0)
Cov_ss = kron(eye(length(time)),diag(sigma2(n_extspec)));

%off-diagonal elements
for i=2:length(time)
    for j=1:(i-1)
        D = diag(sigma2(n_extspec).*rho(time(i),time(j),n_extspec));
        Cov_ss((i-1)*num_of_extspec+(1:num_of_extspec),(j-1)*num_of_extspec
            +(1:num_of_extspec)) = D;
        Cov_ss((j-1)*num_of_extspec+(1:num_of_extspec),(i-1)*num_of_extspec
            +(1:num_of_extspec)) = D;
    end
end
clear D n_extspec num_of_extspec para

%% Create forward problem with CHAOS6 (May take 30 minutes)

%set global variables
global H_grid H_a diffusion_reg

%diffusion_reg = 'No diffusion';
diffusion_reg = 'Include diffusion';

H_a = zeros(length(time)*n_sv*(n_sv+2),bg_modes+length(time)*fg_modes);
for k = 1:length(time)

   fprintf('Design: working on %s\n', my_datestr(time(k)))
%    fprintf('Design: working on %s\n', time(k))

    %compute MF and SV at specific time
    g_mf_chaos = fnval(jd2000(time(k), 1, 1), pp, 0); %MF
    g_mf_chaos = g_mf_chaos(1:n_mf*(n_mf+2));
```

```

    %compute matrix for induction equation
    A = SV_synthesis(n_mf,n_v,n_sv,g_mf_chaos);

    H_a((k-1)*n_sv*(n_sv+2)+(1:n_sv*(n_sv+2)),1:bg_modes) = A*M_bg;
    H_a((k-1)*n_sv*(n_sv+2)+(1:n_sv*(n_sv+2)),bg_modes+(k-1)*fg_modes+(1:
        fg_modes)) = A*M_fg;
end
fprintf('Done.\n')

switch diffusion_reg
    case 'No diffusion'
        H_grid = [G_gauss2spacetime*H_a G_gauss2spacetime]; %include small-
            scale error: augmented state approach
    case 'Include diffusion'
        H_grid = [G_gauss2spacetime*H_a G_gauss2spacetime G_gauss2spacetime];

end

%save weighted square of system matrix
H2 = sparse(H_grid'*W_d*H_grid);
d2 = H_grid'*W_d*d;

H_grid = sparse(H_grid);
H_a = sparse(H_a);

%clear G_gauss2spacetime A g_mf_chaos

%% INVERSION SETTING

%set global variables
global all_m m space_reg time_reg LOD_input reg_measure misfit_measure ...
    lambda_exp lambda2 lambda_t lambda_sym a_fg a_bg e idx_m lambda_g
        lambda_asym ...
    misfit_VO_r misfit_VO_theta misfit_VO_phi misfit_GrObs_r
        misfit_GrObs_theta ...
    misfit_GrObs_phi

%predefine arrays for L-curve output
% lambda_exp = [-3 -0.378 -0.084 0.1 0.276 0.4 0.58 0.80 1.348 2];
lambda_exp = 0.8;
reg_measure = zeros(size(lambda_exp));
misfit_measure = reg_measure;
all_m = NaN(size(H_grid,2),length(lambda_exp));
idx_m = [];
e = [];

factor = 1.5 %increase all reg params (except lambda_Z_t) with same factor
lambda2 = 0.12;
lambda_t = factor*1.5e3
lambda_g = 1;
lambda_asym = 0.85*62.5
lambda_sym = 1.25;
switch diffusion_reg
    case 'Include diffusion'
        lambda_Z_t = 1.1e4
end

%initial values for non-linear solver
eps = 1e-8; %for small values, norm tends to Lp-norm, no need to change
Niter_max = 25; %maximum number of iterations
Niter_min = 2; %mimimum number of iterations
```

```
delta_min = 1e-3; %error of model
ratio_misfit_min = 1e-2; %mimimum rel. change of misfit
ratio_reg_a_min = 1e-2; %minimum rel. change of regularization on modes
ratio_reg_e_min = 1e-2; %minimum rel. change of regularization on small-scale error

% space_reg = 'L1_allvorticity';
% space_reg = 'L2_all';  % model_2
% space_reg = 'L1_allenergy';
% space_reg = 'L1_nonGvorticity_Genergy';
% space_reg = 'L1_nonGvorticity_L2_Genergy';
% space_reg = 'L1_GNGvorticity_L2_background';
% space_reg = 'L1_nonGvorticity_L1_Gvorticity';  % model_1, model_3
% space_reg = 'L1_nonGvorticity_L1_Gvorticity_heavyasym';  % model_4
  space_reg = 'L1_noAsymVorticity'; %NEW SCHEME model1

% time_reg = 'L2_correlation';
  time_reg = 'L2_firstdiff';
% time_reg = 'L1_seconddiff';

LOD_input = 'off';
%LOD_input = 'on';

%% DO INVERSION

global W_e

fileID = fopen('solver_info.dat','a');
fprintf(fileID,'--------------- %s ---------------\n',datestr(now,'dd-mmm-yyyy HH:MM'));

fileID2 = fopen('solver_results.dat','a');
fprintf(fileID,'Flow inversion of real data from VOs and GOs (ca. %i points per epoch).\n',num_of_GrObs+num_of_VO_sw);

switch space_reg
    case 'L1_allvorticity'
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all modes.\n');
    case 'L2_all'
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        n_extspec = zeros(n_v*(n_v+2),1);
        mm = 0;
        for n = 1:n_v
            n_extspec(mm+(1:2*n+1)) = n^3*ones(2*n+1,1);
            mm = mm+2*n+1;
        end
        n_extspec = repmat(n_extspec,2,1);

        % D = diag(n_extspec);
        % W_a = blkdiag(length(time)*M_bg'*D*M_bg,kron(speye(length(time)), M_fg'*D*M_fg));
        % W_a(1:bg_modes,(bg_modes+1):end) = repmat(M_bg'*D*M_fg,1,length(time));
        % W_a((bg_modes+1):end,1:bg_modes) = W_a(1:bg_modes,(bg_modes+1):end)';

        W_a_fg = kron(speye(length(time)),M_fg'*diag(n_extspec)*M_fg);
        W_a_bg = M_bg'*diag(n_extspec)*M_bg;
        clear mm n_extspec
        fprintf(fileID,'Spatial regularization: Gillet-type L2 on toroidal-
```

```
            poloidal spectrum of flow, scales with (degree)^3\n');
    case 'L1_allenergy'
        fprintf(fileID,'Spatial regularization: L1 on energy of all modes.\n');
    case 'L1_nonGvorticity_Genergy'
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all non-geostrophic modes and L1 on energy of geostrophic polynomials.\n');
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        vor_fg = repmat(vor_fg(kmax+1:end),1,length(time));
    case 'L1_nonGvorticity_L2_Genergy'
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all non-geostrophic modes and L2 on energy of geostrophic polynomials.\n');
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        vor_fg = repmat(vor_fg(kmax+1:end),1,length(time));
        W_a_fg_g = ones(kmax,length(time));
    case 'L1_GNGvorticity_L2_background'
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all NG modes and geostrophic modes, and L2 on amplitude of background modes.\n');
        vor_fg = vor(~idx_bg);
        vor_fg = repmat(vor_fg,1,length(time));
    case 'L1_nonGvorticity_L1_Gvorticity'
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        vor_fg_g = repmat(vor_fg(1:kmax),1,length(time));
        vor_fg_ng = repmat(vor_fg(kmax+1:end),1,length(time));
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all modes but two parameters for non-geostrophic modes and geostrophic mode.\n');
    case 'L1_nonGvorticity_L1_Gvorticity_heavyasym'
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        vor_fg_g = repmat(vor_fg(1:kmax),1,length(time));
        vor_fg_sym = repmat(vor_fg(kmax+(1:2*sym_tdep_modes)),1,length(time));
        vor_fg_asym = repmat(vor_fg(kmax+2*sym_tdep_modes+(1:2*asym_tdep_modes)),1,length(time));
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all modes but two parameters for non-geostrophic modes and geostrophic mode while heavier damping time-dependent asymmetric modes.\n');
    case 'L1_noAsymVorticity'
        vor_bg = vor(idx_bg);
        vor_fg = vor(~idx_bg);
        vor_fg_g = repmat(vor_fg(1:kmax),1,length(time));
        vor_fg_sym = repmat(vor_fg(kmax+(1:2*sym_tdep_modes)),1,length(time));
        vor_fg_asym = repmat(vor_fg(kmax+2*sym_tdep_modes+(1:2*asym_tdep_modes)),1,length(time));
        fprintf(fileID,'Spatial regularization: L1 on vorticity of all modes except time-dependent asymmetric.\n');
end

switch time_reg
    case 'L2_correlation'
        %correlation function for correlation in time
        tau_u = 100; %correlation time in yrs
        rho_u = @(t1,t2) exp(-abs(t1-t2)/tau_u);
        [t1,t2] = meshgrid(time,time);
        Cor_t = rho_u(t1,t2);
        W_t = blkdiag(sparse(bg_modes,bg_modes),kron(inv(Cor_t),speye(fg_modes
```

```
            )),sparse(length(time)*n_sv*(n_sv+2),length(time)*n_sv*(n_sv+2)));
        fprintf(fileID,'Temporal regularization: L2 using correlation function
            .\n');
    case 'L2_firstdiff'
        %L2 on first time-difference of flow

        switch diffusion_reg
            case 'No diffusion'
                D1 = kron(spdiags([-ones(length(time),1) ones(length(time),1)
                    ],[0 1],length(time)-1,length(time)),speye(fg_modes));
                W_t = D1'*D1; clear D1;
                W_t = blkdiag(sparse(bg_modes,bg_modes),W_t,sparse(length(time
                    )*n_sv*(n_sv+2),length(time)*n_sv*(n_sv+2)));
                fprintf(fileID,'Temporal regularization: L2 on first
                    difference.\n');
            case 'Include diffusion'
                D1 = kron(spdiags([-ones(length(time),1) ones(length(time),1)
                    ],[0 1],length(time)-1,length(time)),speye(fg_modes));
                W_t = D1'*D1; clear D1;
                D1 = kron(spdiags([-ones(length(time),1) ones(length(time),1)
                    ],[0 1],length(time)-1,length(time)),speye(n_sv*(n_sv+2)))
                    ;
                W_Z_t = D1'*D1; clear D1;
                W_t = blkdiag(sparse(bg_modes,bg_modes),lambda_t*W_t,sparse(
                    length(time)*n_sv*(n_sv+2),length(time)*n_sv*(n_sv+2)),
                    lambda_Z_t*W_Z_t);
                fprintf(fileID,'Temporal regularization: L2 on first
                    difference.\n');
        end
    case 'L1_seconddiff'
        %second difference for correlation in time
        D2 = spdiags([ones(length(time),1) -2*ones(length(time),1) ones(length
            (time),1)],[0 1 2],length(time)-2,length(time));
        D2 = kron(D2,speye(fg_modes));
        fprintf(fileID,'Temporal regularization: L1 on second difference.\n');
end

switch LOD_input
    case 'off'
        fprintf(fileID,'LOD is not included as data.\n');
        d3 = zeros(size(d2));
        G2_LOD = sparse(size(H2,2),size(H2,2));
    case 'on'
        fprintf(fileID,'LOD is included as data (sigma2 = %.4e).\n',sigma2_LOD
            );
        d3 = 1/sigma2_LOD*G_LOD'*d_LOD;
        G2_LOD = 1/sigma2_LOD*(G_LOD'*G_LOD);
end

fprintf(fileID,'Epochs from %9.3f to %9.3f in %3.2f-year steps.\n',time(1),
    time(end),time(2)-time(1));
fprintf(fileID,'Modes are: kmax = %i (zonal), Nmax = %i Mmax = %i (symmetric),
     Nmax = %i Mmax = %i (anti-symmetric), incl. axi-symmetric waves.\n',kmax,
    Nmax_sym,Mmax_sym,Nmax_asym,Mmax_asym);
fprintf(fileID,'Choice of data weight: %s\n',weight_choice);

for i=1:length(lambda_exp)

    %L1-Regularization nonlinear solver
     lambda = factor*10^lambda_exp(i);
    % lambda2 = 10^lambda_exp(i);
    % lambda_t = 10^lambda_exp(i);
```

```
    % lambda_g = 10^lambda_exp(i);

    [~,num_sol] = system('grep -c "Starting non-linear solver" solver_info.dat
        ');
    disp([num2str(str2double(num_sol)+1),' Starting non-linear solver with
        lambda = ',num2str(lambda),' and lambda_t = ',num2str(lambda_t)]);
    fprintf(fileID,'\n%i Starting non-linear solver with lambda = %.4e,
        lambda2 = %.4e and lambda_t = %.4e\n',str2double(num_sol)+1,lambda,
        lambda2,lambda_t);

    switch space_reg
        case 'L1_allvorticity'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg = speye(length(time)*fg_modes); %L1 initialization
        case 'L1_allenergy'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg = speye(length(time)*fg_modes); %L1 initialization
        case 'L1_nonGvorticity_Genergy'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg_g = ones(kmax,length(time));
            W_a_fg_ng = ones(fg_modes-kmax,length(time));
            W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e\n',lambda_g);
        case 'L1_nonGvorticity_L2_Genergy'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg_ng = ones(fg_modes-kmax,length(time));
            W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e\n',lambda_g);
        case 'L1_GNGvorticity_L2_background'
            W_a_bg = speye(bg_modes);
            W_a_fg_g = ones(kmax,length(time));
            W_a_fg_ng = ones(fg_modes-kmax,length(time));
            W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e\n',lambda_g);
        case 'L1_nonGvorticity_L1_Gvorticity'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg_g = ones(kmax,length(time));
            W_a_fg_ng = ones(fg_modes-kmax,length(time));
            W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e\n',lambda_g);
        case 'L1_nonGvorticity_L1_Gvorticity_heavyasym'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg_g = ones(kmax,length(time));
            W_a_fg_sym = ones(2*sym_tdep_modes, length(time));
            W_a_fg_asym = ones(2*asym_tdep_modes, length(time));
            W_a_fg = [lambda_g * W_a_fg_g; W_a_fg_sym; lambda_asym *
                W_a_fg_asym];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e together with lambda_asym =
                %.4e\n',lambda_g, lambda_asym);
        case 'L1_noAsymVorticity'
            W_a_bg = speye(bg_modes); %L1 initialization
            W_a_fg_g = ones(kmax,length(time));
```

```
            W_a_fg_sym = ones(2*sym_tdep_modes, length(time));
            W_a_fg_asym = ones(2*asym_tdep_modes, length(time));
            W_a_fg = [lambda_g * W_a_fg_g; lambda_sym*W_a_fg_sym; lambda_asym
                * W_a_fg_asym];
            W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*
                fg_modes); %L1 initialization
            fprintf(fileID,'and lambda_g = %.4e together with lambda_asym =
                %.4e\n',lambda_g, lambda_asym);
    end

    fprintf(fileID,[repmat('-',1,188),'\n%5s',repmat(' %12s',1,14),'\n',repmat
        ('-',1,188),'\n'],'Niter','delta','sv_misfit','ratio','reg_a0','
        ratio_a0','reg_at','ratio_at','reg_e','ratio_e','LOD_misfit','len_LOD
        ','ratio_sym','ratio_tdep','sa_misfit');

    W_e = speye(length(time)*n_sv*(n_sv+2));
    W_a = blkdiag(W_a_bg,lambda2*W_a_fg); %initial matrix for Ekblom measure

    switch time_reg
        case 'L1_seconddiff'
            W_t = blkdiag(sparse(bg_modes,bg_modes),D2'*D2,sparse(length(time)
                *n_sv*(n_sv+2),length(time)*n_sv*(n_sv+2)));
    end

    switch diffusion_reg
        case 'No diffusion'
            R = blkdiag(lambda*W_a,W_e) + lambda_t*W_t; %augmented
                regularization
        case 'Include diffusion'

            Z_r = importdata('Cor_dnm_r_o.dat'); %load dynamo realizations
            Z_r = Z_r';

            trunc = n_sv*(n_sv+2);
            Z_trunc = Z_r(1:trunc,:); %truncate

            %compute self covariance matrix
            Z1 = zeros(size(Z_trunc));
            Zm = mean(Z_trunc,2);
            for j = 1:size(Z_trunc,2)
                Z1(:,j) = Z_trunc(:,j) - Zm;
            end
            Cov_Z = 1 / (size(Z_trunc,2)-1)*(Z1*transpose(Z1));
            Cov_Z = inv(Cov_Z);
            Cov_Z_cell = repmat({Cov_Z},1,length(time));
            W_z = blkdiag(Cov_Z_cell{:});
            R = blkdiag(lambda*W_a,W_e,W_z) + W_t;
    end

    m = full(H2 + G2_LOD + R)\(d2+d3);  % use full because sparse is not
        multithreaded

    %update matrices
    a_bg = m(1:bg_modes);
    a_fg = reshape(m((bg_modes+1):(bg_modes+length(time)*fg_modes)),fg_modes,
        length(time));


    %statistics
    delta = 1;
    res = H_grid*m-d;
    sv_misfit = res'*W_d*res/length(d);
```

```
    reg_a0 = norm(vor_bg.*a_bg,1) / bg_modes;
    reg_at = norm(repmat(vor_fg,length(time),1).*a_fg(:),1) / (length(time)*
        fg_modes);
    switch diffusion_reg
        case 'No diffusion'
            reg_e = (m(size(H_a,2)+1:end)'*W_e*m(size(H_a,2)+1:end)) / (length
                (m)-size(H_a,2));
            LOD_misfit = 1/sigma2_LOD*mean((d_LOD-G_LOD*m).^2,'omitnan');
            len_LOD = mean((G_LOD*m).^2)/mean((d_LOD).^2,'omitnan');
        case 'Include diffusion'
            reg_e = (m(size(H_a,2)+1:size(H_a,2)+size(W_e,2))'*W_e*m(size(H_a
                ,2)+1:size(H_a,2)+size(W_e,2))) / (length(m)-size(H_a,2)-size(
                W_z,2));
            LOD_misfit = 1/sigma2_LOD*mean((d_LOD-G_LOD*m(1:end-size(W_z,2)))
                .^2,'omitnan');
            len_LOD = mean((G_LOD*m(1:end-size(W_z,2))).^2)/mean((d_LOD).^2,'
                omitnan');
    end
    ratio_misfit = 1;
    ratio_reg_a0 = 1;
    ratio_reg_at = 1;
    ratio_reg_e = 1;
    ratio_sym = 1;
    ratio_tdep = 1;
    sa_misfit = rms( (res(idx_time_plap(~isnan(idx_time_plap)))-res(
        idx_time_nlap(~isnan(idx_time_plap))))./sqrt(2*var_d(idx_time_plap(~
        isnan(idx_time_plap)))) )^2;

    fprintf(fileID,'%5i %12.4e %12.4f %12.4e %12.4e %12.4e %12.4e %12.4e %12.4
        e %12.4e %12.4e %12.4e %12.4f %12.4f %12.4f\n',0,delta,sv_misfit,
        ratio_misfit,reg_a0,ratio_reg_a0,reg_at,ratio_reg_at,reg_e,ratio_reg_e
        ,LOD_misfit,len_LOD,ratio_sym,ratio_tdep,sa_misfit);

    Niter = 1;
    while Niter<=Niter_max

            %save old statistics and output
            misfit_old = sv_misfit;
            reg_a0_old = reg_a0;
            reg_at_old = reg_at;
            reg_e_old = reg_e;
            m_old = m;

            x = repmat(M_bg*a_bg+M_fg*mean(a_fg,2),1,length(time));
            Cov_e = smallscale_cov(Cov_ss,x,time,n_b,n_mf,n_v,n_sv);
            W_e = inv(Cov_e);


            switch space_reg
                case 'L1_allvorticity'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2)
                        ,0,bg_modes,bg_modes);
                    W_a_fg = spdiags(vor_fg.^2./sqrt((vor_fg.*a_fg(:)).^2+eps
                        ^2),0,length(time)*fg_modes,length(time)*fg_modes);
                case 'L1_allenergy'
                    W_a_bg = spdiags(1./sqrt(a_bg.^2+eps^2),0,bg_modes,
                        bg_modes);
                    W_a_fg = spdiags(1./sqrt(a_fg(:).^2+eps^2),0,length(time)*
                        fg_modes,length(time)*fg_modes);
                case 'L1_nonGvorticity_Genergy'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2)
                        ,0,bg_modes,bg_modes);
```

```
                    W_a_fg_g = 1./sqrt(a_fg(1:kmax,:).^2+eps^2);
                    W_a_fg_ng = vor_fg.^2./sqrt((vor_fg.*a_fg(kmax+1:end,:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
                case 'L1_nonGvorticity_L2_Genergy'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2),0,bg_modes,bg_modes);
                    W_a_fg_ng = vor_fg.^2./sqrt((vor_fg.*a_fg(kmax+1:end,:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
                case 'L1_GNGvorticity_L2_background'
                    W_a_fg_ng = vor_fg_ng.^2./sqrt((vor_fg_ng.*a_fg(kmax+1:end,:)).^2+eps^2);
                    W_a_fg_g = vor_fg_g.^2./sqrt((vor_fg_g.*a_fg(1:kmax,:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
                case 'L1_nonGvorticity_L1_Gvorticity'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2),0,bg_modes,bg_modes);
                    W_a_fg_g = vor_fg_g.^2./sqrt((vor_fg_g.*a_fg(1:kmax,:)).^2+eps^2);
                    W_a_fg_ng = vor_fg_ng.^2./sqrt((vor_fg_ng.*a_fg(kmax+1:end,:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g;W_a_fg_ng];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
                case 'L1_nonGvorticity_L1_Gvorticity_heavyasym'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2),0,bg_modes,bg_modes);
                    W_a_fg_g = vor_fg_g.^2./sqrt((vor_fg_g.*a_fg(1:kmax,:)).^2+eps^2);
                    W_a_fg_sym = vor_fg_sym.^2./sqrt((vor_fg_sym.*a_fg(kmax+(1:2*sym_tdep_modes),:)).^2+eps^2);
                    W_a_fg_asym = vor_fg_asym.^2./sqrt((vor_fg_asym.*a_fg(kmax+2*sym_tdep_modes+(1:2*asym_tdep_modes),:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g; W_a_fg_sym; lambda_asym*W_a_fg_asym];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
                case 'L1_noAsymVorticity'
                    W_a_bg = spdiags(vor_bg.^2./sqrt((vor_bg.*a_bg).^2+eps^2),0,bg_modes,bg_modes);
                    W_a_fg_g = vor_fg_g.^2./sqrt((vor_fg_g.*a_fg(1:kmax,:)).^2+eps^2);
                    W_a_fg_sym = vor_fg_sym.^2./sqrt((vor_fg_sym.*a_fg(kmax+(1:2*sym_tdep_modes),:)).^2+eps^2);
                    W_a_fg_asym = 1./sqrt((a_fg(kmax+2*sym_tdep_modes+(1:2*asym_tdep_modes),:)).^2+eps^2);
                    W_a_fg = [lambda_g*W_a_fg_g; lambda_sym*W_a_fg_sym; lambda_asym*W_a_fg_asym];
                    W_a_fg = spdiags(W_a_fg(:),0,length(time)*fg_modes,length(time)*fg_modes); %L1 initialization
            end

            W_a =  blkdiag(W_a_bg,lambda2*W_a_fg);

```

```
        switch time_reg
            case 'L1_seconddiff'
                W_t = blkdiag(sparse(bg_modes,bg_modes),D2'*spdiags(1./
                    sqrt((D2*a_fg(:)).^2+eps^2),0,(length(time)-2)*
                    fg_modes,(length(time)-2)*fg_modes)*D2,sparse(length(
                    time)*n_sv*(n_sv+2),length(time)*n_sv*(n_sv+2)));
        end

        switch diffusion_reg
            case 'Include diffusion'
                R = blkdiag(lambda*W_a,W_e,W_z) + W_t;
            case 'No diffusion'
                R = blkdiag(lambda*W_a,W_e) + lambda_t*W_t;
        end

        %update model vector
        m = full(H2 + G2_LOD + R)\(d2+d3);

        %update matrices
        a_bg = m(1:bg_modes);
        a_fg = reshape(m((bg_modes+1):(bg_modes+length(time)*fg_modes)),
            fg_modes,length(time));

        %save model vector in case loop is terminated
        all_m(:,i) = m;

        %model change
        delta = norm(m(1:size(H_a,2))-m_old(1:size(H_a,2),1))/norm(m_old
            (1:size(H_a,2)),1);

        %misfit norm
        res = H_grid*m-d;
        sv_misfit = (res'*W_d*res) / length(d);
        ratio_misfit = abs(sv_misfit-misfit_old)/sv_misfit;

        %flow regularization norm
        reg_a0 = norm(vor_bg.*a_bg,1) / bg_modes;
        reg_at = norm(repmat(vor_fg,length(time),1).*a_fg(:),1) / (length(
            time)*fg_modes);
        ratio_reg_a0 = abs(reg_a0-reg_a0_old)/reg_a0_old;
        ratio_reg_at = abs(reg_at-reg_at_old)/reg_at_old;

        %small-scale error regularization norm
        switch diffusion_reg
            case 'No diffusion'
                reg_e = (m(size(H_a,2)+1:end)'*(Cov_e\m(size(H_a,2)+1:end)
                    )) / (length(m)-size(H_a,2));
            case 'Include diffusion'
                reg_e = (m(size(H_a,2)+1:size(H_a,2)+size(W_z,2))'*(Cov_e\
                    m(size(H_a,2)+1:size(H_a,2)+size(W_z,2)))) / (length(m
                    )-size(H_a,2)-size(W_z,2));
        end
        ratio_reg_e = abs(reg_e-reg_e_old)/reg_e_old;

        %LOD prediction and measure of fit
        switch diffusion_reg
            case 'No diffusion'
                LOD_misfit = 1/sigma2_LOD*mean((d_LOD-G_LOD*m).^2,'omitnan
                    ');
                len_LOD = mean((G_LOD*m).^2)/mean((d_LOD).^2,'omitnan');
            case 'Include diffusion'
                LOD_misfit = 1/sigma2_LOD*mean((d_LOD-G_LOD*m(1:end-size(
```

```
                        W_z,2))).^2,'omitnan');
                    len_LOD = mean((G_LOD*m(1:end-size(W_z))).^2)/mean((d_LOD)
                        .^2,'omitnan');
            end

            %power ratio of symmetric to total flow at surface
            a = zeros(bg_modes+fg_modes,1);
            a(idx_bg) = a_bg;
            a(~idx_bg) = mean(a_fg,2);
            ratio_sym = (powerspec_flow(Tor_coeff(:,1:kmax+sym_modes)*a(1:kmax
                +sym_modes),n_v)...
                         + powerspec_flow(Pol_coeff(:,1:kmax+sym_modes)*a(1:
                             kmax+sym_modes),n_v))...
                         / (powerspec_flow(Tor_coeff*a,n_v) + powerspec_flow(
                             Pol_coeff*a,n_v));
            ratio_tdep = (powerspec_flow(Tor_coeff(:,~idx_bg)*mean(a_fg,2),n_v
                )...
                         + powerspec_flow(Pol_coeff(:,~idx_bg)*mean(a_fg,2),
                             n_v))...
                         / (powerspec_flow(Tor_coeff*a,n_v) + powerspec_flow(
                             Pol_coeff*a,n_v));
            clear a

            %sa normalized misft assuming gaussian error estimates
            res_sa = zeros(size(d));
            for j=1:length(d)
                if isnan(idx_time_plap(j))
                    res_sa(j) = 0;
                else
                    res_sa(j) = (res(idx_time_plap(j))-res(idx_time_nlap(j)))
                        /(2*time_step);
                end
            end
            sa_misfit = (res_sa'*W_d2*res_sa)/sum(res_sa~=0);
            % sa_misfit = rms( (res(idx_time_plap(~isnan(idx_time_plap)))-res(
                idx_time_nlap(~isnan(idx_time_plap))))./sqrt(2*var_d(
                idx_time_plap(~isnan(idx_time_plap)))) )^2;

            disp(['After iteration ',num2str(Niter),', delta is ',num2str(
                delta)]);
            fprintf(fileID,'%5i %12.4e %12.4f %12.4e %12.4e %12.4e %12.4e
                %12.4e %12.4e %12.4e %12.4e %12.4e %12.4f %12.4f %12.4f\n',
                Niter,delta,sv_misfit,ratio_misfit,reg_a0,ratio_reg_a0,reg_at,
                ratio_reg_at,reg_e,ratio_reg_e,LOD_misfit,len_LOD,ratio_sym,
                ratio_tdep,sa_misfit);

            if delta <= delta_min && ratio_misfit <= ratio_misfit_min &&
                ratio_reg_a0 <= ratio_reg_a_min && ratio_reg_at <=
                ratio_reg_a_min && ratio_reg_e <= ratio_reg_e_min && Niter >=
                Niter_min
                break
            end
            Niter = Niter+1;

    end

    plot_hist_residuals_2020;
    %write L-curve output
    fprintf(fileID2,'%.9e ',m);
    fprintf(fileID2,'\n');
    fprintf(fileID,'> solver_results.dat  # Model was saved to file\n');
    misfit_measure(i) = sv_misfit;
```

```
    disp(['misfit = ',num2str(misfit_measure(i))]);
    disp(['misfit for VO r = ',num2str(misfit_VO_r)]);
    disp(['misfit for VO theta = ',num2str(misfit_VO_theta)]);
    disp(['misfit for VO phi = ',num2str(misfit_VO_phi)]);
    disp(['misfit for GrObs r = ',num2str(misfit_GrObs_r)]);
    disp(['misfit for GrObs theta = ',num2str(misfit_GrObs_theta)]);
    disp(['misfit for GrObs phi = ',num2str(misfit_GrObs_phi)]);
    reg_measure(i) = reg_a0;

end

fprintf(fileID,'lambda_exp = [');
fprintf(fileID,'%.9e ',lambda_exp);
fprintf(fileID,']\n');
fprintf(fileID,'misfit_measure = [');
fprintf(fileID,'%.9e ',misfit_measure);
fprintf(fileID,']\n');
fprintf(fileID,'reg_measure = [');
fprintf(fileID,'%.9e ',reg_measure);
fprintf(fileID,']\n\n');
fclose(fileID);
fclose(fileID2);

if size(all_m,2)==1
    %%
    disp('load solution for plotting')

    ini_solution  % make sure "all_m" exists: either created through model run
         or loaded from solver_results.dat

    f = figure();
    f.Units = 'normalized';
    f.OuterPosition = [0.3 0.1 0.6 0.5];
    ax1 = axes('Parent',f,'position',[0.55 0.39  0.4 0.54]);
    ax2 = axes('Parent',f,'position',[0.08 0.39  0.4 0.54]);
    lim = 1.5*max(max(abs(a_fg)));

    make_sliderplot([],[],a_fg(:,1),ax2,lim)
    b = uicontrol('Parent',f,'Style','slider','Position',[81,54,419,23],...
        'value',1, 'min',1, 'max',length(time),'SliderStep', [1/(length(time)
            -1) 0.1]);
    bgcolor = f.Color;
    bl1 = uicontrol('Parent',f,'Style','text','Position',[50,54,23,23],...
        'String','1','BackgroundColor',bgcolor);
    bl2 = uicontrol('Parent',f,'Style','text','Position',[500,54,23,23],...
        'String',num2str(length(time)),'BackgroundColor',bgcolor);
    bl3 = uicontrol('Parent',f,'Style','text','Position',[240,25,100,23],...
        'String','Time','BackgroundColor',bgcolor);
    set(b,'Callback',@(hObject,eventdata) make_sliderplot(hObject,eventdata,
        a_fg(:,round(get(hObject,'Value'))),ax2,lim) )

    plot(ax1,a_bg);
    axis(ax1,[-99 length(a_bg)+99 -1.1*max(abs(a_bg)) 1.1*max(abs(a_bg))])
    title(ax1,'background modes')

    clear bgcolor lim
else
    %%
    plt_width = 15;

    %interpolate curve and find point of maximum curvature
    pp_reg = pchip(log10(misfit_measure),log10(reg_measure));
```

```
    pp_lambda = pchip(log10(misfit_measure),lambda_exp);
    xq_misfit_measure = linspace(min(misfit_measure),max(misfit_measure),100);

    % Plot L-curve (Regularization norm over normalized misfit^2)
    fig=figure();
    fig.Units = 'centimeters';
    fig.Position=[0 5 plt_width 6*plt_width/8];

    dcm_obj = datacursormode(fig);
    set(dcm_obj,'DisplayStyle','Window','Enable','on')
    plot3(xq_misfit_measure,10.^fnval(pp_reg,log10(xq_misfit_measure)),fnval(
        pp_lambda,log10(xq_misfit_measure)))
    hold on
    scatter3(misfit_measure,reg_measure,lambda_exp,36,lambda_exp,'filled','
        DisplayName','L-curve')
    hold off
    view(2)
    grid off
    set(gca,'xscale','log','yscale','log','TicklabelInterpreter','latex')
    xlabel('normalized misfit')
    ylabel('regularization norm of flow')
    clb = colorbar;
    set(clb,'TicklabelInterpreter','latex')
    ylabel(clb,'$\log_{10}(\lambda)$','Interpreter','latex')

    % save_plot('./figures/Lcurve');

end

%% EXTRA: Find separate tdep amplitude vectors for g, sym, asym (needed for
    some plots)

global a_fg_g a_fg_sym a_fg_asym fg_g_modes fg_sym_modes fg_asym_modes ...
    idx_fg_g idx_fg_sym idx_fg_asym mode_type

%g_tdep_modes = size(a_fg,1) - sym_tdep_modes - asym_tdep_modes;

%mode_type = tab_modes(:,5);
%mode_type = char(mode_type);
%mode_type = mode_type(:,1); %vector with code for mode type indices
%idx_fg_asym = mode_type == 'a';
%idx_fg_asym = idx_fg_asym(~idx_bg);
%idx_fg_sym = mode_type == 's';
%idx_fg_sym = idx_fg_sym(~idx_bg);
%idx_fg_g = mode_type == 'z';
%idx_fg_g = idx_fg_g(~idx_bg);

a_fg_g = a_fg(1:size(W_a_fg_g,1),:);
a_fg_sym = a_fg(size(W_a_fg_g,1)+1:size(W_a_fg_g,1)+size(W_a_fg_sym,1),:);
a_fg_asym = a_fg(size(W_a_fg_g,1)+size(W_a_fg_sym,1)+1:size(W_a_fg_g,1)+size(
    W_a_fg_sym,1)+size(W_a_fg_asym,1),:);

mode_type = zeros(sum(~idx_bg),1);
mode_type = num2str(mode_type);
mode_type(1:size(W_a_fg_g,1)) = 'z';
mode_type(size(W_a_fg_g,1)+1:size(W_a_fg_g,1)+size(W_a_fg_sym,1)) = 's';
mode_type(size(W_a_fg_g,1)+size(W_a_fg_sym,1)+1:end) = 'a';

%a_fg_g = reshape(m((bg_modes+1):(bg_modes+length(time)*size(W_a_fg_g,1))),
    size(W_a_fg_g,1),length(time));
%a_fg_sym = reshape(m((bg_modes+numel(a_fg_g)+1):(bg_modes+numel(a_fg_g)+
    length(time)*size(W_a_fg_sym,1))),size(W_a_fg_sym,1),length(time));
```

```
%a_fg_asym = reshape(m((bg_modes+numel(a_fg_g)+numel(a_fg_sym)+1):(bg_modes+
    numel(a_fg_g)+numel(a_fg_sym)+length(time)*size(W_a_fg_asym,1))),size(
    W_a_fg_asym,1),length(time));

fg_g_modes = size(a_fg_g,1);
fg_sym_modes = size(a_fg_sym,1);
fg_asym_modes = size(a_fg_asym,1);

%idx_fg_g = logical([zeros(bg_modes,1); ones(size(a_fg_g,1),1); zeros(size(
    a_fg_sym,1)+size(a_fg_asym,1),1)]);
%idx_fg_g = idx_fg_g(~idx_bg);
%idx_fg_sym = logical([zeros(bg_modes + size(a_fg_g,1),1); ones(size(a_fg_sym
    ,1),1); zeros(size(a_fg_asym,1),1)]);
%idx_fg_sym = idx_fg_sym(~idx_bg);
%idx_fg_asym = logical([zeros(bg_modes + size(a_fg_g,1) + size(a_fg_sym,1),1);
     ones(size(a_fg_asym,1),1)]);
%idx_fg_asym = idx_fg_asym(~idx_bg);
%% EXTRA: Tukey weighted rms for acceleration


res = H_grid*m - d;  % compared to CoreFlo-LL.1
chaos_res = pred_chaos - d;  % compared to CHAOS
res_sa = zeros(size(d));
for j=1:length(d)
    if isnan(idx_time_plap(j))
        res_sa(j) = 0;
    else
        res_sa(j) = (res(idx_time_plap(j))-res(idx_time_nlap(j)))/(2*time_step
            );
    end
end

var_sa = 2*var_d/(2*time_step)^2;
tukey_rms(res_sa(res_sa~=0 & comp_list=='r'), var_sa(res_sa~=0 & comp_list=='r
    '))


%% EXTRA: flow energy in Q+QG time-dependence

a_t = a_fg - repmat(mean(a_fg, 2), 1, length(time));
Tor_coeff_t = Tor_coeff(:, ~idx_bg);
Pol_coeff_t = Pol_coeff(:, ~idx_bg);
sym_modes_t = 1:(kmax+400);

for i = 1:length(time)
    frac_sym_t(i) = powerspec_flow(Tor_coeff_t(:, sym_modes_t)*a_t(sym_modes_t
        , i), n_v) + powerspec_flow(Pol_coeff_t(:, sym_modes_t)*a_t(
        sym_modes_t, i), n_v);
    frac_sym_t_all(i) = (powerspec_flow(Tor_coeff_t*a_t(:, i), n_v) +
        powerspec_flow(Pol_coeff_t*a_t(:, i), n_v));
end

mean(frac_sym_t)/mean(frac_sym_t_all)

%% EXTRA: Diagnostics

mode_type = tab_modes(:,5);
mode_type = char(mode_type);
mode_type = mode_type(:,1); %vector with code for mode type indices
idx_sym_modes = mode_type == 's' | mode_type == 'z';
idx_sym_t_modes = idx_sym_modes(~idx_bg);
idx_sym_bg_modes = idx_sym_modes(idx_bg);
```

```
a_t = a_fg - repmat(mean(a_fg, 2), 1, length(time));
Tor_coeff_t = Tor_coeff(:, ~idx_bg);
Pol_coeff_t = Pol_coeff(:, ~idx_bg);
sym_modes_t = 1:(kmax+400);

Tor_coeff_bg = Tor_coeff(:, idx_bg);
Pol_coeff_bg = Pol_coeff(:, idx_bg);
%modes_bg = (kmax+401):size(Tor_coeff,2);

for i = 1:length(time)
    frac_sym_t(i) = powerspec_flow(Tor_coeff_t(:, sym_modes_t)*a_t(sym_modes_t
        , i), n_v) + powerspec_flow(Pol_coeff_t(:, sym_modes_t)*a_t(
        sym_modes_t, i), n_v);
    frac_sym_t_all(i) = (powerspec_flow(Tor_coeff_t*a_t(:, i), n_v) +
        powerspec_flow(Pol_coeff_t*a_t(:, i), n_v));
end
frac_all_t = mean(frac_sym_t_all);
frac_sym_bg = powerspec_flow(Tor_coeff_bg(:,idx_sym_bg_modes)*a_bg(
    idx_sym_bg_modes), n_v) + powerspec_flow(Pol_coeff_bg(:, idx_sym_bg_modes)
    *a_bg(idx_sym_bg_modes), n_v);
frac_all_sym = mean(frac_sym_t) + frac_sym_bg;
frac_all_bg = powerspec_flow(Tor_coeff(:,idx_bg)*a_bg, n_v) + powerspec_flow(
    Pol_coeff(:,idx_bg)*a_bg, n_v);
frac_all = mean(frac_sym_t_all) + frac_all_bg;

f_S = frac_all_sym/frac_all;
f_S_t = mean(frac_sym_t)/frac_all_t;
f_t = frac_all_t/frac_all;

%f_t = mean(frac_sym_t_all)/frac_total;
```

Pictures/mainscript.m

B Project Plan for ”Core flows inferred from Swarm satellite magnetic data”

Purpose: The purpose of this project is to derive models of flow in the Earth's core using the latest magnetic data from the SWARM satellites and ground observatories. These models will then be analyzed in an effort to better understand the recent geomagnetic changes in e.g. the Pacific.

Applied techniques: The aforementioned models will be created by inversion of the magnetic data, using the induction equation and assuming that the physics of the flow in the Earth's core corresponds to that of a rapidly rotating fluid. The flow will thus be described as a series of modes as described in [Zhang Liao, 2017]. The overall method of the project is similar to that of [Kloss Finlay, 2019], but with a new and improved regularization scheme and using newer SWARM and ground observatory data as input. The project will use the same MATLAB script that was used to obtain the results presented in [Kloss Finlay, 2019]. This script will, however, require some modification to accommodate the new regularization scheme and data. Apart from these changes, this project also aims to produce an inversion scheme that accounts for contribution from magnetic diffusion, which was assumed to be zero in [Kloss Finlay, 2019]. Work on the project will be done in close collaboration with the authors of this paper, Clemens Kloss and Christopher C. Finlay (project supervisor).

Materials: Apart from relevant literature, materials used will include the aforementioned script, the MATLAB software, new magnetic data from the SWARM satellites and ground observatories, and access to DTU Space's HPC-cluster. The latter will be used for the demanding computations that are required. Data and project results will be stored in the HPC-cluster. All materials are already available and will require no further investment.

Project risks: The project is fairly low risk. Using already available data and a method similar to one that has already been used successfully in [Kloss Finlay, 2019], the chances of completing the project and achieving useful results are high. One aspect of the project that may present some difficulties is the inclusion of magnetic diffusion in the inversion scheme. Should this fail, or not produce satisfying results, it is not detrimental to the project, however.

Time plan:

24th Aug - 24th Sept: Review the new geomagnetic data and relevant literature (Olsen et al. 2014, Hammer et al. 2020, Kloss and Finlay, 2019, Zhang and Liao, 2017). Get familiar with the Core flow modelling script and modify it to ingest the latest ground observatory and Swarm satellite magnetic data.

24th Sept - 24th Oct: Perform first full inversion using new ground and satellite magnetic datasets with previous flow parameterization and regularization scheme. Implement new, improved flow regularization scheme. Minimize core flow mode amplitudes for time-dependent equatorial antisymmetric modes (Zhang and Liao, 2017; Kloss and Finlay, 2019). Adjust regularization parameters. Compare new results with those that used in the previous regularization scheme.

24th Oct- 24th Nov: Implement accounting for magnetic diffusion in inversion. Implement the following (i) Simple approach of correction: Amit Christensen (2008) and (ii) co-estimation of diffusion: Barrois et al. (2017;2019).

24th Nov-24th Dec : Finalize model results and interpret recent field changes in the Pacific

24th Dec -24th Jan: Complete writing of thesis.

References:

- [Zhang & Liao, 2017]. Liao, X. & Zhang, K., 2017. Theory of modelling and rotating fluids : convection, intertial waves, and precession. Cambridge University Press.

- [Kloss & Finlay, 2019]. Kloss, C. & Finlay, C.C., 2019. Time-dependent low-latitude core flow and geomagnetic field acceleration pulses.

- [Hammer, et al. 2020]. Hammer, M.D., Cox, G.A., Brown, W.J., Beggan, C.D. & Finlay, C.C., 2020. Geomagnetic Virtual Observatories: Monitoring geomagnetic secular variation with the Swarm satellites.

- [Amit & Christensen, 2008]. Amit, H. & Christensen U.R., 2008. Accounting for magnetic diffusion in core flow inversions from geomagnetic secular variation.

- [Olsen et al. 2014]. Olsen, N., Hulot, G., Lesur, V., Finlay, C.C., Beggan, C., Chulliat, A., Sabaka, T.J., Floberghagen, R., Friis-Christensen, E., Haagmans, R., Kotsiaros, S., Lühr, H., Tøffner-Clausen, L., Vigneron, P., 2014. The Swarm Inertial Field Model for the 2014 geomagnetic field.

- [Barrois et al. 2017]. Barrois, O., Gillet, N., Aubert, J., 2017. Contributions to the geomagnetic secular variation from a reanalysis of core surface dynamics.

- [Barrois et al. 2018]. Barrois, O., Hammer, M., Finlay, C., Martin, Y., Gillet, N., 2018. Assimilation of ground and satellite magnetic measurements: inference of core surface magnetic and velocity field changes.

- [Barrois et al. 2019]. Barrois, O., Gillet, N., Aubert, J., Hammer, M., Finlay, C., Martin, Y., 2019. Erratum: 'Contributions to the geomagnetic secular variation from a reanalysis of core surface dynamics' and 'Assimilation of ground and satellite magnetic measurements: inference of core surface magnetic and velocity field changes'

Abstract

Observed gradual changes in the geomagnetic field, known as secular variation, are believed to be governed by the flow of liquid metal in the outer region of the Earth's core, near the core-mantle boundary. These core flows and the secular variation observed above the Earth's surface are related through the magnetic induction equation. Satellite and ground observatory measurements of secular variation and the induction equation thus present an inverse problem for determining core flows at the core-mantle boundary. Kloss and Finlay 2019 previously presented a method for solving this inverse problem for the period 2000 to 2018, by parametrizing core flow as a series of normal modes of rapidly rotating flow in a spherical container. In this study, we extend their method by further allowing for smaller-scale, equatorially anti-symmetric flows and accounting for likely contributions of magnetic diffusion to the observed secular variation. We implement this modified method with a new regularization scheme for the inverse problem, and by augmenting the model vector to include secular variation due to diffusion. We apply it to SWARM satellite data covering the period 2014-2019. We find that allowing for more small-scale equatorial anti-symmetry (localized equator crossings) and diffusion allows us to estimate flows that well explain the observed secular variation with flows that are similar to, but simpler than, those described by Kloss and Finlay 2019. In particular, we find a predominantly steady, planetary-scale, eccentric gyre of westward flow along with inter-annual reversals of low-latitude azimuthal flow. We conclude that flows with significant local equator crossings and contributions from diffusion provide consistent explanations of the secular variation observed from 2014 to 2019, demonstrating the non-uniqueness of the inverse problem while adding to the evidence for the robustness of the aforementioned flow features.

Technical
University of
Denmark

Centrifugevej, Building 356
2800 Kgs. Lyngby
Tlf. 4525 1700

www.space.dtu.dk