Numerical simulations of neutron star magnetospheres and magnetar bursts

Author: Daniel Tickner

Primary supervisor: Dr. Samuel Lander

Secondary supervisor: Dr. Robert Ferdman

Student ID: 100359822

Date: 28 February 2025

In partial fulfilment of the degree of *Master of Philosophy* at the University of East Anglia, United Kingdom.

Abstract

Magnetars are highly magnetised, slowly rotating varieties of neutron star. They are violent objects observed to emit bursts of radiation and, extremely rarely, giant flares powerful enough to outshine an entire galaxy. The origin of this transient behaviour is thought to be an ultra-strong internal magnetic field which drives starquakes on the crust, displacing the footpoints of its external magnetic field lines. The field lines undergo magnetic reconnection, relaxing to lower-energy configurations and emitting huge bursts of radiation.

We build and present a numerical code to test this hypothesis by subjecting the magnetosphere of a fiducial neutron star to some initial twist and evolving its response over time. We find that our code is well suited to modelling the persistent twists that are expected to commonly occur in neutron star magnetospheres, but more physics is required before we can simulate reconnection and bursting events.

We develop from scratch a bank of numerical techniques including multidimensional numerical integration in curvilinear coordinate systems, high-accuracy first- and second-order finite differencing schemes, approximation of 1D mathematical functions by Chebyshev series on arbitrary domains - not just the standard $x \in [-1, 1]$ - and approximations of 2D angular functions by vector spherical harmonic series. These techniques are quite general and may find application in any numerical simulation.

All source material is made available through the following `GitHub` repository:

https://github.com/DanTickner/Neutron-star-magnetospheres

Contents

1 Introduction

1.1 Physical background and motivation

Giant flares are extremely energetic and rare explosions in the night sky. Just four have been observed as of February 2025 (Mazets et al., 1979, 1999; Hurley et al., 2005; Mereghetti et al., 2024). They can briefly outshine an entire galaxy: typical giant flare gamma-ray luminosity is $\sim 10^{47}$ erg s^{-1} (Kaspi & Beloborodov, 2017), whereas most galaxies have visible-light luminosity $\sim 10^{42} - 10^{45}$ erg s^{-1} (e.g. Jones et al., 2015, Table 2.1). Despite this, they are widely held to originate from **neutron stars**, stellar remnants typically around 10 km in size. In particular, giant flares are thought to be caused by **magnetars**, young and slowly-rotating variants of neutron star that are the most strongly magnetised objects currently known.

The magnetar scenario for giant flares was proposed by Duncan & Thompson (1992). They suggested that the internal magnetic fields of magnetars are strong enough to cause **starquakes**, deforming the crust into which the footpoints of the external magnetic field are anchored. This imparts twists to the external field, and if the twist is large enough then an explosive release of energy occurs (Lynden-Bell & Boily, 1994).

The steady-state (i.e. time-independent) structure of a neutron star magnetosphere was first described qualitatively by Goldreich & Julian (1969). An analytic form, the **pulsar equation**, was obtained by Michel (1973) and Scharlemann & Wagoner (1973) but not solved until Contopoulos et al. (1999).

However, to test the Thompson-Duncan model and simulate bursting events, a steady-state solution does not suffice. In this thesis, we attempt do develop a computer code which evolves a neutron star magnetosphere over time. This will allow us to impart twists to the magnetosphere and describe the response of the system to these twists. Spitkovsky (2006) was the first to obtain a time-dependent solution of a pulsar magnetosphere, and also the first to do so for arbitrary alignment of the rotational and magnetic axes. He did not simulate bursting events, but expressed hope that such models would follow in the future.

1.2 Thesis layout

The thesis is divided into five roughly self-contained Parts. One is not required to read the thesis linearly, but the chosen layout follows a clear path from theory to code development to result.

- In Part I, we provide a literature review detailing the history of study of pulsars, the discovery of giant flares and the prediction of, and evidence for, the magnetar model.
- In Part II, we review the mathematics that we have developed as part of this project. Much of this is original work. Not all results are used in the final code, but we hope that this Part will act as a useful reference guide and that results might find applications in new contexts outside this project.
- In Part III, we discuss the development of the evolutionary model by a `C++` code. We test the implementation of various mathematical methods from Part II along the way, building up piece-by-piece to a full model that we can have confidence in.
- In Part IV, we run the code for three physical models: a stationary magnetic dipole, a rotating dipole (simulating a pulsar) and a stationary dipole subject to a magnetospheric twist (simulating a magnetar). We discuss the performance of the code for each simulation.
- Finally, in Part V, we provide technical information for readers interested in running the code for themselves. This includes installation instructions and a qualitative overview of the evolutionary procedure.

The reader is invited to begin with the Summary in Chapter 30, where we present a brief overview of the aims and outcomes of the project, the current theoretical understanding of neutron star bursts and an analysis of the final simulations. We refer to the relevant sections as we go, aiding navigation of the thesis. It

is in this chapter that we present our final analysis and conclusions. The reader interested solely in an analysis of the final simulations may skip directly to Part IV. The reader interested solely in running the code may skip directly to Chapter 31 for a description of how to run it and use its output.

Many of the results in Part II are presented in a purely mathematical sense, abstracted quite far from neutron stars. This is done to minimise ambiguity and allow application to any other numerical modelling scheme, since the results are quite general. Many of them, for example Chebyshev polynomials, overlap integrals of associated Legendre functions and vector spherical harmonics, are not used in the final code. They were studied and developed with code applications in mind, but the applications ultimately proved unfeasible or less accurate than expected. These methods are always highlighted as such, and may be skipped to maintain flow. They are retained for future reference before others commit to studying these methods when they may not deliver the results expected, or as a foundation to improve upon with a fresh approach.

Throughout this thesis, we employ a "proposition-proof" structure for introducing various mathematical results. We appreciate that such a layout is highly unconventional in the field of astrophysics, but find it a useful way to improve flow, highlighting the main result and separating its proof from the rest of the text, while still maintaining a clear description of how the result was reached. Further, enumerating each proposition gives a simple way to refer to results throughout the text without repetition.

In addition to the proofs, some results and calculations have been verified with numerical codes. Many of the figures in this thesis were also generated by codes. All such codes are referenced in their corresponding proofs and figure captions, and can be found within the accompanying materials on GitHub.

Part I

Literature review

2 History of neutron stars and the Crab Nebula

The term **neutron star** (NS) was coined by Baade & Zwicky (1934a,b), who suggested that core-collapse supernovae yield very small and dense objects composed mostly of neutrons. They noted that neutrons can be packed together more closely than atomic nuclei, making a neutron star the most stable configuration of matter if the mass density is sufficiently high. Oppenheimer & Volkoff (1939) found an equilibrium solution to the equations of stellar structure under general relativity for a degenerate neutron gas, including an estimate for the maximum possible mass of such a structure.

The first pulsar was discovered by Dame Jocelyn Bell Burnell on 28 November 1967, **PSR B1919+21**[1] (Hewish et al., 1968). She and her collaborators observed a pulsed radio source and concluded that its origin was either a white dwarf or a neutron star. Antony Hewish went on to receive a share of 1974 Nobel Prize in Physics for the discovery, along with Sir Martin Ryle. The overlooking of Bell Burnell's contributions remains a source of controversy (e.g. Judson, 2003).

The authors originally suggested that the pulses were powered by oscillations in the host object, which was further explored by Gold (1968) and Pacini (1968). Alternatively, after a radio source was discovered at the centre of the **Crab Nebula**, Pacini (1967) suggested that nebulae could be powered by rotational and magnetic energy released from a pulsar as it spins down.

The Crab Nebula has taught us much in astrophysics. Jan Oort once stated that the study of astronomy is divided into two halves: the study of the Crab Nebula and the study of the rest of the Universe.

In 1054, a supernova was observed by Chinese astronomers and recorded as a "guest star". It was later referred to as **SN 1054**. John Bevis discovered the Crab Nebula in 1731; Charles Messier discovered it independently in 1758 and designated it **M1** in his Messier catalogue. The supernova and nebula were not yet known to be connected.

Vesto Slipher compiled photographs of the nebula at the Lowell Observatory in Flagstaff, Arizona over a number of years starting in 1912. By comparing the negatives of these photographs, it was shown that the shape of the nebula is changing, and speculated that the cause was either motion within the nebula or local brightening of matter (Lampland, 1921). Later calculations showed that the nebula was expanding (Duncan, 1921).

In 1939, Nicholas Mayall conclusively associated the Crab Nebula with the "guest star" of 1054. The idea had been suggested as early as 1921 when Knut Lundmark collated a list of guest stars that had been seen in ancient times, and strengthened by Edwin Hubble in 1928 who estimated that the nebula would have taken around 900 years to reach its present dimensions (Mayall, 1939; Mayall & Oort, 1942).

Walter Baade noted in 1942 that the expansion must have accelerated, since measured velocities of filaments implied a formation in 1172, contradicting the association with the guest star of 1054. This was counter-intuitive - one would expect the filaments to slow due to resistance from the interstellar medium. He suggested that the central star could be responsible for this acceleration, but ultimately concluded that the measurements must be spurious (Baade, 1942).

John Bolton detected strong radio wave emission in 1949, making it the first night-sky object other than the sun to be observed in the radio (Bolton & Stanley, 1949).

It was already known that the nebula features both line emission and continuum spectra, with the continuum spectrum strongly linearly polarised. Iosif Shklovsky conjectured that the continuum radiation and radio emission were from synchrotron radiation due to electrons being accelerated to relativistic speeds (e.g. Shklovsky, 1958, 1964). He also concluded that the X-ray emission, which had recently been observed, must be synchrotron radiation from electrons. However, the short synchrotron lifetime required a continuous source of new electrons in order to power the observed brightness of the nebula. Woltjer (1958, 1964) explored the possibility of a central star as the source of the synchrotron emission, and also estimated a magnetic field of $10^{14} - 10^{16}$ G using the fossil field scenario (see Proposition 4.7).

[1] Also known as PSR J1921+2153 or, less formally, LGM-1 for "little green men".

Proposition 2.1. *The synchrotron lifetime of electrons radiated from neutron stars is far shorter than the light travel time across them.*

Proof. The average energy loss rate by synchrotron emission is (e.g. Longair, 2011, Eq. (8.9))

$$\frac{dE}{dt} = -\frac{4}{3}\sigma_{\mathrm{T}}\, c\, U_{\mathrm{mag}}\, \frac{v^2}{c^2}\, \gamma^2, \tag{2.1}$$

where $\sigma_{\mathrm{T}} = \frac{e^4}{4\pi\epsilon_0^2 c^4 m_e^2} \approx 6.65 \times 10^{-29}$ m^2 is the Thomson scattering cross section, $U_{\mathrm{mag}} = \frac{1}{2\mu_0} B^2$ is the energy density of the magnetic field B, v is the velocity of the particles and γ is the Lorentz factor. The particles are electrons travelling close to the speed of light, so $\frac{v^2}{c^2} \approx 1$. Typical values of the Lorentz factor for pulsar winds are $\gamma \sim 100$ (e.g. Lyutikov et al., 2019) and we approximate $B = 10^6$ T. Then, $\frac{dE}{dt} \approx 1.1 \times 10^2$ J s^{-1}. The rest mass-energy of the electron is $E = m_e c^2 \approx 8.2 \times 10^{14}$ J. Then, dimensional analysis gives an estimate of the synchrotron lifetime as

$$\tau_{\mathrm{syn}} = \frac{E}{\frac{dE}{dt}} \approx 7.8 \times 10^{-16} \text{ s}. \tag{2.2}$$

If the star has radius R then its light travel time is $\tau_{\mathrm{light}} = \frac{R}{c}$, and $R = 10$ km gives $\tau_{\mathrm{light}} \approx 3.3 \times 10^{-5}$ s. We appreciate that $\tau_{\mathrm{syn}} \ll \tau_{\mathrm{light}}$. □

Franco Pacini conjectured in 1967 that the "central engine" driving the synchrotron radiation must be a neutron star. He estimated that a rotation period 10^{-3} s and magnetic field of 10^{10} G could yield radiation with intensity 10^{40} erg s^{-1}, enough to cause the observed acceleration of the nebula (Pacini, 1967).

Radio emission from near the centre of the nebula was detected by Comella et al. (1969), with a period of 33.09112 ± 0.00003 ms, roughly in line with that expected by Pacini. The authors used this value to argue that the central engine was indeed a rotating neutron star, since a pulsating white dwarf would require excitation of very high-order modes to reproduce it.

Bell Burnell's and Hewish's 1967 discovery of the first pulsar finally confirmed the existence of neutron stars, and led to the suggestion that a neutron star lies in the centre of the Crab Nebula. An object was observed later that year and named the **Crab Pulsar**; its period of 33 ms ruled out a white dwarf (with a radius $R \sim 10^3$ km, a white dwarf would be ripped apart by centripetal forces at this rotation speed), leaving a neutron star as the only viable candidate.

Later observations confirmed a magnetic field of 5×10^{12} G and energy loss rate of 10^{38} erg s^{-1}, matching the luminosity of the Crab nebula and confirming that the Crab Pulsar is able to power the luminosity of the nebula. Modern observations estimate the polar magnetic field of the Crab pulsar at around $B_{\mathrm{pole}} = 7.6 \times 10^{12}$ G (e.g. Kou & Tong, 2015).

3 Formation

Mass density strongly increases toward the centre of a star, so the thermal temperature, which depends strongly on density, also increases. This enables nuclear fusion reactions involving heavier and heavier nuclei as we move toward the core, and so the stellar interior becomes stratified with increasingly massive fusion products. This is the famous **onion-skin model** of a stellar interior.

The greater the initial mass, the greater the central density and core temperature, and so fusion can proceed to more massive nuclei. This occurs up until silicon-56, which represents the peak of the curve of binding energy per nucleon (e.g. Martin, 2009, Figure 2.8); fusion of heavier species is possible but energy is absorbed instead of released, inhibiting further reactions. If any silicon-56 is fused via alpha capture, the loss of energy will lower the surrounding temperature and make further reactions less likely until an equilibrium is reached. Stars with initial mass $\gtrsim 8\,M_\odot$ typically evolve up to the silicon-burning stage.

Energy in an iron core is absorbed by two main processes:

- **Photodisintegration:** Thermal photons carrying sufficient energy are absorbed by nuclei and cause them to break apart into smaller nuclei. The two main processes are photodisintegration of iron-56 and helium-4:

$$\gamma + {}^{56}\mathrm{Fe} \rightleftharpoons 13\,{}^{4}\mathrm{He} + 4\,n, \tag{3.1}$$

$$\gamma + {}^{4}\mathrm{He} \rightleftharpoons 2\,p + 2\,n. \tag{3.2}$$

- **Electron capture**, also known as **neutronisation**. The degenerate electrons may combine with protons to form neutrons and electron neutrinos. The general reaction is

$$p + e^- \rightarrow n + \nu_e. \tag{3.3}$$

  The first reaction that occurs is

$${}^{56}\mathrm{Fe} + e^- \rightarrow {}^{56}\mathrm{Mn} + \nu_e, \tag{3.4}$$

  and the ^{56}Mn may undergo further electron capture to ^{56}Cr and so on. Not only does extreme energy loss occur by the radiation of neutrinos, which pass through the star almost unimpeded, but also the loss of degenerate electrons reduces pressure support and the core continues to collapse.

For an iron core at the Chandrasekhar mass, the processes of photodisintegration and electron capture can each cause around 10^{45} J of energy to be lost within a few seconds (Ryan & Norton, 2010, §7.1).

Proposition 3.1. *It would take roughly the entire main-sequence lifetime of a* $10\,M_\odot$ *star to radiate the same amount of energy released by photodisintegration and electron capture in an iron core.*

Proof. The **mass-luminosity relation** is a semi-empirical formula estimating the relationship between luminosity and mass of main-sequence stars, $L \propto M^\nu$. Comparing to the Sun, we can eliminate the constant of proportionality:

$$\frac{L}{L_\odot} = \left(\frac{M}{M_\odot}\right)^\nu. \tag{3.5}$$

The exponent also depends on the mass, so it is not strictly valid to divide $L \propto M^\nu$ and $L_\odot \propto M_\odot^\nu$ in this way with the same ν, but we only require a rough estimate. In particular, $\nu \approx 3.6$ for $M \approx 2 - 20\,M_\odot$ and $\nu \approx 4.5$ for $M \approx 0.5 - 2\,M_\odot$, so the difference is not too great (Salaris & Cassisi, 2005, §5.7). Assuming that the star has constant luminosity while it is on the main sequence, the time t taken to radiate energy E is $t = \frac{E}{L}$, giving

$$t = \frac{E}{L_\odot}\left(\frac{M}{M_\odot}\right)^{-\nu}. \tag{3.6}$$

For $M = 10\,M_\odot$, $\nu = 3.6$ and $E = 10^{45}$ J, we obtain $t \approx 20.8$ Myr, which is comparable to estimates of the core hydrogen-burning lifetime for such a star from simulations (Salaris & Cassisi, 2005, Table 5.1). □

Proposition 3.2. *Neutronisation can occur if the core temperature exceeds* $T_\mathrm{c} \gtrsim 1.5 \times 10^{10}$ K*, while photodisintegration of iron-56 can occur if* $T_\mathrm{c} \gtrsim 5.7 \times 10^{12}$ K*.*

Proof. The degenerate electrons may combine with protons if their energy exceeds the mass-energy difference between a proton and a neutron:

$$E_e > (m_p - m_n)c^2 \sim 1.3 \text{ MeV}, \tag{3.7}$$

which corresponds to a thermal temperature $T > \frac{E_e}{k_B} \approx 1.5 \times 10^{10}$ K. For photodisintegration, thermal photons must carry energy $56\,E_B$, where $E_B \approx 8.8$ MeV is the binding energy per nucleon of iron-56. This corresponds to a core temperature $T_\mathrm{c} > \frac{56\,E_B}{k_B} \approx 5.7 \times 10^{12}$ K. □

Once the iron core exceeds the Chandrasekhar mass, electron degeneracy pressure becomes insufficient and it collapses. The gravitational free-fall timescale for a spherical mass with constant density ρ and no pressure support is (e.g. Ryan & Norton, 2010, Eq. (2.5))

$$T_\mathrm{ff} = \sqrt{\frac{3\pi}{32G\rho}}, \tag{3.8}$$

so an iron core with density $\rho \sim 10^{14}$ kg m^{-3} will collapse on a timescale of 7 ms. Collapse continues until nuclear density is reached, at which point the strong force prevents further infall. The core rebounds and produces a shockwave, which propagates to the weakly gravitationally bound envelope and ejects it. This releases an incredibly vast amount of energy and is observed as a **Type II supernova**.

Proposition 3.3. *The number density and mass density of nuclear matter are* $n \approx 1.3 \times 10^{44}\ \mathrm{m}^{-3} = 0.13\ \mathrm{fm}^{-3}$ *and* $\rho \sim 10^{17}\ \mathrm{kg\ m}^{-3}$*, approximately independent of the species.*

Proof. An atomic nucleus with mass number A (the total number of neutrons and protons) has mass $m = Au$, where $u \approx 1.66 \times 10^{-27}$ kg is the **atomic mass unit**, or the **dalton**. Nuclei can be reasonably well represented as homogeneous spheres with radius $r = r_0 A^{1/3}$, where $r_0 = 1.21$ fm is a constant obtained by fitting to empirical data (e.g. Martin, 2009, §2.2.1). Then, the nucleus has volume

$$V = \frac{4}{3}\pi r_0^3 A. \tag{3.9}$$

The density is then

$$\rho = \frac{M}{V} = \frac{3u}{4\pi r_0^3} = \text{constant} \approx 2.24 \times 10^{17}\ \mathrm{kg\ m}^{-3}, \tag{3.10}$$

and the number density is

$$n = u\rho \approx 1.35 \times 10^{44}\ \mathrm{m}^{-3}. \tag{3.11}$$

□

Proposition 3.4. *A core of iron-56 collapsing from the Chandrasekhar mass down to nuclear density releases around* 10^{46} J *of energy.*

Proof. The iron core immediately before collapse has radius around $R_1 \approx 3000$ km (Sukhbold et al., 2016). If it collapses down to nuclear density ρ_{nucl}, its final radius assuming a uniform sphere of mass $M = M_{\mathrm{Ch}} = 1.4\,M_\odot$ will be

$$R_2 = \left(\frac{3}{4\pi}\frac{M_{\mathrm{Ch}}}{\rho_{\mathrm{nucl}}}\right)^{1/3} \approx 14.3\ \mathrm{km}. \tag{3.12}$$

A crude estimate of the total gravitational potential energy contained in a uniform sphere of mass M and radius R can be given by splitting it into two hemispheres and considering the gravitational potential energy between them. Each has mass $\frac{1}{2}M$. The centre-of-mass of a uniform hemisphere of radius R is located a distance $\frac{3}{8}R$ from the flat surface, so the separation between the centres-of-mass is twice this, $\frac{3}{4}R$. We have

$$U\left(\tfrac{1}{2}M, \tfrac{3}{4}R\right) = -\frac{G\left(\frac{1}{2}M\right)\left(\frac{1}{2}M\right)}{\left(\frac{3}{4}R\right)} = -\frac{GM^2}{3R}. \tag{3.13}$$

Collapse from R_1 to R_2 with constant mass then gives a change in gravitational potential energy

$$\Delta U = U_2 - U_1 = U\left(\tfrac{1}{2}M, \tfrac{3}{8}R_2\right) - U\left(\tfrac{1}{2}M, \tfrac{3}{8}R_1\right) = \frac{GM^2}{3}\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \approx -1.1 \times 10^{46}\ \mathrm{J}. \tag{3.14}$$

Since $\Delta U < 0$, this energy is released and not absorbed, as we may expect. □

The energy released by core collapse far exceeds that released by photodisintegration and that observed as kinetic energy of the expanding debris. It is thought that the excess energy is radiated as neutrinos. Neutrinos may provide an early warning sign of supernovae, allowing astronomers to prepare for observations. Several neutrinos were detected a few hours before SN1987A (Hirata et al., 1987, 1988; Bionta et al., 1987; Bratton et al., 1988; Alexeyev et al., 1988; Kunkel et al., 1987). The SNEWS collaboration consists of seven neutrino detectors across that globe (Antonioli et al., 2004) that hope to allow triangulation of an imminent supernova's position to a few percent of the sky (Linzer & Scholberg, 2019).

4 Properties

Proposition 4.1. *The time derivative of the angular frequency of a rotating magnetic dipole is*

$$\dot{\Omega} \;=\; -\frac{2\pi}{3\mu_0 c^3}\,\frac{B_{\text{pole}}^2 R^6 \Omega^3 \sin^2(\alpha)}{I}, \tag{4.1}$$

where B_{pole} is the magnetic field strength of the dipole at its pole, R is its radius (distance from the centre to each charge), α is the angle between the magnetic dipole moment and the rotation axis and I is the moment of inertia of the dipole.

Proof. We use the method shown in §5.4.1 of Rosswog & Brüggen (2007). A magnetic dipole has magnetic field (e.g. Griffiths, 2017, Eq. (5.88))

$$\mathbf{B}(\mathbf{r}) \;=\; \frac{\mu_0\, m}{4\pi r^3}\Big[2\cos(\theta)\,\mathbf{e}_r + \sin(\theta)\,\mathbf{e}_\theta\Big], \tag{4.2}$$

where $m = |\mathbf{m}|$ is the magnitude of its magnetic dipole moment $\mathbf{m}$. Then, at the North pole we have

$$\mathbf{B}_{\text{pole}} \;=\; \mathbf{B}(R,0,0) \;=\; \frac{\mu_0\, m}{2\pi R^3}\,\mathbf{e}_r \;=\; \frac{\mu_0}{2\pi R^3}\,\mathbf{m}, \tag{4.3}$$

where, according to Rosswog & Brüggen (2007), $\mathbf{m} \parallel \mathbf{r}$ and $r = R\,\mathbf{e}_r$ so that $\mathbf{m} = m\,\mathbf{e}_r$. Now, the power radiated by an *electric* dipole is given by the Larmor formula (e.g. Griffiths, 2017, Eq. (11.61))

$$\left(\frac{dW}{dt}\right)_{\text{electr}} \;=\; -\frac{\mu_0}{6\pi c}\,(\ddot{p})^2, \tag{4.4}$$

where $p = |\mathbf{p}|$ is the magnitude of the electric dipole moment, so by analogy the power radiated by a *magnetic* dipole is

$$\left(\frac{dW}{dt}\right)_{\text{mag}} \;=\; -\frac{\mu_0}{6\pi c^3}\,(\ddot{m})^2, \tag{4.5}$$

where we introduced a factor $\frac{1}{c^2}$ to balance the units. Suppose that the dipole rotates about the z-axis with angular frequency Ω, and that the magnetic dipole moment makes an angle α to the z-axis (Rosswog & Brüggen, 2007, Figure 5.6). Then, we can express $\mathbf{m}$ in spherical coordinates and use it to calculate $(\ddot{m})^2$:

$$\mathbf{m} \;=\; \frac{2\pi}{\mu_0}\,R^3 B_{\text{pole}}\Big[\sin(\alpha)\cos(\Omega t)\mathbf{i} + \sin(\alpha)\sin(\Omega t)\mathbf{j} + \cos(\alpha)\mathbf{k}\Big], \tag{4.6}$$

$$\Rightarrow\; \ddot{\mathbf{m}} \;=\; -\frac{2\pi}{\mu_0}\,R^3 B_{\text{pole}}\,\omega^2 \sin(\alpha)\Big[\cos(\Omega t)\mathbf{i} + \sin(\Omega t)\mathbf{j}\Big], \tag{4.7}$$

$$\Rightarrow\; (\ddot{\mathbf{m}})^2 \;=\; \ddot{\mathbf{m}}\cdot\ddot{\mathbf{m}} \;=\; \left(-\frac{2\pi}{\mu_0}\,R^3 B_{\text{pole}}\,\Omega^2 \sin(\alpha)\right)^2\Big[\cos^2(\Omega t) + \sin^2(\Omega t)\Big] \;=\; \frac{4\pi^2}{\mu_0^2}\,B_{\text{pole}}^2\Omega^4 R^6 \sin^2(\alpha). \tag{4.8}$$

The rotational energy of an object with moment of inertia I is $E_{\text{rot}} = \frac{1}{2}\,I\Omega^2$. Assuming a rigid sphere such that I is constant in time, the rate of change of rotational energy is then $\dot{E}_{\text{rot}} = I\Omega\dot{\Omega}$. Assume that all the radiated power comes from rotational energy, $\left(\frac{dW}{dt}\right)_{\text{mag}} = \dot{E}_{\text{rot}}$. Rearranging for $\dot{\Omega}$, we find

$$\dot{\Omega} \;=\; \frac{\dot{E}_{\text{rot}}}{I\Omega} \;=\; \frac{1}{I\Omega}\left(\frac{dW}{dt}\right)_{\text{mag}} \;=\; -\frac{4\pi}{6\mu_0 c^3}\,\frac{B_{\text{pole}}^2 R^6 \Omega^3 \sin^2(\alpha)}{I}. \tag{4.9}$$

□

Proposition 4.2. *The magnetic field strength at the poles of a neutron star can be estimated given measured values of its rotation period P and period derivative $\dot{P}$ and an assumed radius R and moment of inertia I:*

$$B_{\text{pole}} \;=\; \frac{1}{R^3 \sin(\alpha)} \sqrt{\frac{3\mu_0 c^3}{8\pi^3}\, I P \dot{P}}. \tag{4.10}$$

Proof. Rotation period and angular velocity are related by $P = \frac{2\pi}{\Omega}$, and so the period derivative can be written by the chain rule as $\dot{P} = -2\pi\, \frac{\dot{\Omega}}{\Omega^2}$. Substitute the expression for $\dot{\Omega}$ in Eq. (4.1) and rearrange. The result agrees with Eq. (33) in Chapter 16 of Carroll & Ostlie (2014). □

A rotating electric dipole emits radiation and loses energy, causing it to slow down. Note that $\dot{\omega} < 0$, so the dipole indeed slows down over time. For a uniform solid sphere, $I = \frac{2}{5} M R^2$. Typical values are $M = 1.4\, M_\odot$, $R = 10$ km and $P = 10$ s. Using these, a measured spin-down rate $\dot{P} \approx 10^{-12}$ s s^{-1} implies a magnetic field $B \approx 10^{10}$ T $= 10^{14}$ G.

Proposition 4.3. *Assume that pulsar spin-down follows a power law,*

$$\dot{\Omega} \;=\; -K\, \Omega^n, \tag{4.11}$$

where K is some proportionality constant and n is called the **braking index**. *Then, the* **characteristic age** *of the pulsar is*

$$\tau \;=\; -\frac{1}{n-1}\, \frac{\Omega}{\dot{\Omega}} \;=\; \frac{1}{n-1}\, \frac{P}{\dot{P}}. \tag{4.12}$$

Proof. We use the method shown in §5.4.2 of Rosswog & Brüggen (2007). Separate the variables and integrate from the time of pulsar formation t_0, when the angular velocity was $\Omega_0 = \Omega(t_0)$, to an arbitrary time t, when the angular velocity is Ω:

$$\int_{\Omega_0}^{\Omega} (\Omega')^{-n}\, d\Omega' \;=\; -K \int_{t_0}^{t} dt', \tag{4.13}$$

$$\Rightarrow \quad \left[\frac{1}{-n+1}\, (\Omega')^{-n+1} \right]_{\Omega_0}^{\Omega} \;=\; -K \left[t' \right]_{t_0}^{t}, \tag{4.14}$$

$$\Rightarrow \quad -\frac{1}{n-1} \left[\frac{1}{\Omega^{n-1}} - \frac{1}{\Omega_0^{n-1}} \right] \;=\; -K \left[t - t_0 \right], \tag{4.15}$$

where $-n+1 = 1-n = -(n-1)$ and the primes denote dummy variables for integration. Rearrange for the age of the pulsar $\tau \equiv t - t_0$, using that $-\frac{1}{K} = \frac{\Omega^n}{\dot{\Omega}}$:

$$t - t_0 \;=\; -\frac{1}{n-1}\, \frac{\Omega^n}{\dot{\Omega}} \left[\frac{1}{\Omega^{n-1}} - \frac{1}{\Omega_0^{n-1}} \right] \;=\; -\frac{1}{n-1}\, \frac{\Omega}{\dot{\Omega}} \left[1 - \left(\frac{\Omega}{\Omega_0} \right)^{n-1} \right]. \tag{4.16}$$

Most observed pulsars are relatively old so that they have slowed significantly, $\Omega \ll \Omega_0$. Further, observed braking indices are around $2-3$, so $\frac{\Omega}{\Omega_0} \ll 1$ and $\left(\frac{\Omega}{\Omega_0}\right)^{n-1} \ll 1$ also, so this term can be ignored and we obtain the given expression. To convert from angular frequency to period, note that $\Omega = \frac{2\pi}{P}$ and so the chain rule gives $\dot{\Omega} = -\frac{2\pi}{P^2}\, \dot{P}$, so that $\frac{\Omega}{\dot{\Omega}} = -\frac{P}{\dot{P}}$. □

We see from Eq. (4.1) that the dipole approximation gives $n = 3$ and hence a **dipole age**

$$\tau_{\text{dipole}} \;=\; -\frac{1}{2}\, \frac{\Omega}{\dot{\Omega}} \;=\; \frac{1}{2}\, \frac{P}{\dot{P}}. \tag{4.17}$$

Proposition 4.4. *Suppose that the pulsar formed at time t_0 with initial period P_0, and suppose that the product $P\dot{P}$ remains constant over time. Then, its period and period derivative as a function of its age $\tau \equiv t - t_0$ at time t are*

$$P(\tau) = P_0\sqrt{1+f\tau}, \tag{4.18}$$

$$\dot{P}(\tau) = \frac{1}{2}P_0\frac{f}{\sqrt{1+f\tau}}, \tag{4.19}$$

where $f \equiv \frac{16\pi^3}{3\mu_0 c^3}\frac{B^2R^6\sin^2(\alpha)}{I}\frac{1}{P_0^2}$ is used for notational shorthand. To a good approximation these reproduce the dipole age $\tau = \frac{P}{2\dot{P}}$. The pulsar follows a line of constant B on the $P-\dot{P}$ diagram as it evolves.

Proof. We use the method outlined in Chapter 16, Problem 18 of Carroll & Ostlie (2014). Rearrange Eq. (4.10) for the constant $P\dot{P}$, use that $\dot{P} = \frac{dP}{dt}$, separate the variables and integrate from time t_0, when the period was P_0, to t, when the period is P:

$$\int_{P_0}^{P} P'\,dP' = \int_{t_0}^{t}\frac{8\pi^3}{3\mu_0c^3}\frac{B^2R^6\sin^2(\alpha)}{I}\,dt', \tag{4.20}$$

$$\Rightarrow \left[\frac{1}{2}(P')^2\right]_{P_0}^{P} = \left[\frac{8\pi^3}{3\mu_0c^3}\frac{B^2R^6\sin^2(\alpha)}{I}\,t'\right]_{t_0}^{t}, \tag{4.21}$$

$$\Rightarrow \frac{1}{2}\left(P^2-P_0^2\right) = \frac{8\pi^3}{3\mu_0c^3}\frac{B^2R^6\sin^2(\alpha)}{I}\left(t-t_0\right), \tag{4.22}$$

which rearranges to the given expression for P. The expression for $\dot{P}$ follows by differentiation with the chain rule. Let us now attempt to reproduce the expression for the dipole age:

$$\frac{P}{2\dot{P}} = \frac{P_0\sqrt{1+f\tau}}{2\cdot\frac{1}{2}P_0\frac{f}{\sqrt{1+f\tau}}} = \frac{1}{f}\left(1+f\tau\right) = \frac{1}{f}+\tau. \tag{4.23}$$

Let us use values for the Crab Pulsar: $B = 7.6\times10^8$ T, $M = 1.4\,M_\odot$, $R\sim10^4$ m and $P_0 = 0.033$ s yield $f\approx 2\times10^{-11}$ s^{-1} and $\frac{1}{f}\approx 4\times10^{10}$ s. Typical pulsars are in the range of kyr $\sim10^{10}$ s to Myr $\sim10^{13}$ s, so for most ages we have $\frac{1}{f}\ll\tau$ and the added term $\frac{1}{f}$ can be neglected. The dependence on B^2R^6 makes the age at which $\frac{1}{f}<\tau$ quite sensitive to the input parameters; we must remember that the expression $\tau = \frac{P}{2\dot{P}}$ is only a rough estimate.

We should not be surprised that B remains constant in this model, since it was among our assumptions when we performed the integration over time. However, it is useful to show this explicitly now. Taking logarithms of both expressions and substituting one into the other yields

$$\log_{10}(P) = \log_{10}(P_0) + \frac{1}{2}\log_{10}(1+f\tau), \tag{4.24}$$

$$\log_{10}(\dot{P}) = -\log_{10}(2) + \log_{10}(P_0) + \log_{10}(f) - \frac{1}{2}\log_{10}(1+f\tau) \tag{4.25}$$

$$= -\log_{10}(2) + \log_{10}(f) - \log_{10}(P), \tag{4.26}$$

so we see that $\frac{d\log_{10}(\dot{P})}{d\log_{10}(P)} = -1$ and the pulsar indeed evolves along a line of constant magnetic field. □

Proposition 4.5. *A pulsar with a relatively long period is unlikely to have spun down to this value within a reasonably short time unless its magnetic field were very strong.*

Proof. Eq. (4.18) rearranges to give

$$\tau = \frac{1}{f}\left[\left(\frac{P}{P_0}\right)^2 - 1\right]. \tag{4.27}$$

The same fiducial properties as in Proposition 4.4 yield a time of $\tau \sim 1.2$ Myr for the Crab Pulsar to spin-down to $P = 1$ s and $\tau \sim 120$ Myr to spin-down to $P = 10$ s. At $B = 1 \times 10^8$ T, we have 72 Myr and 7.2 Gyr respectively, the latter being of order the age of the Galaxy and hence extremely unlikely. However, if we allow an initial magnetic field of $B = 10^{10}$ T, these become 7.2 kyr and 720 kyr respectively. □

We plot the period and period derivative as a function of time on a $P - \dot{P}$ diagram in Figure 4.1, using data for the Crab Pulsar: initial period $P_0 = 33$ ms (i.e. using the present-day value and evolving into the future) and mass $1.4\,M_\odot$. To emphasise the direction of motion across the diagram, the red star represents the initial values and the blue star represents the final values, taken at $\tau = 1$ Myr into the future.

As we predicted above, the pulsar evolves along a line of constant magnetic field, beginning near the top left of the figure and ending near the bottom right as its period increases and its rate of spin-down decreases. As a result of the latter, the jumps between successive gridpoints, which are evenly spaced in time, become progressively smaller. That is, we expect younger pulsars to experience significant spin-down but older pulsars to be relatively stable. This can also be appreciated from Figure 4.3, where the lines of constant age, while evenly spaced on that figure, represent tenfold jumps in pulsar age.

The track is similar to the blue trace in Figure 3 of Kou & Tong (2015). Our simple model does not take into account phenomena such as time-dependence of the magnetic field, pulsar glitches and a possible time-dependence of the braking index, but is enough to give an overview.

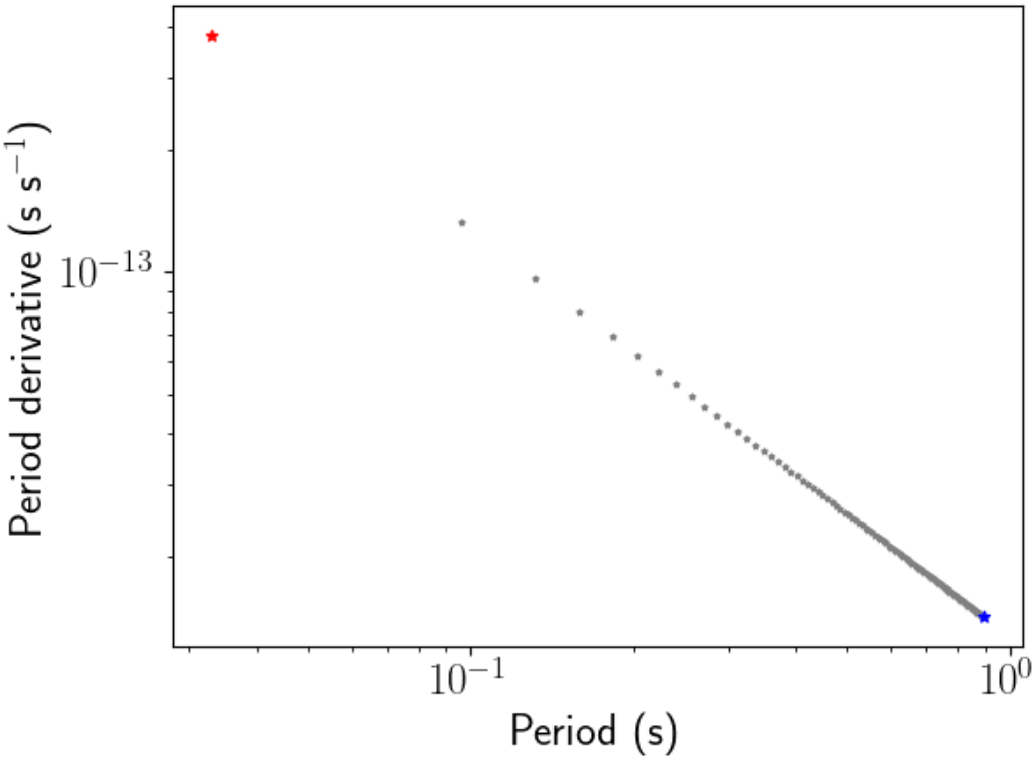

Figure 4.1: Evolutionary track of the Crab Pulsar across the $P - \dot{P}$ diagram from the present day (red star, top left) to 1 Myr in the future (blue star, bottom right).
Produced by code: `Plot_Pulsar_Period_as_Function_of_Time.py`

Proposition 4.6. *A solid homogeneous sphere with initial radius R_1 and rotational period P_1 that collapses to new radius R_2 will have new rotational period*

$$P_2 = P_1 \left(\frac{R_2}{R_1} \right)^2. \tag{4.28}$$

Proof. The magnitude of the angular momentum of a rotating object is $L = I\omega$, where I is its moment of inertia and ω is its angular velocity. The angular velocity and rotational period are linked by $\omega = \frac{2\pi}{P}$. For a solid homogeneous sphere of mass M and radius R, $I = \frac{2}{5} MR^2$. Then,

$$L = \frac{4\pi}{5} \frac{MR^2}{P}. \tag{4.29}$$

Angular momentum is a conserved quantity, so the sphere will have the same value of L before its collapse (with radius R_1 and period P_1) and after its collapse (with R_2 and P_2). Equating these and cancelling the constants, including the mass, we find that

$$\frac{R_1^2}{P_1} = \frac{R_2^2}{P_2}, \tag{4.30}$$

which rearranges to the given result. □

The Sun exhibits differential rotation, with equatorial rotational period 25.6 days and polar rotational period 33.5 days. Let us crudely approximate it as a uniformly rotating sphere whose period is the average of these,

$$P_1 = \frac{(25.6 \text{ d}) + (33.5 \text{ d})}{2} \approx 29.6 \text{ d} \approx 2.55 \times 10^6 \text{ s}. \tag{4.31}$$

Its equatorial radius is $R_1 \approx 6.96 \times 10^5$ km. If it were to suddenly collapse to a neutron star with $R_2 = 10$ km, its rotation period would be $P_2 \approx 5.27 \times 10^{-4}$ s, roughly comparable with some of the fastest observed pulsars.

Proposition 4.7. *A sphere with a dipolar magnetic field of initial strength B_1 compressing/expanding from radius R_2 to R_1 will experience a magnetic field amplification/suppression such that its new field strength is*

$$B_2 = B_1 \left(\frac{R_2}{R_1} \right)^2. \tag{4.32}$$

Proof. Magnetic flux Φ is measured in webers, where 1 Wb = 1 T m^2. Thus, we have $\Phi \propto BA$, where B is the magnetic field and A is the area through which the flux passes. Since magnetic flux is conserved, we have $B_1 A_1 = B_2 A_2$. Now, we must be careful which area we choose: Gauss's law for magnetism states that the flux through any *closed* surface is zero, so we cannot use the entire surface of the sphere. Instead, let us choose a surface defining only the northern hemisphere, such that all the field lines point "out", not "in". Then, $A = \frac{1}{2} \frac{4}{3} \pi R^2 = \frac{2}{3} \pi R^2$ and we obtain $B_1 R_1^2 = B_2 R_2^2$, which rearranges to the given result. □

One explanation for the high magnetic field strengths of pulsars, attractive for its simplicity, is the **fossil field scenario**, which states that the magnetic field is simply that of the progenitor star, amplified by flux conservation during the supernova event. Main-sequence stars have been observed with magnetic fields as high as $B_1 \approx 10^4$ G (Donati & Landstreet, 2009), and an iron core may have initial radius $R_1 \approx 3000$ km (Sukhbold et al., 2016). These results suggest that under collapse to a neutron star with $R_2 \approx 10$ km, flux conservation alone could generate magnetic fields up to 10^8 G. However, this does not agree well with measured NS magnetic fields; for example, the Crab Pulsar has $B \approx 7.6 \times 10^{12}$ G (e.g. Kou & Tong, 2015).

Additionally, the Galactic distribution of magnetic fields of massive stars is insufficient to reproduce the observed pulsar and magnetar distribution (Makarenko et al., 2021). Further, the progenitor star exhibits core

convection and the proto-neutron star is expected to be unstable to particularly vigorous convection driven by a neutrino flux of $\sim 10^{53}$ erg s^{-1} in its first few seconds of existence (Thompson & Duncan, 1993), so that young neutron stars have extremely high values of the magnetic Reynolds number $R_\mathrm{m} \sim 10^{17}$; compare this to $R_\mathrm{m} \sim 10^{10}$ for the Sun.

Rapidly-moving conducting fluids within an object with high R_m can generate intense magnetic fields by **dynamo action**; specifically, the **$\boldsymbol{\alpha - \Omega}$ dynamo mechanism** (e.g. Davidson, 2001, §6.2.1). If the bulk movement of matter is powerful enough, it can drag magnetic field lines along with it, possibly leading to magnetic field amplification. If the rotation period is less than the convective turnover time $P < \tau_\mathrm{con}$, this amplification becomes global: strong fields of 10^{15} G or higher permeate the entire star and its magnetosphere. Proto-NSs which achieve this are known as **magnetars**. For longer periods, the amplification is only local (confined within a small region of the NS) and the NS becomes a "normal radio pulsar". Thompson & Duncan (1993) estimate $\tau_\mathrm{con} \sim 10^{-3}$ s (Table 1 in their paper) and suggest that a proto-NS born with $P < 10$ ms could become a magnetar through dynamo action. Lander (2021) further explores the use of dynamo phases in NS magnetic field generation.

Proposition 4.8. *Neutrons far outnumber protons within the core of a neutron star, hence the name.*

Proof. We use a method outlined in §7.2.1 of Ryan & Norton (2010). When the mass density exceeds that of an atomic nucleus $\rho \sim 10^{17}$ kg m^{-3}, **neutron drip** occurs, in which neutrons leak out of nuclei. Some of these free neutrons then decay via

$$n \leftrightharpoons p + e^- + \overline{\nu}_e, \tag{4.33}$$

to give a degenerate gas of neutrons, protons and electrons. Under "normal conditions", free neutrons decay via the above reaction with a half-life of around 14 minutes. However, neutron star cores are so dense that few quantum states are available for the protons and electrons to occupy, inhibiting this reaction. Specifically, neutron decay is favoured when the Fermi energy of the neutrons exceeds the sum of the Fermi energies of the protons and electrons (such that the system seeks its lowest-energy configuration) and inhibited in the opposite case. We thus have equilibrium when the Fermi energies balance:

$$E_\mathrm{F}(n) = E_\mathrm{F}(p) + E_\mathrm{F}(e). \tag{4.34}$$

The neutrons and protons are non-relativistic, while the electrons are ultra-relativistic. Their Fermi energies are

$$E_\mathrm{F}(n,p) = m_{n,p}\,c^2 + \frac{p_\mathrm{F}^2(n,e)}{2m_{n,e}}, \tag{4.35}$$

$$E_\mathrm{F}(e) = p_\mathrm{F}(e)\,c, \tag{4.36}$$

where the Fermi momentum is

$$p_\mathrm{F} = \left(\frac{3}{8\pi}\right)^{1/3} h\,n^{1/3}. \tag{4.37}$$

Consider a system with mass density ρ consisting of several species of particles. For species i whose particles each have mass m_i, the the number density n_i (particles per unit volume) and mass fraction $X_i \in [0,1]$ (dimensionless number giving the contribution of the species to the total mass of the system) are related by

$$n = \frac{\rho\,X}{m}. \tag{4.38}$$

We have a system of free neutrons, protons and electrons. Charge neutrality gives that $n_p = n_e$, and the mass fractions must sum to unity, $X_n + X_p + X_e = 1$. Then, n_p and hence n_e can be expressed terms of n_n:

$$n_p = \frac{\rho - m_n\,n_n}{m_p + m_e}. \tag{4.39}$$

Combining these results, we find an expression in terms of a single variable[1] n_n for a given mass density ρ, which is equal to zero at equilibrium:

$$f(n_n) = \frac{1}{2}\left(\frac{3}{8\pi}\right)^{2/3} h^2 \left[\frac{1}{m_n} n_n^{2/3} - \frac{1}{m_p} n_p^{2/3}\right] - \left(\frac{3}{8\pi}\right)^{1/3} hc\, n_p^{1/3} + (m_n - m_p)c^2 \stackrel{!}{=} 0. \tag{4.40}$$

The expression can be solved numerically; we write the `Python` code
`Decimal_search_NS_neutron_number_density.py`
which determines n_n to a chosen accuracy by a decimal-search method: slowly increasing n_n from an initial guess until the sign of $f(n_n)$ changes and then repeating for a smaller interval around that value. This requires an initial guess for n_n: We choose an initial value such that $n_n = n_p$, which yields

$$n_{n,\text{initial}} = \frac{\rho}{m_n + m_p + m_e}, \tag{4.41}$$

and fine-tune this to aid code convergence. The resulting n_n is plotted in Figure 4.2 for ρ ranging from 1.4 to 3.0 kg m^{-3}, alongside the resulting ratio n_n/n_p of the number of neutrons to protons. We find $n_n/n_p \sim 200-300$ for this range of densities, confirming that neutrons are indeed the dominant species within NS cores. □

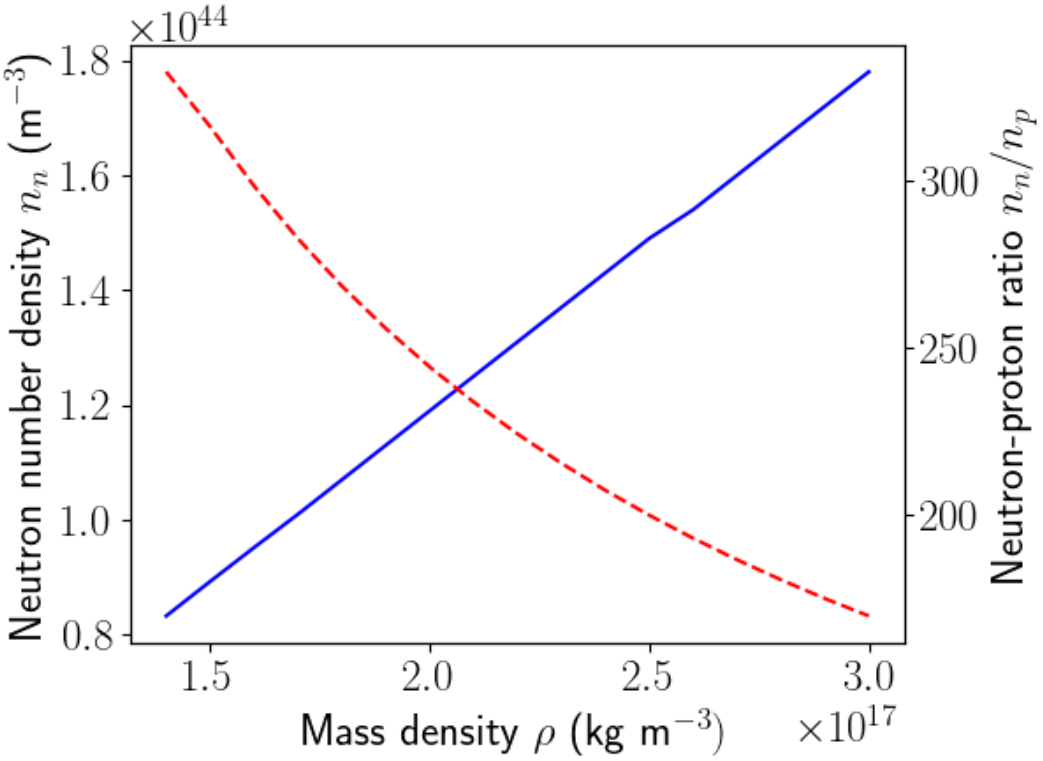

Figure 4.2: Neutron number density (solid blue line) and neutron-to-proton ratio (dashed red line) of a degenerate gas of non-relativistic neutrons and protons and ultra-relativistic electrons, as a function of mass density. Produced by codes: `Decimal_Search_NS_Neutron_Number_Density.py`
and `Plot_NS_Neutron_to_Proton_Ratio.py`

[1] We keep n_p as a separate term since the explicit expression does nothing to simplify the function, and it is useful to know n_p anyway.

An approximate **mass-radius relation** for neutron stars is (Ryan & Norton, 2010, §7.2.2)

$$R = \left(\frac{729}{32\pi^4}\right)^{1/3} \frac{1}{G} \frac{h^2}{5} \frac{1}{m_n^{8/3}} \frac{1}{M^{-1/3}} \approx 16.30 \text{ km} \left(\frac{M}{M_\odot}\right)^{-1/3}. \tag{4.42}$$

For a neutron star with the Chandrasekhar mass, $R \approx 14.57$ km. Comparing to Eq. (3.12), this puts the star around nuclear mass density on average. The core region is expected to be far above nuclear density.

Neutron star masses should all exist within a relatively narrow range, roughly $1.4 - 2.9\,M_\odot$. The iron core must reach the Chandrasekhar mass in order to collapse, so if no further mass loss occurs during formation then this represents the lowest possible NS mass. An upper limit can be derived using the same logic as for the maximum mass of a white dwarf.[2] The first to do this were Tolman (1939) and Oppenheimer & Volkoff (1939), so the maximum has been named the **Tolman-Oppenheimer-Volkoff (TOV) limit**. The exact value of the TOV limit is unknown because calculations are highly sensitive to the still-unknown equation of state of a neutron star interior. The original authors' estimate was $M_{\text{TOV}} \approx 0.7\,M_\odot$; recently, Jiang et al. (2020) used NICER X-ray data to find $M_{\text{TOV}} \approx 2.04-2.40\,M_\odot$ depending on the equation of state. It is thought that rotation may increase the maximum mass by 18−20% (Rezzolla et al., 2018), raising this estimate to $2.40-2.88\,M_\odot$. The most massive observed pulsar is PSR J0952–0607 with $M = 2.35 \pm 0.17\,M_\odot$ (Romani et al., 2022). Calculations by Clifford & Ransom (2019) estimate that PSR J1748-2021B has mass $M = 2.548^{+0.047}_{-0.078}\,M_\odot$.

Proposition 4.9. *In a $P - \dot{P}$ diagram, straight lines of gradient $+1$ (bottom left to top right) represent constant characteristic age.*

Proof. Note that $\omega = \frac{2\pi}{P}$ and so by the chain rule $\dot{\omega} = -2\pi \frac{\dot{P}}{P^2}$. Then, Eq. (4.12) for the characteristic age of a pulsar can be written

$$\tau = \frac{1}{n-1} \frac{P}{\dot{P}}. \tag{4.43}$$

Take the logarithm of both sides and rearrange for $\log_{10}(\dot{P})$:

$$\log_{10}(\tau) = -\log_{10}(n-1) + \log_{10}(P) - \log_{10}(\dot{P}), \tag{4.44}$$

$$\Rightarrow \log_{10}(\dot{P}) = \log_{10}(P) - \log_{10}(\tau) - \log_{10}(n-1). \tag{4.45}$$

Differentiate with respect to $\log_{10}(P)$:

$$\frac{d\log_{10}(\dot{P})}{d\log_{10}(P)} = \frac{d\log_{10}(P)}{d\log_{10}(P)} - \frac{d\log_{10}(\tau)}{d\log_{10}(P)} - \frac{d\log_{10}(n-1)}{d\log_{10}(P)} = 1 - \frac{d\log_{10}(\tau)}{d\log_{10}(P)}. \tag{4.46}$$

where we recall that n is a proportionality constant so its derivative is zero. Then, a straight line of gradient $+1$ implies that $\frac{d\log_{10}(\tau)}{d\log_{10}(P)} = 0$; that is, it is a line of constant characteristic age with respect to the period. □

Proposition 4.10. *In a $P - \dot{P}$ diagram, straight lines of gradient -1 (i.e. going top left to bottom right) represent constant dipole field.*

Proof. Take the logarithm of both sides of Eq. (eq: NS properties: Estimate of polar magnetic field strength) and rearranges for $\log_{10}(\dot{P})$:

$$\log_{10}(B_{\text{pole}}) = -\log_{10}\left(R^3 \sin(\alpha)\right) + \frac{1}{2}\log_{10}\left(\frac{6\mu_0 c^3}{16\pi^3}\right) + \frac{1}{2}\log_{10}(I) + \frac{1}{2}\log_{10}(P) + \frac{1}{2}\log_{10}(\dot{P}), \tag{4.47}$$

$$\Rightarrow \log_{10}(\dot{P}) = -\log_{10}(P) + 2\log_{10}(B_{\text{pole}}) + 2\log_{10}\left(R^3 \sin(\alpha)\right) - \log_{10}(I) - \log_{10}\left(\frac{6\mu_0 c^3}{16\pi^3}\right). \tag{4.48}$$

[2] Which, of course, is the Chandrasekhar mass.

Differentiate with respect to $\log_{10}(P)$. The angle α is independent of the period, and we can assume that the mass and radius are too. Then, $I \sim MR^2$ is also independent and we obtain

$$\frac{d\log_{10}(\dot{P})}{d\log_{10}(P)} = -\frac{d\log_{10}(P)}{d\log_{10}(P)} + 2\frac{d\log_{10}(B_{\rm pole})}{d\log_{10}(P)} + 2\frac{d\log_{10}\left(R^3\sin(\alpha)\right)}{d\log_{10}(P)} - \frac{d\log_{10}(I)}{d\log_{10}(P)} - \frac{d\log_{10}\left(\frac{6\mu_0 c^3}{16\pi^3}\right)}{d\log_{10}(P)} \quad (4.49)$$

$$= -1 + 2\frac{d\log_{10}(B_{\rm pole})}{d\log_{10}(P)}. \quad (4.50)$$

Then, a straight line of gradient -1 implies that $\frac{d\log_{10}(B_{\rm pole})}{d\log_{10}(P)} = 0$; that is, it is a line of constant polar magnetic field with respect to the period. □

The Australia Telescope National Facility maintains the **ATNF Pulsar Catalogue**, a publicly available database of all measured pulsars (Manchester et al., 2005). It is available online at

https://www.atnf.csiro.au/research/pulsar/psrcat/

Data can be selected on the website and presented in a tabular format for the user to copy into a `CSV` file for offline analysis. We used the ATNF catalogue to produce a $P - \dot{P}$ diagram for all known pulsars, shown in Figure 4.3. We downloaded data with the headings `P0` (period), `P1` (period derivative), `BINARY` (information on binary companions, if the pulsar has one) and `Type` (categories of pulsar, including "AXP") and wrote the `Python` code

`Plot_ATNF_P0_and_P1_02.py`

to plot them on a diagram, categorised such that the AXPs and those with a binary companion are highlighted separately. For the lines of constant characteristic age, we assume a braking index $n = 3$. For the lines of constant polar magnetic field, we assume an angle $\alpha = 90°$ between the magnetic and rotation axes, a radius $R = 10$ km, a mass $M = 1.4\,M_\odot$ and a uniform sphere $I = \frac{2}{5}MR^2$.

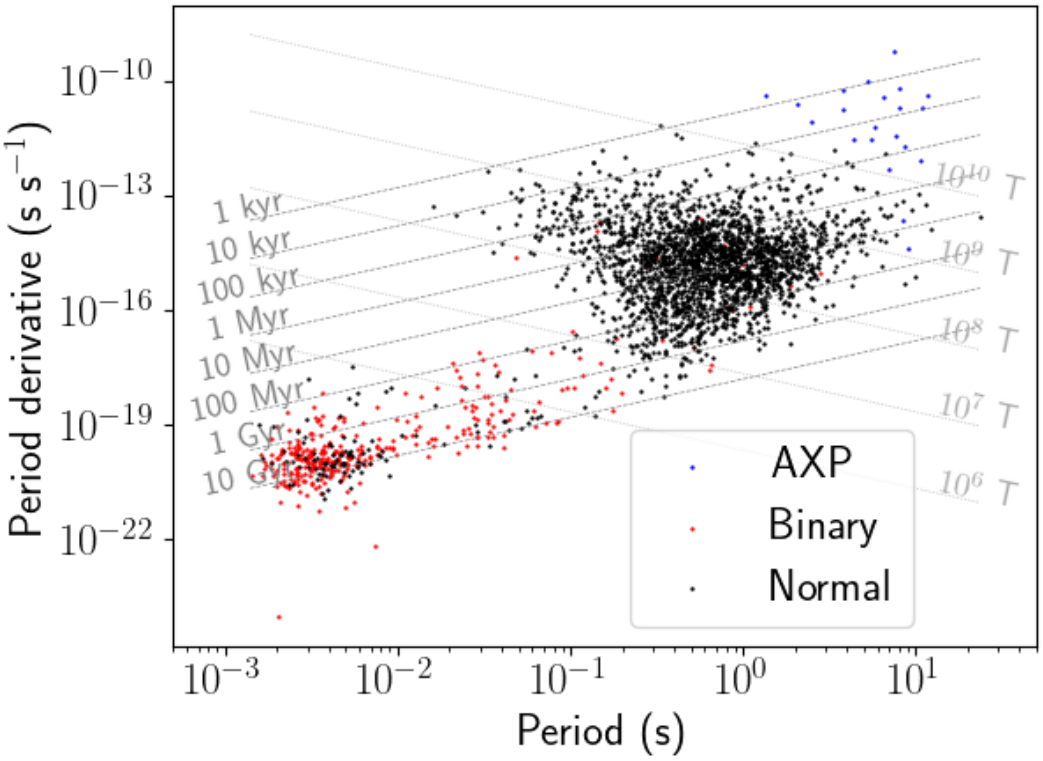

Figure 4.3: $P - \dot{P}$ diagram of all known pulsars as of 13 September 2024, using data from the ATNF Pulsar Catalogue v2.4.0.
Produced by code: `Plot_ATNF_P0_and_P1_v02.py`

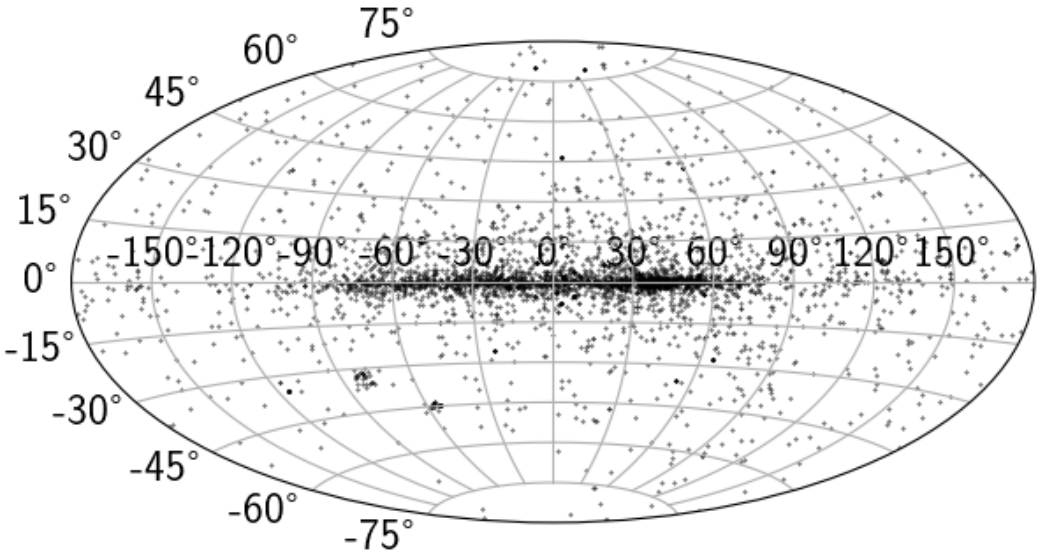

Figure 4.4: Galactic distribution of all known pulsars as of 11 September 2024, using data from the ATNF Pulsar Catalogue v2.4.0.
Produced by code: `Plot_ATNF_Gl_and_Gb.py`

5 Neutron stars as candidates for pulsars

Having outlined some key properties of neutron stars in the preceding chapters, let us use our results to argue for neutron stars as the most likely drivers behind pulsar activity, over alternatives such as pulsating main-sequence stars and white dwarfs.

Proposition 5.1. *The pulse length from a typical pulsar is too short for a main sequence star to be the source of its radiation.*

Proof. Suppose that the source of the pulses is a spherical object of radius R. Suppose that each burst is emitted isotropically over an infinitesimally small duration. Then, any observed duration of a pulse, lasting time Δt, can be assumed to be entirely due to the difference in light travel time from photons originating on different regions of its surface. We see the star as a disc in the night sky. Suppose that a photon originating from the centre of the disc travels a distance d_1 to Earth, while a photon originating from an edge travels a distance d_2. By trigonometry, these paths form a right-angled triangle with shorter sides d_1 and R, and hypotenuse d_2. Since the hypotenuse must be shorter than the other two sides combined, $d_2 < d_1 + R$. Then, if the light travel time from the centre of the star is $t_1 = \frac{d_1}{c}$, the light travel time from the edge is

$$t_2 \;=\; \frac{d_2}{c} \;<\; \frac{d_1 + R}{c} \;=\; t_1 + \frac{R}{c}. \tag{5.1}$$

The maximum pulse duration is the difference between these two extremal arrival times, $(\Delta t)_{\max} = \max(d_2 - d_1) = \frac{R}{c}$, and so the radius of the source is at most $R_{\max} = c(\Delta t)_{\max}$. Using $\Delta t = 14.5$ ms (Comella et al., 1969, Table 1), we obtain $R \approx 4.3 \times 10^6$ m $\approx 6 \times 10^{-3}\, R_\odot$, and so the source is far too small to be a star. Interestingly however, the original report by Hewish et al. (1968) quoted $\Delta t \approx 0.3$ s, resulting in $R \approx 9 \times 10^7$ m $\approx 0.1\, R_\odot$, which would not quite be small enough to discount a main-sequence star as a candidate. □

The result of Proposition 5.1 eliminates main sequence stars as candidates for pulsars. An approximate mass-radius relation for white dwarfs is (Ryan & Norton, 2010, Eq. (6.6))

$$R \;\approx\; 9.41 \times 10^6 \text{ m} \left(\frac{M}{M_\odot}\right)^{-1/3}, \tag{5.2}$$

and an alternative is (Nauenberg, 1972)

$$R \;\approx\; 7.83 \times 10^6 \text{ m} \left[\left(\frac{M}{1.44\, M_\odot}\right)^{-2/3} - \left(\frac{M}{1.44\, M_\odot}\right)^{2/3}\right]^{1/2}. \tag{5.3}$$

These are plotted in Figure 5.1. Note that the Nauenberg expression predicts radii which decline more rapidly approaching the Chandrasekhar limit than the Ryan-Norton expression, thus accounting for the star *overcoming* electron degeneracy pressure. It has the unphysical limit $\lim_{M \to 1.44\, M_\odot}(R) = 0$ and does not give real radii for $M > 1.44\, M_\odot$, representing the moment at which electron degeneracy pressure is overcome and a new model (i.e. a neutron star) is needed. According to these mass-radius relations, a theoretical pulsating white dwarf would still have a radius consistent with observed pulsar emission as calculated in Proposition 5.1. We require further restrictions in order to eliminate white dwarfs as candidates. According to the neutron star mass-radius relation in Eq. (4.42), theoretical pulsating neutron stars would also be consistent with pulsars. However, white dwarfs and neutron stars are too dense to pulsate, a claim we make without proof.

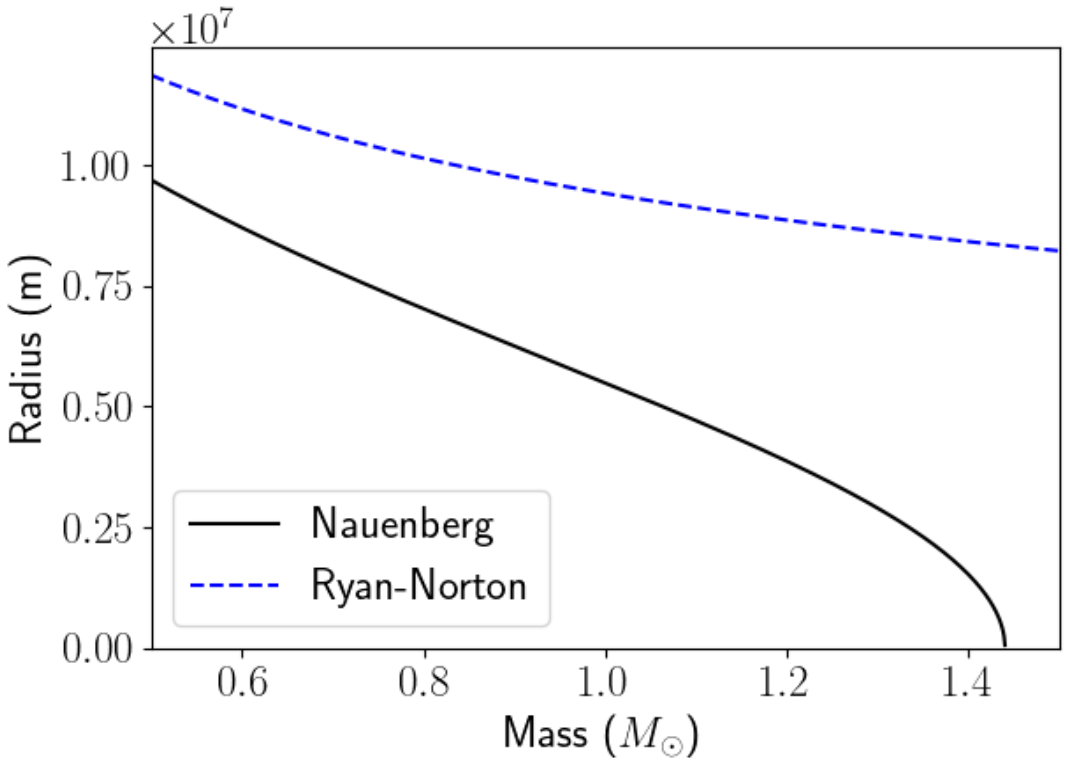

Figure 5.1: Two white dwarf mass-luminosity relations, described in the text.
Produced by code: `Plot_White_Dwarf_Mass_Luminosity_Relations.py`

Proposition 5.2. *An object of mass M and radius R cannot rotate with a period less than*

$$P_{\min} = 2\pi\sqrt{\frac{R^3}{GM}} \tag{5.4}$$

without breaking apart.

Proof. Consider a test mass m on the surface of a sphere with mass M and radius R rotating at angular frequency Ω. As the rotation increases, the centripetal force on the mass will eventually overcome the gravitational attraction keeping the mass on the surface; this is a good approximation for a star breaking up. Then, the maximum rotation rate that the object can have without breaking up is given when the gravitational and centripetal forces on the test mass balance:

$$\frac{GMm}{R^2} = mR\Omega_{\max}^2. \tag{5.5}$$

The test mass cancels and the expression rearranges to

$$\Omega_{\max} = \sqrt{\frac{GM}{R^3}}. \tag{5.6}$$

Finally, angular velocity and rotation period are related by $\Omega = \frac{2\pi}{P}$, so a maximum Ω corresponds to a minimum P. Substituting this, we obtain the given result. □

The above expression rearranges to give a maximum radius

$$R_{\max} = \left(\frac{1}{4\pi^2} GM P_{\min}^2\right)^{1/3}. \tag{5.7}$$

Substituting the Chandrasekhar mass and a nominal period $P = 0.1$ s, we obtain $R_{\max} \approx 3.6 \times 10^5$ m . This is far smaller than expected white dwarf radii from the mass-radius relations in Eqs. (5.2) and (5.3), ruling them out as candidates, but well within tolerance for the expected radii of neutron stars.

Proposition 5.3. *The Crab Pulsar is emitting sufficient energy to supply the observed luminosity of its surrounding nebula.*

Proof. Assume that the Crab Nebula is entirely rotation-powered. That is, all of the energy it receives is due to the energy lost by its central pulsar as it slows down. Then, as we saw in the proof of Proposition 4.1, the power fed by the pulsar to the nebula is $P = I\Omega\dot{\Omega}$. The Crab pulsar has angular frequency $\Omega \approx 190$ s^{-1} and angular frequency derivative $\dot{\Omega} \approx -2.43 \times 10^{-9}$ s^{-2} (Lyne et al., 1993, Table 1)[1] Assuming a uniform sphere with $M = 1.4\,M_\odot$ and $R \approx 10$ km such that $I = \frac{2}{5}MR^2 \approx 1.11 \times 10^{38}$ kg, we obtain $P = -5.13 \times 10^{31}$ W. This agrees reasonably well with calculations of the luminosity of the Crab Nebula at around 5×10^{31} W (Ryan & Norton, 2010, §7.3.3). □

Having argued that neutron stars, not pulsating main sequence stars or white dwarfs, are the most likely candidates for pulsars, our final calculation above has demonstrated that a rotating neutron star is indeed a viable candidate for the central engine driving the observed luminosity of an emission nebula such as the Crab.

[1] The authors quote linear frequencies, so we multiply the values given in the cited source by 2π.

6 History of study of giant flares

Table 6.1: Rough maximum wavelengths and minimum photon energies for X-ray and gamma-ray bands. The wavelengths and energies are related by $E = \frac{hc}{\lambda}$.

Band	Wavelength λ (m)	Photon energy E (keV)
Soft X-ray	$10^{-8} = 10$ nm	0.1
Hard X-ray	$10^{-10} = 100$ pm	10
Gamma-ray	$10^{-11} = 10$ pm	100

In this chapter, we provide a brief review of the history of observation of giant flares and their connection to a hyper-magnetic class of neutron star known as a magnetar. An in-depth review of magnetars was produced by Kaspi & Beloborodov (2017), while a description aimed at a general audience was written by Kouveliotou et al. (2003).

Although neutron stars are most commonly associated with radio pulsars, some also emit strongly in X-rays. Observations over the years have led to the classification of many X-ray-emitting neutron stars as **soft gamma repeaters (SGRs)** or **anomalous X-ray pulsars (AXPs)**, though it is now widely held that these classes are both part of the same unified group. They are often associated with nearby supernova remnants, making them particularly young NSs (10^4 yr). X-ray-emitting NSs have four key observational properties:

1. Isolated (i.e. no binary companion).
2. Persistent but variable X-ray emission comprising a two-component spectrum: a soft (~ 0.5 keV) black body and a hard (~ 100 keV) tail.
3. Long spin periods $P \sim 2 - 12$ s compared to radio pulsars.
4. High period derivatives $\dot{P} \sim 10^{-13} - 10^{-11}$ s s^{-1}.

We have seen from Proposition 4.2 that Properties 3 and 4 imply an inferred polar magnetic field $B_{\text{dipole}} \sim 10^{14} - 10^{15}$ G for magnetars and $\sim 10^{12} - 10^{13}$ G for pulsars. The magnetic fields of magnetars are among the strongest currently known in the universe. Although supported by theory and simulations, it must be noted that no direct measurement of such a strong magnetic field has yet been made.

X-ray emission consists of a persistent component as well as transient behaviour in the form of bursts. Bursts occur on a range of different time- and energy-scales, and are roughly categorised accordingly as **short bursts**, **intermediate bursts** and **giant flares**.

So far, four giant flares have been detected, summarised in Table 6.2 below. They have all exhibited similar profiles, namely a hard initial spike lasting $0.1 - 0.4$ s followed by a softer tail lasting a few minutes. The tail is pulsed at the neutron star period. The total energy radiated is around 10^{44} erg, or around the energy radiated by the Sun in 10^3 yr. A second burst is observed around a day afterward, with total luminosity around an order of magnitude lower than the first. The most recent giant flare occurred in the starburst galaxy M82, too distant for the softer pulsed tail to be visible, but the initial spike was indeed detected (Mereghetti et al., 2024).

Giant flares seem to correspond to **braking glitches**, or large spikes in the spin-down rate which is otherwise roughly constant before and after. For example, the 27 August 1998 flare was accompanied by a glitch which caused $\frac{\Delta P}{P} \sim 10^{-4}$ (Woods et al., 1999) and released roughly 10^{41} erg, or around 0.5% of the energy that the giant flare would release. Thompson et al. (2000) presented a possible mechanism for the braking glitch.

The **Eddington luminosity** of an object is the maximum luminosity that it can possess without breaking itself apart. At higher luminosities, the radiation pressure becomes strong enough to overcome the gravitational binding energy of the object. Let us assume naïvely that the opacity around a neutron star is due to neutral hydrogen, which allows us to quantify the opacity as $\kappa = \frac{\sigma_{\text{T}}}{m_p} \approx 0.0398$ m^2 kg^{-1}. Even if this is unlikely, it

allows us to obtain a rough estimate and the true source of opacity may not yield a significant difference. Then, we have

$$L_{\rm Edd} = \frac{4\pi cGM}{\kappa} \approx 1.76\times10^{38}\,\frac{M}{1.4\,M_\odot}\left(\frac{\kappa}{0.0398\ {\rm m^2\ kg^{-1}}}\right)^{-1}\ {\rm erg\ s^{-1}}. \quad (6.1)$$

The measured energies of giant flares hence indicate that they are super-Eddington events.

The magnetic fields of magnetars exceed the **Schwinger limit**[1]

$$B_{\rm Q} = \frac{m_e^2c^2}{\hbar e} \approx 4.4\times10^9\ {\rm T} = 4.4\times10^{13}\ {\rm G}, \quad (6.2)$$

and so the effects of quantum electrodynamics (QED) start to become apparent. An overview of the most important phenomena at such high field strengths is given by Duncan (2000).

Table 6.2: List of observed giant flares.

Date	Object	Location	Satellite	Example citation
5 March 1979	SGR 0526-66[2]	N49, LMC	Venera 11, 12 / KONUS	Mazets et al. (1979)
27 August 1998	SGR 1900+14	Milky Way	Wind / KONUS Ulysses BeppoSAX	Mazets et al. (1999) Hurley et al. (1999) Feroci et al. (1999)
27 December 2004	SGR 1806-20	1806-20 (open cluster)	INTEGRAL	Hurley et al. (2005)
15 November 2023	(GRB 231115A)	M82	INTEGRAL	Mereghetti et al. (2024)

6.1 SGR 0526-66

The first observed giant flare inspired much research in order to understand the mechanism behind it; the burst has been covered so many times in the literature that it is commonly referred to simply as **the (1979) March 5 event**.

The 1979 March 5 flare was detected in the energy range $50-150$ keV ($8-25$ pm). It featured an initial burst with rise time 15 ms and decay 150 ms. The hard tail had energy $400-500$ keV, which the authors identified as electron-positron annihilation with a gravitational redshift due to an object with $M = M_\odot$ and $R = 10$ km, giving further evidence that the burst was from a neutron star. However, it was initially speculated to be an accreting binary. The pulsed tail was similar in shape to continuous X-ray sources, so was identified to be a neutron star. It was fitted to a burst spectrum with $k_{\rm B}T = 30$ keV, or within the hard X-ray band. The second burst was detected around 14 hours later. It followed a similar profile to the first, but at around 1% of the intensity.

The authors identified that the burst came from the Large Magellanic Cloud and speculated that it originated from the Brasil Nebula N49. Armed with a known distance to the LMC, they estimated luminosities of 5×10^{44} erg s^{-1} in the initial phase and 3.6×10^{44} erg s^{-1} in the pulsating phase, for a total released energy $> 4.6\times10^{44}$ erg. Later, Cline et al. (1982) used the fact that the radiation had been detected by multiple satellites throughout the solar system at slightly different times to locate it within a box of size 0.1 arcmin^2 (i.e. side 19 arcsec). By comparison, N49 has angular diameter 84 arcsec (Badenes et al., 2010, Table 1).

[1]The definition in cgs units requires an extra value of c so that $B_{\rm Q} = \frac{m_e^2c^3}{\hbar e}$, so to avoid confusion we first quote the definition in SI units and then convert it to gauss.

[2]Also referred to by some sources as SGR 0525-66.

Although the pulsation period of the latter spectrum and the identification with N49 were strong evidence of a neutron star, there were still issues to be resolved:

1. The period of 8 seconds was far longer than for most known pulsars. The magnetar scenario neatly resolves this: Eq. (4.27) with $B = 10^{15}$ G and other quantities taking fiducial values yields a time $\tau \sim 5$ kyr for a magnetar to spin-down to $P = 8$ s, in good agreement with the estimated 4.8 kyr age of N49 (Park et al., 2012).

2. The NS associated with N49 is significantly offset from the centre of the SNR, implying a very high kick velocity 10^3 km s^{-1} when it was formed. Duncan & Thompson (1992) speculated that anisotropies in the extremely violent neutrino emission during NS formation could lead to an anisotropic momentum loss that scales with B, with magnetar-strength magnetic fields easily yielding the required kick velocity.

3. The super-Eddington luminosity could not be explained; accretion had only previously resulted in outbursts slightly above $L_{\rm Edd}$. Besides, X-ray-emitting neutron stars are not observed to have binary companions. However, Paczyński (1992) argued that strong magnetic fields reduce the opacity of electrons to outgoing radiation, and calculated that luminosities up to $10^4\,L_{\rm Edd}$ could be sustained if $B \sim 10^{14}$ G. Thus, the luminosity may be explained not by accretion but by a starquake releasing a magnetic flare.

The radiation from the burst was around 100 times stronger than previously detected gamma-ray bursts and the source was found to emit 16 further bursts over the ensuing four years (Golenetskii et al., 1984), whereas traditional GRBs are expected to be standalone events. Another source of repeated gamma-ray bursts, later named **SGR 1806-20**, was also detected (Atteia et al., 1987), and the term **soft gamma repeater** was coined to distinguish these objects from GRBs. The 1979 March 5 event was named **GRB 790305b** and its source named **SGR 0526-66**.

The magnetar scenario was laid out by Thompson & Duncan (1995, 1996). The discovery of a pulsar within SGR 1806-20 by Kouveliotou et al. (1998) finally gave credence to this scenario.

A statistical analysis by Cheng et al. (1996) and verified by Göğüş et al. (1999, 2000) highlighted the similarity of the energy distribution of SGR outbursts to those of earthquakes on Earth, giving evidence for a system of **self-organised criticality**. These results gave credence to the starquake model of magnetar eruptions.

6.2 SGR 1806-20

Kulkarni & Frail (1993) compared approximate locations of bursts from SGR1806-20, which had been dormant for around ten years, with a supernova catalogue, and found it roughly coincident with the radio nebula G10.0-0.3. They used this, combined with the association of SGR 0526-66 with N49, as further evidence that SGRs are young neutron stars. The SGR became active again in September that year, being detected by the BATSE experiment on NASA's Compton Gamma Ray Observatory (Kouveliotou et al., 1994), and this led Murakami et al. (1994), who had detected the same burst with the ASCA satellite, to definitively locate it with this SNR. Follow-up radio images of G10.0-0.3 by Kulkarni et al. (1994) suggested that the SGR1806-20 is an isolated pulsar with both steady-state and transient emission which power the growth of the SNR. As with SGR0526-66, the pulsar was found to be offset from the centre, implying a large kick velocity. Kouveliotou et al. (1998) studied the persistent X-ray radiation of the SGR, finding $P = 7.47$ s and $\dot{P} = 8.25 \times 10^{-11}$. Eqs. (4.10) and (4.12) then yield $B_{\rm pole} \approx 2 \times 10^{15}$ G and $\tau \approx 1500$ yr, consistent with values found for SGR 0526-66 and further strengthening the burst mechanism suggested by Thompson & Duncan (1995).

6.3 SGR 1900+14

Despite having been studied intensely for decades and the results used as conclusive evidence that SGRs are magnetars, the second giant flare to be observed did not originate from SGR 1806-20.

On 27 August 1998, an extremely intense burst of gamma rays were detected by several satellites throughout the Solar System (Mazets et al., 1999; Hurley et al., 1999; Feroci et al., 1999). Although the total radiated power was only around one tenth that of SGR0526-66, the proximity of this second flare meant that the detected radiation was far more intense - the most intense gamma-ray burst ever detected from beyond the Solar System, as of 2003.

The profile of the flare was remarkably similar to that of the 1979 event: a rapid rise, followed by a tail lasting a few hundred seconds and pulsed at a period of 5.16 s.

Wachter et al. (2008) performed observations that associated the SGR with a cluster of massive stars, which had previously been suggested by Vrba et al. (1996). This gave further evidence that magnetars are formed from massive stars.

7 The magnetar scenario of giant flares and SGR bursts

Over a series of papers culminating in Thompson & Duncan (1995), Christopher Thompson and Robert Duncan argued that the source of giant flares was a neutron star with a decaying ultra-strong magnetic field. They coined the term **magnetar** to describe such an object and their model has become known as the **magnetar scenario** of giant flares. In particular, they argued the following points:

1. SGR bursts show no correlation between energy released and time between subsequent bursts. This eliminates accretion as a possible trigger. Additionally, optical counterparts are too faint to allow for the presence of a sizeable accretion disk (e.g. Hulleman et al., 2000).

2. Giant flares have highly super-Eddington luminosities 10^3 - $10^4\,L_{\rm Edd}$, enough to rip the source apart if it were supported by gravity alone. Since the object is preserved, a mechanism must exist which confines the emitting plasma around it. The proposed mechanism was magnetic confinement.

3. The more frequent short and intermediate bursts may be caused by fracturing of the neutron star crust, a process that is particularly common in young magnetars.

The required magnetic field strengths $10^{14} - 10^{15}$ G were unprecedented in the literature, but Thompson and Duncan gave a total of six independent arguments for the decaying ultra-strong magnetic field hypothesis:

1. The supernova remnant N49 associated with SGR 0526-66 has age $\sim 10^4$ yr and period 8.0 s. Such a spin-down requires $B_{\rm dipole} \sim 6 \times 10^{14}$ G.

2. If magnetic fields are the cause, the magnetic free energy of a neutron star must be far greater than the total energy of a giant flare. Then, the energy $\sim 5 \times 10^{44}$ erg released in the March 5 event implies a magnetic field $B_{\rm dipole} \gg 8 \times 10^{13}$ G.

3. For $B_{\rm surface} \gtrsim 3 \times 10^{15}$ G, diffusion of magnetic field lines through the crust and the core occur on a timescale of 10^4 yr, comparable to the age of SGRs.

4. Strong magnetic fields suppress the electron scattering opacity of photons (e.g. Uzdensky, 2011), allowing more radiation to escape and resulting in super-Eddington luminosities. In particular, $B_{\rm surface} \gtrsim 3{\times}10^{14}$ G allows the $\sim 10^4\,L_{\rm Edd}$ observed in SGRs. This argument was previously made by Paczyński (1992) as evidence that the March 5 event was caused by a strongly magnetised NS.

5. A crustal magnetic field $B_{\rm crust} \gtrsim 1 \times 10^{15}$ G is required for the magnetic energy of the crust to be high enough to power the persistent X-ray emission of SGR 0526-066.

6. The March 5 event began with a spike in the hard X-rays of duration 0.15 ms. If $B \gtrsim 7 \times 10^{14}$ G, this is comparable to the **Alfvén crossing time**[1] of the star.

The Thompson-Duncan model has become widely accepted, although the extreme rarity of giant flares makes it difficult to obtain enough observational data to conclusively verify it. It has even been suggested that giant flares may be a possible progenitor of some gamma-ray bursts (e.g. Burns et al., 2021).

[1] The time taken by an MHD wave to cross an object.

7.1 Issues extending the Goldreich-Julian pulsar model to magnetars

The magnetic field lines of a rotating object are bound to its surface at locations known as **footpoints**, and this causes the magnetosphere to co-rotate with it. The classical model of a pulsar, presented by Goldreich & Julian (1969), is a rotating dipole whose co-rotating magnetic field is potential (current-free, §9.3) except for a narrow bunch of lines which extend out to the **light cylinder** $R_{\rm LC}$. This is the radial distance at which a particle co-rotating with the pulsar must move at the speed of light to keep up. Nothing beyond the pulsars's light cylinder can co-rotate with it, so field lines extending beyond it necessarily remain open and dissipate energy. Since charged particles flow along magnetic field lines in the absence of other forces, this causes the emission of synchrotron electrons that can power the luminosity of a host nebula if one is present. See Figure 22.1 for a diagram of the magnetic field structure of a non-rotating dipole. That of a typical neutron star is shown in Figure 7.1.

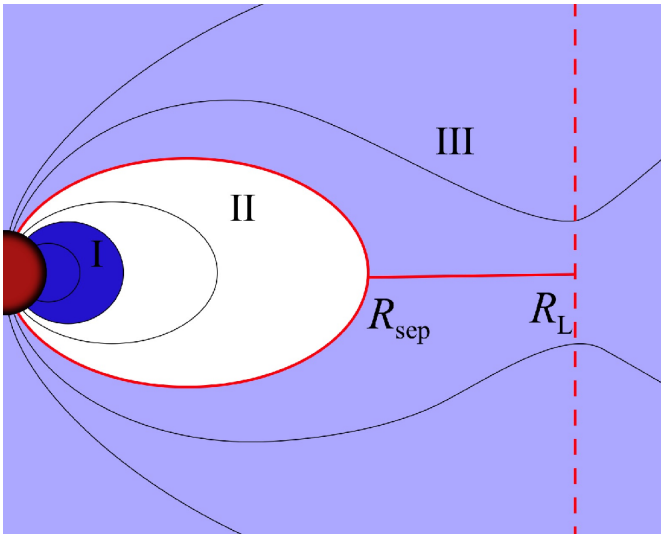

Figure 7.1: Magnetic field lines of a rotating neutron star. The force-free approximation holds within the last closed field line (regions I and II). Note that many models, including ours, assume that the last closed field line intersects the light cylinder, i.e. $R_{\rm sep} = R_{\rm LC}$. This is Figure 5 of Glampedakis et al. (2014); reproduced with permission.

Can the Goldreich-Julian model be applied to magnetars? Dipoles are axisymmetric, but **Cowling's neutral point theorem** states that a steady, axisymmetric dynamo cannot exist (e.g. Davidson, 2001, §6.2.3). Therefore, the magnetic field of a magnetar must vary from that of a dipole if magnetic field amplification is to occur by convection (see Proposition 4.7 and the discussion following it).

Proposition 7.1. *The magnetic field of a dipole is purely poloidal.*[a]

[a] See §9.2 for a discussion on poloidal and toroidal vectors.

Proof. The magnetic field of a dipole is (e.g. Griffiths, 2017, Eq. (5.88))

$$\mathbf{B}(\mathbf{r}) \;=\; \frac{\mu_0 m}{4\pi}\left(\frac{2\cos(\theta)}{r^3}\,\mathbf{e}_r + \frac{\sin(\theta)}{r^3}\,\mathbf{e}_\theta\right), \tag{7.1}$$

where m is the magnitude of its magnetic dipole moment. This has no ϕ-component and its r,θ-components are independent of ϕ so, using the expression in Eq. (9.4), its curl may only have a ϕ-component. The explicit expression is

$$\boldsymbol{\nabla}\times\mathbf{B} \;=\; \frac{\mu_0 m}{4\pi}\,\frac{4\cos(\theta)-3\sin(\theta)}{4}\,\mathbf{e}_\phi. \tag{7.2}$$

We stated in §9.2 that the curl of a poloidal field is purely azimuthal, so it follows that our $\mathbf{B}$ is poloidal. □

It has been shown (e.g. Braithwaite, 2009, and references therein) that the internal magnetic field of a NS must contain both poloidal and toroidal components because each of these are unstable on their own. Lander & Jones (2011a,b) carried out numerical simulations on neutron stars with purely toroidal and purely poloidal magnetic fields, confirming that both were unstable but finding that rotation has a stabilising effect in both cases. The strong internal magnetic fields of magnetars cause large deviations of the external field from that of a dipole, and their slow rotation means that instabilities are not suppressed if the field is assumed to be purely poloidal as for pulsars. This may explain why the model remains acceptable for pulsars, even if it breaks down for magnetars.

A particular difficulty in numerical simulations of NS magnetospheres is that the magnetic field becomes discontinuous across $R_{\rm LC}$. Full time-dependent solutions have so far proved elusive, but so-called **steady-state solutions**, those that create a snapshot of the magnetic field at a given moment in time, have been found. Michel (1973) and Scharlemann & Wagoner (1973) presented the **pulsar equation**, which can be solved to describe the field everywhere in the magnetosphere at a given point in time. Contopoulos et al. (1999) were the first to solve it for a rotating dipole with a large magnetospheric charge density. Numerical steady-state solutions for dipole in vacuum with arbitrary alignment between its magnetic and rotational axes were found by Deutsch (1955).

It must also be noted that Goldreich & Julian (1969) only ever presented a qualitative picture of how a pulsar magnetosphere might appear; their model had little mathematical grounding and certainly was not obtained by solving some differential equation such as the pulsar equation or Grad-Shafranov equation (§9.4). It is telling that, in the decades since, authors have still struggled to produce stable magnetospheric models even as the complexity of numerical models and computational power has increased. As such, although we may expect at a first approximation to obtain magnetospheres resembling the Goldreich-Julian model with our own code later in this paper, there is no certainty that such a model is the only correct stable configuration. If (when) we run into issues regarding the convergence of the model, this may be worth remembering. Recently, Contopoulos et al. (2024) explored alternative solutions for the magnetic field around a pulsar.

7.2 Crustal fractures and magnetic reconnection

Thompson & Duncan (1996) explained the persistent X-ray emission of SGRs and AXPs as follows. The internal magnetic field of a NS diffuses outward through the core by **ambipolar diffusion** and through the crust by **Hall drift** (Goldreich & Reisenegger, 1992). In magnetars, the internal magnetic field is so strong that it can fracture the crust as it does so. This suddenly displaces the footpoints, imparting a twist (an Alfvén wave) in the magnetospheric field lines located near the NS surface. At the same time, electrical currents are passed from the interior to the exterior. These currents push a twist in a given magnetic flux tube further outwards, so that it becomes distributed more evenly along the tube and hence enters a lower-energy configuration (see Fig. 1 in Thompson et al., 2002). The external current is driven by stripping charges from the surface, and this results in a pair-production avalanche in the magnetosphere. A plasma corona is created, which organises into a quasi-steady state that can persist for 1 - 10 yr; this may be responsible for the observed persistent X-ray emission (Beloborodov & Thompson, 2007). A quantitative description of the buildup of stresses in a magnetar crust was given by Lander (2016).

If the Alfvén wave is ejected to a maximum radius large enough that the magnetic field strength there is low, the result is instead a relativistic outflow in the form of a magnetic reconnection event. Thus, short and intermediate bursts are caused by relaxation of twisted magnetic field lines, while giant flares are caused by large-scale magnetic reconnection events.

7.3 Modelling magnetospheric twists in axisymmetry

Despite the difficulties in extending the Goldreich-Julian pulsar model to magnetars as discussed in §7.1, much progress has been made in the literature under the dipole approximation.

Consider an object with an axisymmetric magnetic field, and let us twist its magnetic field footpoints about the symmetry axis. Lynden-Bell & Boily (1994) showed that, after $\frac{1}{\sqrt{3}} \approx 0.58$ turns,[2] the magnetic field remains finite but the current becomes infinite and a large-scale discharge of magnetic energy is triggered. Aly (1984, 1991) and Sturrock (1991) conjectured that the energy of the sheared field cannot exceed that of an open field,[3] and this was supported by Wolfson (1995).

Thompson et al. (2002) modified the Goldreich-Julian model to allow for a current to flow across the NS surface, a consequence of the outwardly-diffusing internal magnetic field. They considered an axisymmetric NS in which the twist was global, not local: one hemisphere was rotated relative to the other. They looked for self-similar solutions, resulting in a sequence of equilibria similarly to the methods of Lynden-Bell & Boily (1994) and Wolfson (1995). They derived an equation for the twist angle reached by a field line as a function of the polar angle of its footpoints and found a maximum value of π. This is below the angle for explosive energy release found by Lynden-Bell & Boily (1994), albeit in a different physical setup, so is reasonable for persistent X-ray emission. A number of their predictions from the model corroborated with observations; examples are given below.

- A persistent current can be maintained by stripping electrons and ions from the NS surface, prolonging the life of the plasma corona for ~ 30 yr. This is in line with the persistent X-ray emission from SGR 0526-66, which is decaying but still detectable even though it has not undergone a strong burst since 1983 (Aptekar et al., 2001; Park et al., 2020).
- Spin-down rate is correlated to twist angle, for a given polar magnetic field strength.
- Emission from the NS surface near the equatorial plane experiences strong resonant Compton scattering at the cyclotron resonance, while emission from near the poles undergoes little scattering (Figure 5 in their paper). Large twists yield multiple scatterings and a hard thermal X-ray tail in the spectrum. Spectral hardness is correlated with observed burst activity (Marsden & White, 2001) and observed spectra have a black-body-plus-power-law shape (Mereghetti, 2008); resonant Compton scattering can explain both these trends (Turolla et al., 2015).

7.4 Understanding magnetic reconnection from other areas of physics

Analogies can be drawn with models of solar flares, in which the magnetic footpoints are displaced by turbulent convection below the photosphere. This opens the intriguing possibility of studying giant flare mechanics through the wealth of observational data from the Sun. A comprehensive review of the magnetohydroynamics of solar flares was presented by Shibata & Magara (2011), while Wiegelmann & Sakurai (2021) detailed the treatment of force-free magnetic fields in solar physics. Magnetic reconnection in the context of solar physics was discussed by Mikić & Linker (1994). Magnetic reconnection is also commonly studied in the contexts of Earth's magnetosphere and of nuclear fusion.

However, care must be taken: these alternative applications represent low-energy environments in which only the charged particles (electrons and ions) are important. However, in high-density plasmas, electron-positron pairs are created so vigorously that they cover the reconnection region in an optically thick region, trapping any photons within and causing radiation pressure to dominate. As a result, the effects of radiative transfer must also be considered when studying magnetic reconnection in NSs (Uzdensky, 2011). The magnetic fields of magnetars are strong enough for spontaneous pair creation to occur (for a review, see Daugherty & Harding, 1983), so these effects are even more important in the context of magnetars.

[2] A twist angle $\frac{2\pi}{\sqrt{3}} \approx 1.15\,\pi \approx 3.63$ rad.

[3] Specifically, for a force-free field (see §9.3 and Chapter 20).

7.5 Simulations of reconnection events

Parfrey et al. (2012) developed the code PHAEDRA in order to simulate relativistic magnetospheres under the force-free condition. They reasoned that pre-existing methods, which evolve the magnetic field using finite differences or finite volumes, are prone to numerical noise which can erroneously lead to instabilities and reconnection events. To combat this, they developed a method in which the fields are expanded into orthogonal basis functions, similar to contemporary spectroscopic codes. This is the method that we intend to develop throughout the remainder of the report.

The authors used PHAEDRA to simulate reconnection in strongly twisted magnetospheres (Parfrey et al., 2013). They defined the **shear** ψ to be the azimuthal separation between the two footpoints of a magnetic field line,[4] and used performed calculations using various **shearing profiles** which each concentrated the shear within a certain latitude. The shear was imparted slowly through a sequence of equilibrium states. For all models, the shear caused the magnetic field lines to expand to larger radii, with the radial extent of field lines found to be quite sensitive to the shear angle. Departures from equilibrium were noted at $\psi \gtrsim 3$ rad, indicating that the extent of field lines was such that the weaker magnetic field at their maximum extent could not retain Alfvén waves. The authors concluded that every shearing profile possesses a critical shear ψ_{crit} beyond which a magnetic reconnection event is inevitable.

Continued twisting has a stabilising effect, allowing the magnetosphere to temporarily possess supercritical twist amplitudes $\psi > \psi_{\text{crit}}$; however, a reconnection event is inevitable once ψ_{crit} is exceeded. The same tests were performed on rotating stars, with the conclusion that rotating stars must also possess a critical shear. If the shear is continued after a reconnection event, a **twisted reservoir** forms from those field lines which did not fully open, and after further shearing this will also expand explosively. Reconnection events exhibit a spike in the spindown torque on the NS, corroborating the second prediction of Thompson et al. (2002) in §7.3.

7.6 Note on general relativity

We will wish to incorporate general relativistic corrections: the work of Thorne & MacDonald (1982); MacDonald & Thorne (1982) will allow us to do so while still invoking the notion of an "absolute time". This will allow us to evolve the electric and magnetic fields as vectors without the need to introduce an electromagnetic field tensor. We will be able to develop a non-relativistic version of our code and later add general relativistic corrections when required, without the need for a full tensor treatment from the outset.

The abovementioned code PHAEDRA was developed in a way that would allow for general relativistic treatments in future versions; these have been performed in the contexts of black hole magnetospheres and accreting pulsars (Parfrey, 2019; Parfrey & Tchekhovskoy, 2017).

[4] The coordinate system is aligned with the magnetic axis, so an untwisted field line should start and end at the same azimuthal angle in axisymmetry. The authors also used the term **twist** to refer to ψ.

7.7 Currently known magnetars

A total of 30 magnetars are known as of May 2022, summarised in Table 7.1. A catalogue of magnetars was compiled by Olausen & Kaspi (2014) and is maintained online in the **McGill Online Magnetar Catalog**;[5] a list of observed bursts is maintained in the **Amsterdam Magnetar Burst Library**.[6]

Table 7.1: List of currently known magnetars.

Discovered	Confirmed	Object	Example citation
2001	2001	CXOU J010043.1-721134	Lamb et al. (2002)
1978	2000	4U 0142+61	Hulleman et al. (2000)
2009	2010	SGR 0418+5729	van der Horst et al. (2010)
2008	2010	SGR 0501+4516	Göğüş et al. (2010b)
1979	1992	SGR 0526-66	Duncan & Thompson (1992)
1979	2002	1E 1048.1-5937	Wang & Chakrabarty (2002)
1980	2007	1E 1547.0-5408[7]	Gelfand & Gaensler (2007)
2010	2010	PSR J1622-4950	Levin et al. (2012)
1998	2004	SGR 1627-41	Wachter et al. (2004)
2006	2006	CXOU J164710.2-455216	Muno et al. (2006)
1996	2003	1RXS J170849.0-400910	Israel et al. (2003)
2008	2010	CXOU J171405.7-381031	Halpern & Gotthelf (2010)
2013	2013	SGR J1745-2900	Kennea et al. (2013)
1979	2003	SGR 1806-20	Ibrahim et al. (2003)
2004	2004	XTE J1810-197	Ibrahim et al. (2004)
2020	2020	Swift J1818.0-1607	Lower et al. (2020)
2011	2011	Swift J1822.3-1606[8]	Livingstone et al. (2011)
2010	2010	SGR 1833-0832	Göğüş et al. (2010a)
2011	2011	Swift J1834.9-0846	Kargaltsev et al. (2012)
1985	2004	1E 1841-045	Wachter et al. (2004)
2014	2014	3XMM J185246.6+003317	Zhou et al. (2014)
1979	1999	SGR 1900+14	Kouveliotou et al. (1999)
2014	2014	SGR 1935+2154	Kozlova et al. (2016)
1980	2001	1E 2259+586	Hulleman et al. (2001)
2016	(Candidate)	SGR 0755-2933	Doroshenko et al. (2021)
2000	(Candidate)	SGR 1801-23	Cline et al. (2000)
2003	(Candidate)	SGR 1808-20	Molkov et al. (2005)
2012	(Candidate)	AX J1818.8-1559	Mereghetti et al. (2012)
1998	(Candidate)	AX J1845.0-0258	Tam et al. (2006)
2011	(Candidate)	SGR 2013+34	Sakamoto et al. (2011)
2000	(Candidate)	PSR J1846-0258	Kumar & Safi-Harb (2008)

[5] http://www.physics.mcgill.ca/~pulsar/magnetar/main.html.
[6] https://staff.fnwi.uva.nl/a.l.watts/magnetar/mb.html.
[8] Also referred to as PSR J1550–5418.
[8] Also referred to as SGR 1822-1606.

Part II

Mathematical results

8 Coordinate systems

8.1 General 3D coordinates

We use the term **curvilinear coordinates** to refer to coordinate systems in Euclidean space. In 3D, curvilinear coordinates (u_1, u_2, u_3) may be derived from Cartesian coordinates by invertible one-to-one transformations

$$x = x(u_1, u_2, u_3), \tag{8.1}$$
$$u_1 = u_1(x, y, z), \tag{8.2}$$

and so on. Let P represent some point (x, y, z) in a 3D Cartesian coordinate system, and let $\mathbf{r}$ be its position vector:

$$\mathbf{r} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{8.3}$$

where the Cartesian unit vectors are defined by the derivative of the location of P with respect to each coordinate:

$$\mathbf{i} = \frac{\partial \mathbf{r}}{\partial x}, \tag{8.4}$$

and so on. Applying this to our general curvilinear system at P, we can define three basis vectors:

$$\mathbf{h}_i = \frac{\partial \mathbf{r}}{\partial u_i}, \quad i = 1, 2, 3. \tag{8.5}$$

These may not have unit length and may not be orthogonal. If they *are* orthogonal, we define the **Lamé coefficients** as $h_i = |\mathbf{h}_i|$, so that the **orthonormal basis vectors** are

$$\mathbf{e}_i = \frac{\mathbf{h}_i}{h_i} = \frac{\partial \mathbf{r}/\partial u_i}{\left|\partial \mathbf{r}/\partial u_i\right|}. \tag{8.6}$$

Coordinates and Lamé coefficients for the three most common 3D coordinate systems are given in Table 8.1.

Table 8.1: Coordinates u_i and Lamé coefficients h_i in various curvilinear coordinate systems.

Coordinate system	u_1	u_2	u_3	h_1	h_2	h_3
Cartesian	x	y	z	1	1	1
Cylindrical polar	ρ	ϕ	z	1	ρ	1
Spherical polar	r	θ	ϕ	1	r	$r\sin(\theta)$

Proposition 8.1. *An element of arc can be written*

$$ds^2 = h_1^2\,du_1^2 + h_2^2\,du_2^2 + h_3^2\,du_3^2. \tag{8.7}$$

Proof. From the definition of $\mathbf{r}$ in (8.3) and (8.4), we can write

$$d\mathbf{r} = \frac{\partial \mathbf{r}}{\partial u_1}\,du_1 + \frac{\partial \mathbf{r}}{\partial u_2}\,du_2 + \frac{\partial \mathbf{r}}{\partial u_3}\,du_3 \tag{8.8}$$
$$= \mathbf{h}_1\,du_1 + \mathbf{h}_2\,du_2 + \mathbf{h}_3\,du_3 \tag{8.9}$$
$$= h_1\mathbf{e}_1\,du_1 + h_2\mathbf{e}_2\,du_2 + h_3\mathbf{e}_3\,du_3. \tag{8.10}$$

Writing $ds^2 = d\mathbf{r} \cdot d\mathbf{r}$ and substituting this expression, we obtain the result. □

Proposition 8.2. *An element of volume can be written*

$$dV = h_1 h_2 h_3 \, du_1 \, du_2 \, du_3. \tag{8.11}$$

Proof. Recall that the scalar triple product $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ between three non-coplanar vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ can be interpreted as the volume of the paralleliped spanned by the three vectors. Then, we can write

$$\begin{aligned} dV &= \mathbf{h}_1 \, du_1 \cdot (\mathbf{h}_2 \, du_2 \times \mathbf{h}_3 \, du_3) && (8.12)\\ &= h_1 \mathbf{e}_1 \, du_1 \cdot (h_2 \mathbf{e}_2 \, du_2 \times h_3 \mathbf{e}_3 \, du_3) && (8.13)\\ &= h_1 h_2 h_3 \, du_1 \, du_2 \, du_3 \, \mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3). && (8.14) \end{aligned}$$

If the basis vectors are orthonormal, $\mathbf{e}_1$ is perpendicular to both $\mathbf{e}_2$ and $\mathbf{e}_3$ and therefore parallel to their vector product. But since $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are unit vectors, $\mathbf{e}_2 \times \mathbf{e}_3$ must have unit magnitude, as does $\mathbf{e}_1$, so it must be that $\mathbf{e}_2 \times \mathbf{e}_3 = \pm \mathbf{e}_1$. Now, cross products are anti-commutative; that is, for two general vectors $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$, we have $\tilde{\mathbf{a}} \times \tilde{\mathbf{b}} = -(\tilde{\mathbf{b}} \times \tilde{\mathbf{a}})$. Then, we can choose the positive result without loss of generality by simply changing the order in which we enumerate the second and third coordinates, and so $\mathbf{e}_2 \times \mathbf{e}_3 = \mathbf{e}_1$. Then,

$$\begin{aligned} dV &= h_1 h_2 h_3 \, du_1 \, du_2 \, du_3 \, \mathbf{e}_1 \cdot \mathbf{e}_1 && (8.15)\\ &= h_1 h_2 h_3 \, du_1 \, du_2 \, du_3, && (8.16) \end{aligned}$$

completing the proof. □

8.2 Spherical polar coordinates

Consider a point in 3D space, which can be modelled in Cartesian coordinates as $\mathbf{r} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k}$ or in spherical polar coordinates are $\mathbf{r} = r\,\mathbf{e}_r$. We obtain the Cartesian coordinates from the spherical polar coordinates by

$$\begin{aligned} x &= r \sin(\theta) \cos(\phi), && (8.17)\\ y &= r \sin(\theta) \sin(\phi), && (8.18)\\ z &= r \cos(\theta). && (8.19) \end{aligned}$$

We obtain the spherical polar coordinates from the Cartesian coordinates by

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2}, && (8.20)\\ \theta &= \arccos\left(\frac{z}{r}\right), && (8.21)\\ \phi &= \operatorname{atan2}(y, x), && (8.22) \end{aligned}$$

where $\operatorname{atan2}(y, x)$ is the two-dimensional arctangent. For $x > 0$, which will be the only situations of interest to us, we have $\operatorname{atan2}(y, x) = \tan^{-1}(\frac{y}{x})$.

Proposition 8.3. *The spherical polar unit vectors are recovered from the Cartesian unit vectors by*

$$\begin{aligned} \mathbf{e}_r &= \sin(\theta) \cos(\phi)\,\mathbf{i} + \sin(\theta) \sin(\phi)\,\mathbf{j} + \cos(\theta)\,\mathbf{k}, && (8.23)\\ \mathbf{e}_\theta &= \cos(\theta) \cos(\phi)\,\mathbf{i} + \cos(\theta) \sin(\phi)\,\mathbf{j} - \sin(\theta)\,\mathbf{k}, && (8.24)\\ \mathbf{e}_\phi &= -\sin(\phi)\,\mathbf{i} + \cos(\phi)\,\mathbf{j}. && (8.25) \end{aligned}$$

Proof. In spherical polar coordinates, we see that

$$\mathbf{h}_r = \frac{\partial \mathbf{r}}{\partial r} = \frac{\partial x}{\partial r}\,\mathbf{i} + \frac{\partial y}{\partial r}\,\mathbf{j} + \frac{\partial z}{\partial r}\,\mathbf{k} = \sin(\theta) \cos(\phi)\,\mathbf{i} + \sin(\theta) \sin(\phi)\,\mathbf{j} + \cos(\theta)\,\mathbf{k}, \tag{8.26}$$

where the Cartesian unit vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are independent of position. We simply have

$$h_r = \sqrt{h_x^2 + h_y^2 + h_z^2} = \sqrt{\sin^2(\theta)\cos^2(\phi) + \sin^2(\theta)\sin^2(\phi) + \cos^2(\theta)} \tag{8.27}$$

$$= \sqrt{\sin^2(\theta)\big[\cos^2(\phi) + \sin^2(\phi)\big] + \cos^2(\theta)} = \sqrt{\sin^2(\theta) + \cos^2(\theta)} = 1, \tag{8.28}$$

where we choose the positive square root, and so using $\mathbf{e}_r = \mathbf{h}_r/h_r$ yields the quoted result. The results for $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ are obtained in the same way. □

The Cartesian unit vectors are recovered from the spherical polar unit vectors by

$$\mathbf{i} = \sin(\theta)\,\cos(\phi)\,\mathbf{e}_r + \cos(\theta)\,\cos(\phi)\,\mathbf{e}_\theta - \sin(\phi)\,\mathbf{e}_\phi, \tag{8.29}$$

$$\mathbf{j} = \sin(\theta)\,\sin(\phi)\,\mathbf{e}_r + \cos(\theta)\,\sin(\phi)\,\mathbf{e}_\theta + \cos(\phi)\,\mathbf{e}_\phi, \tag{8.30}$$

$$\mathbf{k} = \cos(\theta)\,\mathbf{e}_r - \sin(\theta)\,\mathbf{e}_\theta. \tag{8.31}$$

This can be shown by invoking a Jacobian matrix and noting that transformations back to the old coordinate system are handled by taking the inverse of this matrix, a process which is simplified for orthogonal coordinate systems, but it is beyond the scope of this project and we simply quote the well-known result.

Proposition 8.4. *Let a 3D vector* $\mathbf{A}$ *be expressed as* $A_x\,\mathbf{i} + A_y\,\mathbf{j} + A_z\,\mathbf{k}$ *in Cartesian coordinates and* $A_r\,\mathbf{e}_r + A_\theta\,\mathbf{e}_\theta + A_\phi\,\mathbf{e}_\phi$ *in spherical polar coordinates. Then, the components can be calculated by*

$$A_x = A_r\,\sin(\theta)\,\cos(\phi) + A_\theta\,\cos(\theta)\,\cos(\phi) - A_\phi\,\sin(\phi), \tag{8.32}$$

$$A_y = A_r\,\sin(\theta)\,\sin(\phi) + A_\theta\,\cos(\theta)\,\sin(\theta) + A_\phi\,\cos(\phi), \tag{8.33}$$

$$A_z = A_r\,\cos(\theta) - A_\theta\,\sin(\theta). \tag{8.34}$$

Proof. Write $\mathbf{A} = A_r\,\mathbf{e}_r + A_\theta\,\mathbf{e}_\theta + A_\phi\,\mathbf{e}_\phi$, substitute the expressions for the spherical polar unit vectors in Eqs. (8.23), (8.24) and (8.25) and equate the resulting Cartesian coefficients with $\mathbf{A} = A_x\,\mathbf{i} + A_y\,\mathbf{j} + A_z\,\mathbf{k}$. □

8.3 Axisymmetric spherical polar coordinates

An **axisymmetric vector function $\mathbf{A}(\mathbf{r})$** is one whose components do not depend on the ϕ-coordinate. That is, in spherical polar coordinates, the components are $A_i = A_i(r, \theta) \neq A_i(r, \theta, \phi)$.

In axisymmetry, we have the freedom to choose ϕ without affecting any results. Choosing $\phi = 0$, as we will for the 2D evolution in this project, the relations between the coordinates simplify to

$$x = r\,\sin(\theta), \tag{8.35}$$
$$y = 0, \tag{8.36}$$
$$z = r\,\cos(\theta), \tag{8.37}$$

and the relations between the unit vectors simplify to

$$\mathbf{i} = \sin(\theta)\,\mathbf{e}_r + \cos(\theta)\,\mathbf{e}_\theta, \tag{8.38}$$
$$\mathbf{j} = \mathbf{e}_\phi, \tag{8.39}$$
$$\mathbf{k} = \cos(\theta)\,\mathbf{e}_r - \sin(\theta)\,\mathbf{e}_\theta, \tag{8.40}$$

and

$$\mathbf{e}_r = \sin(\theta)\,\mathbf{i} + \cos(\theta)\,\mathbf{k}, \tag{8.41}$$
$$\mathbf{e}_\theta = \cos(\theta)\,\mathbf{i} - \sin(\theta)\,\mathbf{k}, \tag{8.42}$$
$$\mathbf{e}_\phi = \mathbf{j}. \tag{8.43}$$

Then, an axisymmetric vector $\mathbf{A}$ has components

$$A_x = A_r\,\sin(\theta) + A_\theta\,\cos(\theta), \tag{8.44}$$
$$A_y = A_\phi, \tag{8.45}$$
$$A_z = A_r\,\cos(\theta) - A_\theta\,\sin(\theta). \tag{8.46}$$

9 Vector functions

9.1 Vector calculus

Consider spherical polar coordinates (r, θ, ϕ). Suppose that we have a scalar function $f(r, \theta, \phi)$ and a vector function

$$\mathbf{A}(r, \theta, \phi) \;=\; A_r(r, \theta, \phi)\, \mathbf{e}_r + A_\theta(r, \theta, \phi)\, \mathbf{e}_\theta + A_\phi(r, \theta, \phi)\, \mathbf{e}_\phi. \tag{9.1}$$

The gradient $\boldsymbol{\nabla} f$ of the scalar function is

$$\boldsymbol{\nabla} f \;=\; \frac{\partial f}{\partial r}\, \mathbf{e}_r + \frac{1}{r}\, \frac{\partial f}{\partial \theta}\, \mathbf{e}_\theta + \frac{1}{r\, \sin(\theta)}\, \frac{\partial f}{\partial \phi}\, \mathbf{e}_\phi. \tag{9.2}$$

The divergence $\boldsymbol{\nabla} \cdot \mathbf{A}$ and curl $\boldsymbol{\nabla} \times \mathbf{A}$ of the vector function are

$$\boldsymbol{\nabla} \cdot \mathbf{A} \;=\; \frac{\partial A_r}{\partial r} + \frac{2}{r}\, A_r + \frac{1}{r}\, \frac{\partial A_\theta}{\partial \theta} + \frac{\cos(\theta)}{r\, \sin(\theta)}\, A_\theta + \frac{1}{r\, \sin(\theta)}\, \frac{\partial A_\phi}{\partial \phi}, \tag{9.3}$$

$$\begin{aligned} \boldsymbol{\nabla} \times \mathbf{A} \;=\; & \left[\frac{1}{r}\, \frac{\partial A_\phi}{\partial \theta} + \frac{\cos(\theta)}{r \sin(\theta)}\, A_\phi - \frac{1}{r \sin(\theta)}\, \frac{\partial A_\theta}{\partial \phi}\right] \mathbf{e}_r + \left[\frac{1}{r \sin(\theta)}\, \frac{\partial A_r}{\partial \phi} - \frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r}\right] \mathbf{e}_\theta \\ & + \left[\frac{\partial A_\theta}{\partial r} + \frac{A_\theta}{r} - \frac{1}{r}\, \frac{\partial A_r}{\partial \theta}\right] \mathbf{e}_\phi. \end{aligned} \tag{9.4}$$

In axisymmetry, these reduce to

$$\boldsymbol{\nabla} \cdot \mathbf{A} \;=\; \frac{\partial A_r}{\partial r} + \frac{2}{r}\, A_r + \frac{1}{r}\, \frac{\partial A_\theta}{\partial \theta} + \frac{\cos(\theta)}{r\, \sin(\theta)}\, A_\theta, \tag{9.5}$$

$$\boldsymbol{\nabla} \times \mathbf{A} \;=\; \left[\frac{1}{r}\, \frac{\partial A_\phi}{\partial \theta} + \frac{\cos(\theta)}{r \sin(\theta)}\, A_\phi\right] \mathbf{e}_r + \left[-\frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r}\right] \mathbf{e}_\theta + \left[\frac{\partial A_\theta}{\partial r} + \frac{A_\theta}{r} - \frac{1}{r}\, \frac{\partial A_r}{\partial \theta}\right] \mathbf{e}_\phi. \tag{9.6}$$

For all scalar functions $f(\mathbf{r})$ and vector functions $\mathbf{A}(\mathbf{r})$,

$$\boldsymbol{\nabla} \cdot (f\, \mathbf{A}) \;=\; f\, \boldsymbol{\nabla} \cdot \mathbf{A} + \mathbf{A} \cdot (\boldsymbol{\nabla} f), \tag{9.7}$$

$$\boldsymbol{\nabla} \times (f\, \mathbf{A}) \;=\; f\, \boldsymbol{\nabla} \times \mathbf{A} + \mathbf{A} \times (\boldsymbol{\nabla} f). \tag{9.8}$$

In particular, for purely radial functions $f(r)$,

$$\boldsymbol{\nabla} f \;=\; \frac{df}{dr}\, \mathbf{e}_r, \tag{9.9}$$

$$\boldsymbol{\nabla} \cdot \big[f\, \mathbf{A}\big] \;=\; f\, \boldsymbol{\nabla} \cdot \mathbf{A} + \frac{df}{dr}\, \mathbf{A}_r, \tag{9.10}$$

$$\boldsymbol{\nabla} \times (f\, \mathbf{A}) \;=\; f\, \boldsymbol{\nabla} \times \mathbf{A} + \frac{df}{dr}\Big(- A_\phi\, \mathbf{e}_\theta + A_\theta\, \mathbf{e}_\phi\Big). \tag{9.11}$$

.

9.2 Poloidal-toroidal decomposition

A **divergenceless** vector field, also known as a **solenoidal** vector field, is one whose divergence is zero:

$$\boldsymbol{\nabla}\cdot\mathbf{F} \;=\; 0. \tag{9.12}$$

If this is the case, the vector only has $n-1$ degrees of freedom, where n is the number of spatial dimensions. For a 3D vector, we can express these degrees of freedom as a **poloidal field P** and a **toroidal field T**, which are independent of each other:

$$\mathbf{F} \;=\; \mathbf{P}+\mathbf{T}. \tag{9.13}$$

The expression of a divergenceless vector in this form is known as its **poloidal-toroidal decomposition**, or **Mie decomposition**; it is a special case of the **Helmholtz decomposition**.

A poloidal field is one whose curl is orthogonal to the radial vector $\mathbf{r}$ in spherical coordinates, and a toroidal field $\mathbf{T}$ is one which is itself orthogonal to $\mathbf{r}$ (Figure 9.1):

$$\mathbf{r}\cdot\mathbf{T} \;=\; 0, \qquad \mathbf{r}\cdot(\boldsymbol{\nabla}\times\mathbf{P}) \;=\; 0. \tag{9.14}$$

They may be expressed as

$$\mathbf{P} \;=\; \boldsymbol{\nabla}\times(\mathbf{r}\times\boldsymbol{\nabla}P) \;=\; \boldsymbol{\nabla}\times\big[\boldsymbol{\nabla}\times(\mathbf{r}P)\big], \tag{9.15}$$

$$\mathbf{T} \;=\; \mathbf{r}\times\boldsymbol{\nabla}T \;=\; \boldsymbol{\nabla}\times(\mathbf{r}T), \tag{9.16}$$

where $\mathbf{r}$ is a given vector[1] and $P(r,\theta,\phi)$ and $T(\theta,\phi)$ are called the **poloidal and toroidal scalars** for $\mathbf{F}$ (Backus, 1986). The toroidal scalar has no radial dependence. The functions $P, T, \mathbf{P}, \mathbf{T}$ are unique up to the addition of an arbitrary function of r only. The curl of a poloidal field is azimuthal and the curl of an azimuthal field is poloidal. Usually the poloidal and toroidal components are specified first and the field is calculated from that.

Proposition 9.1. *Given P, T, we have in spherical coordinates that*

$$\mathbf{P} \;=\; -\frac{1}{r}\left[\frac{\partial^2 P}{\partial\theta^2}+\frac{\cos(\theta)}{\sin(\theta)}\frac{\partial P}{\partial\theta}+\frac{1}{\sin^2(\theta)}\frac{\partial^2 P}{\partial\phi^2}\right]\mathbf{e}_r+\left[\frac{\partial^2 P}{\partial r\partial\theta}+\frac{1}{r}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_\theta+\left[\frac{1}{\sin(\theta)}\frac{\partial^2 P}{\partial r\partial\phi}+\frac{1}{r\sin(\theta)}\frac{\partial P}{\partial\phi}\right]\mathbf{e}_\phi, \tag{9.17}$$

$$\mathbf{T} \;=\; \frac{1}{\sin(\theta)}\frac{\partial T}{\partial\phi}\,\mathbf{e}_\theta-\frac{\partial T}{\partial\theta}\,\mathbf{e}_\phi, \tag{9.18}$$

so the vector is

$$\begin{aligned}\mathbf{F} \;=\; &-\frac{1}{r}\left[\frac{\partial^2 P}{\partial\theta^2}+\frac{\cos(\theta)}{\sin(\theta)}\frac{\partial P}{\partial\theta}+\frac{1}{\sin^2(\theta)}\frac{\partial^2 P}{\partial\phi^2}\right]\mathbf{e}_r+\left[\frac{1}{\sin(\theta)}\frac{\partial T}{\partial\phi}+\frac{\partial^2 P}{\partial r\partial\theta}+\frac{1}{r}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_\theta\\ &+\left[\frac{1}{\sin(\theta)}\frac{\partial^2 P}{\partial r\partial\phi}+\frac{1}{r\sin(\theta)}\frac{\partial P}{\partial\phi}-\frac{\partial T}{\partial\theta}\right]\mathbf{e}_\phi.\end{aligned} \tag{9.19}$$

Proof. Given two scalar functions $P(r,\theta,\phi)$ and $T(\theta,\phi)$, and taking $\mathbf{r}=r\,\mathbf{e}_r$, we have

$$\boldsymbol{\nabla}\times(P\mathbf{r}) \;=\; \frac{1}{\sin(\theta)}\frac{\partial P}{\partial\phi}\,\mathbf{e}_\theta-\frac{\partial P}{\partial\theta}\,\mathbf{e}_\phi, \tag{9.20}$$

$$\Rightarrow\;\boldsymbol{\nabla}\times(P\mathbf{r})+T\mathbf{r} \;=\; Tr\,\mathbf{e}_r+\frac{1}{\sin(\theta)}\frac{\partial P}{\partial\phi}\,\mathbf{e}_\theta-\frac{\partial P}{\partial\theta}\,\mathbf{e}_\phi. \tag{9.21}$$

The original vector $\mathbf{F}$ is given by the curl of this expression, which returns (9.19). By taking the divergence of this result, it can be confirmed after some algebra that $\boldsymbol{\nabla}\cdot\mathbf{F}=0$ as required for P, T to exist. When this is done, notice that no derivatives of T wrt r are taken and the equality is still true in general, reflecting the fact that only the poloidal component carries a radial dependence. □

[1] The natural choice is the radial vector $\mathbf{r}=r\,\mathbf{e}_r$ in spherical coordinates.

Notice that the radial component is only determined by the poloidal component, not the toloidal component. It is also possible given F_r, F_θ, F_ϕ to find P, T; this is done by solving equations analogous to Laplace's equation, but is beyond the scope of this text.

Proposition 9.2. *If the vector field is* **axisymmetric**, *that is, its spherical polar components are ϕ-independent $F_i = F_i(r,\theta) \neq F_i(r,\theta,\phi)$, the poloidal and toroidal scalars are also ϕ-independent and the toroidal field is equal to the azimuthal field. The latter implies that the poloidal field is the sum of the radial and polar fields:*

$$\mathbf{T} = F_\phi\,\mathbf{e}_\phi = -\frac{\partial T}{\partial\theta}\,\mathbf{e}_\phi, \tag{9.22}$$

$$\mathbf{P} = F_r\,\mathbf{e}_r + F_\theta\,\mathbf{e}_\theta = -\frac{1}{r}\left[\frac{\partial^2 P}{\partial\theta^2} + \frac{\cos(\theta)}{\sin(\theta)}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_r + \left[\frac{\partial^2 P}{\partial r\partial\theta} + \frac{1}{r}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_\theta. \tag{9.23}$$

Proof. We do not attempt to prove that P, T are ϕ-independent, but the reverse (if P, T are ϕ-independent then $\mathbf{F}$ must be axisymmetric) follows immediately from the fact that we expressed $\mathbf{F}$ solely in terms of P, T in (9.19). If $\mathbf{F}$ is axisymmetric, the toroidal and poloidal components are The expression for $\mathbf{F}$ becomes

$$\mathbf{F} = -\frac{1}{r}\left[\frac{\partial^2 P}{\partial\theta^2} + \frac{\cos(\theta)}{\sin(\theta)}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_r + \left[\frac{\partial^2 P}{\partial r\partial\theta} + \frac{1}{r}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_\theta - \frac{\partial T}{\partial\theta}\,\mathbf{e}_\phi, \tag{9.24}$$

and we see that P only appears in F_r, F_θ and T only appears in F_ϕ. Further, from (9.17) and (9.18) the vectors $\mathbf{P}, \mathbf{T}$ are exactly equal to these components:

$$\mathbf{P} = -\frac{1}{r}\left[\frac{\partial^2 P}{\partial\theta^2} + \frac{\cos(\theta)}{\sin(\theta)}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_r + \left[\frac{\partial^2 P}{\partial r\partial\theta} + \frac{1}{r}\frac{\partial P}{\partial\theta}\right]\mathbf{e}_\theta + 0\,\mathbf{e}_\phi = F_r\,\mathbf{e}_r + F_\theta\,\mathbf{e}_\theta, \tag{9.25}$$

$$\mathbf{T} = 0\,\mathbf{e}_\theta - \frac{\partial T}{\partial\theta}\,\mathbf{e}_\phi = F_\phi\,\mathbf{e}_\phi. \tag{9.26}$$

□

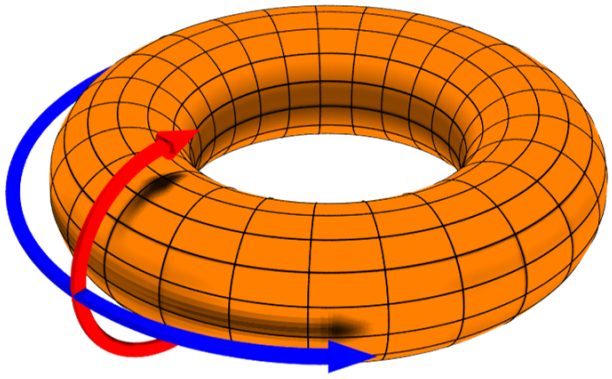

Figure 9.1: Vectors in the poloidal (red) and toroidal (blue) directions (Burke, 2006, used under CC BY 2.5).

9.3 Force-free magnetic fields

Force-free magnetic fields are those in which the magnetic flux density $\mathbf{B}$ is the only relevant force acting on particles within them. We will discuss force-free electrodynamics fully in Chapter 20; for now, let us lay out some of their mathematical properties.

Proposition 9.3. *Force-free magnetic fields satisfy*

$$(\boldsymbol{\nabla} \times \mathbf{B}) \times \mathbf{B} \;=\; \mathbf{0}, \tag{9.27}$$

or equivalently

$$\boldsymbol{\nabla} \times \mathbf{B} \;=\; \alpha(\mathbf{r})\, \mathbf{B}. \tag{9.28}$$

They also satisfy

$$(\nabla^2 + \alpha^2)\, \mathbf{B} \;=\; \mathbf{0}. \tag{9.29}$$

Here, the **force-free parameter** $\alpha(\mathbf{r})$ *may be a function of position (Davidson, 2001, §2.3).*

Proof.

- If the electric field is negligible compared to the magnetic field, we can approximate that $\boldsymbol{\nabla} \times \mathbf{B} = \mu_0\, \mathbf{J}$ and approximate the Lorentz force as

$$\boldsymbol{\mathcal{L}} \;=\; \mathbf{J} \times \mathbf{B} \;=\; \frac{1}{\mu_0} (\boldsymbol{\nabla} \times \mathbf{B}) \times \mathbf{B}. \tag{9.30}$$

  Setting $\boldsymbol{\mathcal{L}} = \mathbf{0}$, we arrive at $(\boldsymbol{\nabla} \times \mathbf{B}) \times \mathbf{B} = \mathbf{0}$.

- If two vectors $\mathbf{B}$ and $\mathbf{C}$ are parallel $\mathbf{B} \parallel \mathbf{C}$, then we have $\mathbf{B} \times \mathbf{C} = \mathbf{0}$ and we can write one vector as a multiple of the other, $\mathbf{C} = \alpha(\mathbf{r})\mathbf{B}$ or $\mathbf{B} = \mathbf{C}/\alpha(\mathbf{r})$, where $\alpha(\mathbf{r})$ is a scalar that can in general depend on the position vector $\mathbf{r}$. It follows that the magnetic field satisfies $\boldsymbol{\nabla} \times \mathbf{B} = \alpha(\mathbf{r})\, \mathbf{B}$.

- Using the definition of the vector Laplacian, we have

$$\nabla^2 \mathbf{B} \;=\; \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{B}) - \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{B}) \;=\; 0 - \boldsymbol{\nabla} \times (\alpha \mathbf{B}) \;=\; -\alpha \boldsymbol{\nabla} \times \mathbf{B} \;=\; -\alpha^2 \mathbf{B}. \tag{9.31}$$

□

In mathematics, a **Beltrami field** is any general field $\mathbf{F}$ satisfying $\boldsymbol{\nabla} \times \mathbf{F} = a\, \mathbf{F}$ for constant a. In this case, we have a **linear force-free magnetic field** and the equation can be solved exactly, resulting in an infinite series involving **Gegenbauer polynomials** (Chandrasekhar, 1956). Various other solution methods are listed in §2 of Wiegelmann & Sakurai (2021). The special case $\alpha = 0$ corresponds to

$$\boldsymbol{\nabla} \times \mathbf{B} \;=\; \mathbf{0}. \tag{9.32}$$

Any curl-free vector can be expressed in terms of a scalar potential, so magnetic fields satisfying this expression are known as **potential magnetic fields**. Using Eq. (20.4) and neglecting $\mathbf{E}$ compared to $\mathbf{B}$, it occurs when there is no current $\mathbf{J} = \mathbf{0}$. As such, the term **current-free magnetic field** may also be used.

Force-free fields are a useful approximation when magnetic fields dominate, as is the case in a neutron star magnetosphere or the solar magnetosphere, but no truly force-free fields exist.

Proposition 9.4. *There are no force-free fields other than* $\mathbf{B} = \mathbf{0}$ *for which* $\mathbf{J}$ *is localised in space and* $\mathbf{B}$ *is differentiable everywhere and* $\mathcal{O}(x^{-3})$ *at infinity.*

Proof. The Biot-Savart law for the electrostatic case $\mathbf{E} \neq \mathbf{E}(t)$ becomes

$$\mathbf{B} \;=\; \frac{\mu_0}{4\pi}\int \frac{\mathbf{J}\times(\mathbf{x}-\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^3}\,d^3\mathbf{x}' \;=\; \frac{1}{4\pi}\int \frac{(\boldsymbol{\nabla}\times\mathbf{B})\times(\mathbf{x}-\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^3}\,d^3\mathbf{x}' \;=\; \frac{\alpha}{4\pi}\int \frac{\mathbf{B}\times(\mathbf{x}-\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|^3}\,d^3\mathbf{x}', \tag{9.33}$$

but since $(\mathbf{B}\times\mathbf{r}) \perp \mathbf{B} \perp \mathbf{r}$, this implies $\mathbf{B} \perp \mathbf{B}$. Such a condition is only satisfied if $\mathbf{B} = \mathbf{0}$. □

Proposition 9.5. *In a stationary fluid, a force-free field decays as*

$$\mathbf{B}(t) = \mathbf{B}_0\, e^{-\lambda\alpha^2 t}. \tag{9.34}$$

Proof. In a stationary fluid, we have $\mathbf{u} = \mathbf{0}$ and the induction equation becomes $\partial_t\mathbf{B} = \mathbf{0} + \lambda\nabla^2\mathbf{B} = -\lambda\alpha^2\mathbf{B}$, so that $\mathbf{B}(t) \;=\; \mathbf{B}_0\, e^{-\lambda\alpha^2 t}$. □

Proposition 9.6. *The poloidal and toroidal fields corresponding to a force-free field* $\mathbf{F}$ *can be written*

$$\mathbf{P} \;=\; \frac{(\boldsymbol{\nabla}P)\times\mathbf{e}_\phi}{r\,\sin(\theta)}, \qquad \mathbf{T} \;=\; \frac{1}{r\sin(\theta)}\int \alpha\, dP. \tag{9.35}$$

In particular, we have

$$F_r \;=\; -\frac{1}{r^2}\frac{\partial P}{\partial \mu} \;=\; \frac{1}{r^2\sin(\theta)}\frac{\partial P}{\partial \theta}, \qquad F_\theta \;=\; -\frac{1}{r\sin(\theta)}\frac{\partial P}{\partial r}. \tag{9.36}$$

This is the most general expression for an axisymmetric divergenceless field.

Proof. Consider spherical coordinates and axisymmetry. For the poloidal component, the gradient of the poloidal scalar $P(r,\theta)$ is

$$\boldsymbol{\nabla}P \;=\; \frac{\partial P}{\partial r}\,\mathbf{e}_r + \frac{1}{r}\frac{\partial P}{\partial \theta}\,\mathbf{e}_\theta, \tag{9.37}$$

so we have

$$(\boldsymbol{\nabla}P)\times\mathbf{e}_\phi \;=\; \frac{\partial P}{\partial r}\,\mathbf{e}_r\times\mathbf{e}_\phi + \frac{1}{r}\frac{\partial P}{\partial \theta}\,\mathbf{e}_\theta\times\mathbf{e}_\phi \;=\; \frac{1}{r}\frac{\partial P}{\partial \theta}\,\mathbf{e}_r - \frac{\partial P}{\partial r}\,\mathbf{e}_\theta, \tag{9.38}$$

$$\Rightarrow\; \frac{(\boldsymbol{\nabla}P)\times\mathbf{e}_\phi}{r\sin(\theta)} \;=\; \frac{1}{r^2\sin(\theta)}\frac{\partial P}{\partial \theta}\,\mathbf{e}_r - \frac{1}{r\sin(\theta)}\frac{\partial P}{\partial r}\,\mathbf{e}_\theta. \tag{9.39}$$

□

9.4 Grad-Shafranov equation

The **Grad-Shafranov equation (GSE)** is used to describe the interiors of neutron stars. While not used within this project, our results are dependent on the configuration of the neutron star surface and so it will be useful in future applications for modelling relationships between the magnetosphere, surface and interior. In this section, we briefly describe the Grad-Shafranov equation and its implementation across the literature. Let us adopt cylindrical coordinates (ϖ, ϕ, z) to match the majority of the literature.

9.4.1 Flux functions in axisymmetric systems

The solenoidal condition $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$, Eq. (20.2), is not enforced by the equations of motion, so a nonzero magnetic field divergence may arise as we evolve the electric and magnetic fields with time. This is especially true if our code is prone to numerical errors. Nonzero magnetic field divergences often manifest as energy leakages because energy propagating along magnetic field lines to large radii might not return to the system.

In many codes evolving the electromagnetic fields, the solenoidal condition is checked at each timestep in order to prevent large dissipation from occurring.[2] However, if the configuration is axisymmetric (no ϕ-dependence), we can *enforce* Eq. (20.2) by introducing a **flux function**[3] $u(\mathbf{r})$, defined such that

$$\mathbf{B} = -\frac{1}{\varpi}\frac{\partial u}{\partial z}\mathbf{e}_\varpi + B_\phi \mathbf{e}_\phi + \frac{1}{\varpi}\frac{\partial u}{\partial \varpi}\mathbf{e}_z = \frac{1}{\varpi}(\boldsymbol{\nabla} u) \times \mathbf{e}_\phi + B_\phi \mathbf{e}_\phi. \tag{9.40}$$

Proposition 9.7. *The expression (9.40) satisfies (20.2) for any arbitrary function u, and implies that u is constant along field lines.*

Proof.

- **Flux function enforces incompressibility:** Take the divergence of (9.40) in cylindrical polar coordinates:

$$\boldsymbol{\nabla} \cdot \mathbf{B} = \frac{1}{\varpi}\frac{\partial(\varpi B_\varpi)}{\partial \varpi} + \frac{1}{\varpi}\frac{\partial B_\phi}{\partial \phi} + \frac{\partial B_z}{\partial z} = -\frac{1}{\varpi}\frac{\partial^2 u}{\partial \varpi \partial z} + 0 + \frac{1}{\varpi}\frac{\partial^2 u}{\partial z \partial \varpi} = 0. \tag{9.41}$$

- **Relation between flux function and vector potential:** The definition of the curl in cylindrical polar coordinates for an axisymmetric system is

$$\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A} = \left[\frac{1}{\varpi}\frac{\partial A_z}{\partial \phi} - \frac{\partial A_\phi}{\partial z}\right]\mathbf{e}_\varpi + \left[\frac{\partial A_\varpi}{\partial z} - \frac{\partial A_z}{\partial \varpi}\right]\mathbf{e}_\phi + \frac{1}{\varpi}\left[\frac{\partial(\varpi A_\phi)}{\partial \varpi} - \frac{\partial A_\varpi}{\partial \phi}\right]\mathbf{e}_z \tag{9.42}$$

$$= -\frac{\partial A_\phi}{\partial z}\mathbf{e}_\varpi + \left[\frac{\partial A_\varpi}{\partial z} - \frac{\partial A_z}{\partial \varpi}\right]\mathbf{e}_\phi + \frac{1}{\varpi}\frac{\partial(\varpi A_\phi)}{\partial \varpi}\mathbf{e}_z. \tag{9.43}$$

  Comparing to our definition of $\mathbf{B}$ by a flux function (9.40), we see that this is satisfied if $A_\phi = u/\varpi$ and the $\mathbf{e}_\phi$ term remains unevaluated $\partial A_\varpi/\partial_z - \partial A_z/\partial A_\varpi = B_\phi$. Further, note that

$$(\boldsymbol{\nabla} u) \times \mathbf{e}_\phi = \left(\frac{\partial u}{\partial \varpi}\mathbf{e}_\varpi + \frac{1}{\varpi}\frac{\partial u}{\partial \phi}\mathbf{e}_\phi + \frac{\partial u}{\partial z}\mathbf{e}_z\right) \times \mathbf{e}_\phi = \frac{\partial u}{\partial \varpi}\mathbf{e}_z + \mathbf{0} - \frac{\partial u}{\partial z}\mathbf{e}_\varpi. \tag{9.44}$$

- **Flux function is constant along magnetic field lines:** Note that $(\boldsymbol{\nabla} u) \times \mathbf{e}_\phi$ is perpendicular to both $\boldsymbol{\nabla} u$ and $\mathbf{e}_\phi$. In other words, $\boldsymbol{\nabla} u \perp \mathbf{B}$, i.e. $(\boldsymbol{\nabla} u) \cdot \mathbf{B} = 0$.

□

[2] Alternatively, energy conservation may be checked as a proxy.
[3] Sometimes called a **streamfunction** in analogy to fluid flows.

The magnetic field may also be written in terms of a vector potential $\mathbf{A}$ such that $\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A}$; the flux function and vector potential are related by

$$u \;=\; \varpi\, A_\phi. \tag{9.45}$$

We can then choose to evolve the flux functions instead of the fields or their potentials, and guarantee that the magnetic field remains solenoidal.

An axisymmetric magnetic field $\mathbf{B}$ and current density $\mathbf{j}$ can be written in the general form (Lander & Jones, 2009, §A.1)

$$\mathbf{B} \;=\; \frac{1}{\varpi}\,(\boldsymbol{\nabla} u) \times \mathbf{e}_\phi + B_\phi\, \mathbf{e}_\phi, \tag{9.46}$$

$$\mathbf{j} \;=\; \frac{1}{4\pi\varpi}\,\boldsymbol{\nabla}(\varpi B_\phi) \times \mathbf{e}_\phi - \frac{1}{4\pi\varpi}\,(\Delta_* u)\,\mathbf{e}_\phi. \tag{9.47}$$

9.4.2 Formulations

The GSE relates the magnetic field and current of a system in axisymmetric perfect MHD with mixed poloidal and toroidal fields (Grad & Rubin, 1958; Shafranov, 1966). It has been formulated in many different ways in the literature; we list them here.

- Lander & Jones (2009), Eq. (12):

$$4\pi\rho\,\frac{dM}{du} \;=\; -\frac{1}{\varpi^2}\left[\Delta_* u + f(u)\,\frac{df}{du}\right], \tag{9.48}$$

  where ρ is the density, u is a flux function defined by (9.40), M, f are arbitrary functions of u to be specified by the exact situation at hand,[4] and for brevity the differential operator Δ_* was defined as

$$\Delta_* \;\equiv\; \frac{\partial^2}{\partial\varpi^2} - \frac{1}{\varpi}\,\frac{\partial}{\partial\varpi} + \frac{\partial^2}{\partial z^2}. \tag{9.49}$$

  Let us take this as the main formulation for our paper, so that we can compare the others.

- Grad & Rubin (1958), Eq. (18):

$$\Delta^*\psi + \mu r^2 p'(\psi) + \mu f'(\psi) \;=\; 0, \tag{9.50}$$

  where (r, θ, z) is the coordinate system, ψ is the flux function,[5] $\Delta^* \equiv \Delta_*$, μ is the vacuum permeability and p, f are arbitrary functions of ψ. In our notation, this is

$$\Delta_* u + \mu_0 \varpi^2\,\frac{dp}{du} + \mu_0\,\frac{df}{du} \;=\; 0. \tag{9.51}$$

- Haverkort (2009a,b), Eq. (13) in his first paper:

$$R^2\,\boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\psi}{R^2}\right) \;=\; -\mu_0 R^2\,\frac{\partial p}{\partial\psi} - F\,\frac{dF}{d\psi}, \tag{9.52}$$

  where (R, ϕ, Z) is the coordinate system, ψ is the flux function, p is the magnetic pressure and F is an arbitrary function of ψ. In our notation, this is

$$\Delta_* u + \mu_0 \varpi^2\,\frac{\partial p}{\partial\psi} + F\,\frac{dF}{d\psi} \;=\; 0. \tag{9.53}$$

[4] In fact M is defined in terms of the Lorentz force, itself a cross product between $\mathbf{j}$ and $\mathbf{B}$, but it can be treated as an arbitrary function until specified (Lander & Jones, 2009, §2.1,§A.1).

[5] But defined under a different sign convention to (9.40): $B_r = (1/r)\,\partial_z\psi$ and $B_z = -(1/r)\,\partial_r\psi$.

- Thompson et al. (2002), Eq. (6) and Turolla et al. (2015), Eq. (4):

$$(1-\mu^2)f'' + p(p+1)f + Cf^{1+2/p} = 0, \tag{9.54}$$

where $f = f(\mu) = f[\cos(\theta)]$ is the subject of the equation, $p \in [0,1]$ is the **radial index**[6] and C is an arbitrary constant. This version is only valid in axisymmetric spherical coordinates and in the force-free approximation. Note that Thompson et al. do not explicitly name this as the GSE but Turolla et al. do.

Proposition 9.8. *The LHS of (9.53) is equivalent to* Δ_*.

Proof. We have

$$\frac{\boldsymbol{\nabla} u}{\varpi^2} = \frac{1}{\varpi^2}\left(\frac{\partial u}{\partial \varpi}\mathbf{e}_\varpi + \frac{1}{\varpi}\frac{\partial u}{\partial \phi}\mathbf{e}_\phi + \frac{\partial u}{\partial z}\mathbf{e}_z\right) = \frac{1}{\varpi^2}\frac{\partial u}{\partial \varpi}\mathbf{e}_\varpi + \frac{1}{\varpi^2}\frac{\partial u}{\partial z}\mathbf{e}_z, \tag{9.55}$$

$$\begin{aligned}\Rightarrow\ \varpi^2 \boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla} u}{\varpi^2}\right) &= \varpi^2\left(\frac{1}{\varpi}\frac{\partial}{\partial \varpi}\left[\frac{1}{\varpi}\frac{\partial u}{\partial \varpi}\right] + 0 + \frac{1}{\varpi^2}\frac{\partial^2 u}{\partial z^2}\right) = \varpi\left[-\frac{1}{\varpi^2}\frac{\partial u}{\partial \varpi} + \frac{1}{\varpi}\frac{\partial^2 u}{\partial \varpi^2}\right] + \frac{\partial^2 u}{\partial z^2} \\ &= \frac{\partial^2 u}{\partial \varpi^2} - \frac{1}{\varpi}\frac{\partial u}{\partial \varpi} + \frac{\partial^2 u}{\partial z^2} = \Delta_*,\end{aligned} \tag{9.56}$$

where the flux function is not a function of ϕ and so $\partial_\phi u = 0$. □

Low & Lou (1990) solved the GSE in axisymmetric spherical polar coordinates, (r,θ,ϕ) with ϕ-invariance, yielding an ordinary differential equation that can be solved numerically. The numerical solution of the resulting ODE was further discussed by Wolfson (1995).

[6] This describes how the magnetic field falls off with radial distance, but also, as Thompson et al. (2002) found, the amount of shear in the field.

10 Numerical integration

10.1 Fourth-order Runge-Kutta (RK4) method

Perhaps the most popular one-dimensional numerical integration technique is the **fourth-order Runge-Kutta method** (hereafter **RK4 method**). If the functional form of a quantity depending on x and y is known to be $f(x,y) = \frac{dy}{dx}$, the integral from some coordinate x_n with known function value $y_n = y(x_n)$ to some other coordinate $x_{n+1} = x_n + h$ with known function value $y_{n+1} = y(x_n + h)$ is well approximated by

$$y_{n+1} \;=\; \int_{x_n}^{x_n+h} f(x,y)\,dx \;\approx\; \frac{1}{6}\big(k_1 + 2k_2 + 2k_3 + k_4\big), \tag{10.1}$$

where

$$k_1 \;=\; h\,f\!\left(\,x_n\,,\,y_n\,\right), \tag{10.2}$$

$$k_2 \;=\; h\,f\!\left(\,x_n + \frac{h}{2}\,,\,y_n + \frac{k_1}{2}\,\right), \tag{10.3}$$

$$k_3 \;=\; h\,f\!\left(\,x_n + \frac{h}{2}\,,\,y_n + \frac{k_2}{2}\,\right), \tag{10.4}$$

$$k_4 \;=\; h\,f\!\left(\,x_n + h\,,\,y_n + k_3\,\right). \tag{10.5}$$

If there is no y-dependence, this reduces to

$$y_{n+1} \;=\; \frac{h}{6}\left[f(x_n) + 4f\!\left(x_n + \frac{h}{2}\right) + f(x_n + h)\right]. \tag{10.6}$$

The RK4 method converges relatively quickly, is relatively computationally efficient and offers a global truncation error (i.e. after a large number of iterations) $\mathcal{O}(h^4)$. This makes it one of the most popular one-dimensional numerical integration techniques (Hoffman, 1992, §7.5, §7.9). However, it does not automatically provide an explicit error estimate; instead, one must compute the integral again with two half-steps $\frac{h}{2}$, yielding a single-step error estimate given in Proposition 10.1. This requires three times as many derivative function evaluations per step, so can dramatically reduce computational efficiency. If a more efficient error-control method exists for the intermediate steps,[1] one may choose to calculate this error only at the final step in order to yield a quantitative estimate.

Proposition 10.1. *An estimate of the single-step numerical error from the RK4 method is*

$$(\text{Error}) \;=\; \frac{16}{15}\left[y_{n+1}\!\left(\frac{h}{2}\right) - y_{n+1}(h)\right]. \tag{10.7}$$

Proof. Suppose we calculate y_{n+1} with an nth-order finite difference equation that uses a single step h. Then, the single-step error scales as h^{n+1} (multiplied by some unknown constant coefficient C). The exact solution $\overline{y}_{n+1}$ and the calculated value $y_{n+1}(h)$ are related by[2]

$$\overline{y}_{n+1} \;=\; y_{n+1}(h) + Ch^{n+1} + \mathcal{O}(n+2). \tag{10.8}$$

[1] For example, a calculation to determine whether some conserved quantity such as energy does not change appreciably given y_{n+1}; see §22.3.

[2] Here, the notation $y_{n+1}(h)$ means the value of y_{n+1} when calculated with step size h.

Our goal is to calculate the error term Ch^{n+1}. We do this by repeating the calculation with two half-steps and hence two errors each scaling as $(h/2)^{n+1}$ that are summed together:

$$\overline{y}_{n+1} \;=\; y_{n+1}\left(\frac{h}{2}\right) + 2C\left(\frac{h}{2}\right)^{n+1}. \tag{10.9}$$

We have two simultaneous equations, which we can solve for the error term for the error term Ch^{n+1}. Subtract (10.9) from (10.8) and rearrange for the error term:

$$0 \;=\; y_{n+1}(h) - y_{n+1}\left(\frac{h}{2}\right) + Ch^{n+1} - 2C\left(\frac{h}{2}\right)^{n+1} \;=\; y_{n+1}(h) - y_{n+1}\left(\frac{h}{2}\right) + Ch^{n+1}\left(1-2^{-n}\right), \tag{10.10}$$

$$\Rightarrow\; Ch^{n+1} \;=\; \left(1-2^{-n}\right)^{-1}\left[y_{n+1}\left(\frac{h}{2}\right) - y_{n+1}(h)\right] \;=\; \frac{2^n}{2^n-1}\left[y_{n+1}\left(\frac{h}{2}\right) - y_{n+1}(h)\right]. \tag{10.11}$$

Finally, the single-step truncation error for the RK4 method is h^4, so we have $n = 4$ and the prefactor becomes $16/15$. □

Proposition 10.2. *The RK4 method is exact for polynomials up to third-order. Since the RK4 error estimate is taken by half-steps, the error will be identically zero if the algorithm is exact.*

Proof. For the function $f(x) = x^n$, the exact integral between $x = x_0$ and $x = h$ is

$$I_{\text{exact}} \;=\; \int_{x_0}^{x_0+h} x^n\,dx \;=\; \frac{x^{n+1}}{n+1}\bigg|_{x_0}^{x_0+h} \;=\; \frac{(x_0+h)^{n+1} - x_0^{n+1}}{n+1} \;=\; \frac{1}{n+1}\sum_{k=0}^{n+1}\binom{n+1}{k} x_0^k\, h^{n+1-k} - \frac{x_0^{n+1}}{n+1}. \tag{10.12}$$

The final term $n+1$ in the summation will cancel with $-x_0^{n+1}/(n+1)$. Let us separate this from the sum, along with the penultimate term n (which will later lead to simplification),

$$\frac{1}{n+1}\binom{n+1}{n+1} x_0^{n+1}\, h^{n+1-(n+1)} + \frac{1}{n+1}\binom{n+1}{n} x_0^n\, h^{n+1-n} \;=\; \frac{x_0^{n+1}}{n+1} + x_0^n\, h, \tag{10.13}$$

and alter the binomial coefficient,

$$\binom{n+1}{k} \;=\; \frac{(n+1)\,n!}{k!\,(n+1-k)\,(n-k)!} \;=\; \frac{n+1}{n+1-k}\binom{n}{k}, \tag{10.14}$$

giving

$$I_{\text{exact}} \;=\; h\,x_0^n + \sum_{k=0}^{n-1}\binom{n}{k} x_0^k\, h^{n+1-k}\,\frac{1}{n+1-k}. \tag{10.15}$$

For the RK4 integral, the k-coefficients are

$$k_1 \;=\; h\,x_0^n, \tag{10.16}$$

$$k_2 \;=\; k_3 \;=\; h\left(x_0 + \frac{h}{2}\right)^n \;=\; h\sum_{k=0}^{n}\binom{n}{k} x_0^k \left(\frac{h}{2}\right)^{n-k} \;=\; h\sum_{k=0}^{n}\binom{n}{k} x_0^k\, h^{n-k}\, 2^{k-n}, \tag{10.17}$$

$$k_4 \;=\; h(x_0+h)^n \;=\; h\sum_{k=0}^{n}\binom{n}{k} x_0^k\, h^{n-k}, \tag{10.18}$$

giving

$$I_{\rm RK4} = \frac{h}{6}\left[x_0^n + 4\Big(x_0+\frac{h}{2}\Big)^n + (x_0+h)^n\right] = \frac{h_0}{6}\left[x_0^n + 4\sum_{k=0}^{n}\binom{n}{k}x_0^k\left(\frac{h}{2}\right)^{n-k} + \sum_{k=0}^{n}\binom{n}{k}x_0^k\,h^{n-k}\right] \quad (10.19)$$

$$= \frac{h}{6}\left[\sum_{k=0}^{n}\binom{n}{k}x_0^k\,h^{n-k}\left(1+2^{k-n+2}\right) + x_0^n\right] = h\,x_0^n + \frac{1}{6}\sum_{k=0}^{n-1}\binom{n}{k}x_0^k\,h^{n+1-k}\big(1+2^{k-n+2}\big), \quad (10.20)$$

where in the last equality we separated the final term $\binom{n}{n}x_0^n\,h^{n-n}\big(1+2^{n-n+2}\big) = 5\,x_0^n$. The error of the RK4 method is then

$$(\text{error}) = I_{\rm exact} - I_{\rm RK4} = \sum_{k=0}^{n-1}\binom{n}{k}x_0^k\,h^{n+1-k}\left[\frac{1}{n+1-k} - \frac{1}{6}\left(1+2^{k-n+2}\right)\right] \quad (10.21)$$

$$= \frac{1}{6}\sum_{k=0}^{n-1}\binom{n}{k}x_0^k\,h^{n+1-k}\,\frac{6-\big(n+1-k\big)\big(1+2^{k-n+2}\big)}{n+1-k}. \quad (10.22)$$

The numerator in the fraction above is zero only if $k \in \{\,n-1\,,\,n-2\,,\,n-3\,\}$:

$$6-\big(n+1-[n-1]\big)\big(1+2^{(n-1)-n+2}\big) = 6-2\big(1+2^1\big) = 0, \quad (10.23)$$

$$6-\big(n+1-[n-2]\big)\big(1+2^{(n-2)-n+2}\big) = 6-3\big(1+2^0\big) = 0, \quad (10.24)$$

$$6-\big(n+1-[n-3]\big)\big(1+2^{(n-3)-n+2}\big) = 6-4\big(1+2^{-1}\big) = 0, \quad (10.25)$$

$$6-\big(n+1-[n-4]\big)\big(1+2^{(n-4)-n+2}\big) = 6-5\big(1+2^{-2}\big) = -\frac{1}{24}, \quad (10.26)$$

and so on. Hence, we can end the summation at $k = n-4$:

$$(\text{error}) = \frac{1}{6}\sum_{k=0}^{n-4}\binom{n}{k}x_0^k\,h^{n+1-k}\,\frac{6-\big(n+1-k\big)\big(1+2^{k-n+2}\big)}{n+1-k}. \quad (10.27)$$

If $n < 4$, the summation contains no terms, so the error is zero. If $n \geq 4$, the error is nonzero. Hence, the RK4 method is *exact* for $f(x) = \text{const}$, $f(x) = x$, $f(x) = x^2$ and $f(x) = x^3$, but *not* for a general power $f(x) = x^n$. This proof still holds for a polynomial due to linearity arguments. □

10.2 Runge-Kutta-Fehlberg (RKF) method

Some integration methods automatically provide an error estimate. An example is the **Runge-Kutta-Fehlberg method** (Fehlberg, 1966):

$$y_{n+1} = y_n + \frac{16}{135}\,k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}\,k_4 - \frac{9}{50}\,k_5 + \frac{2}{55}\,k_6, \quad (10.28)$$

where

$$k_1 = h\,f\big(\,x_n\,,\,y_n\,\big), \quad (10.29)$$

$$k_2 = h\,f\left(\,x_n + \frac{h}{4}\,,\,y_n + \frac{k_1}{4}\,\right), \quad (10.30)$$

$$k_3 = h\,f\left(\,x_n + \frac{3}{8}\,h\,,\,y_n + \frac{3\,k_1 + 9\,k_2}{32}\,\right), \quad (10.31)$$

$$k_4 = h\,f\left(\,x_n + \frac{12}{13}\,h\,,\,y_n + \frac{1932\,k_1 - 7200\,k_2 + 7296\,k_3}{2197}\,\right), \quad (10.32)$$

$$k_5 = h\,f\left(\,x_n + h\,,\,y_n + \frac{439}{216}\,k_1 - 8k_2 + \frac{3680}{513}\,k_3 - \frac{845}{4104}\,k_4\,\right), \quad (10.33)$$

$$k_6 = h\,f\left(\,x_n + \frac{h}{2}\,,\,y_n - \frac{8}{27}\,k_1 + 2k_2 - \frac{3544}{2565}\,k_3 + \frac{1859}{4104}\,k_4 - \frac{11}{40}\,k_5\,\right). \quad (10.34)$$

The single-step error estimate is given in terms of the same coefficients:

$$(\text{Error}) \;=\; \frac{1}{360}\,k_1 - \frac{128}{4275}\,k_3 - \frac{2197}{75240}\,k_4 + \frac{1}{50}\,k_5 + \frac{2}{55}\,k_6. \tag{10.35}$$

The easily calculable error term can make the RKF method preferable to the RK4 method. It has global truncation error $\mathcal{O}(h^5)$, but is less computationally efficient than the RK4 method because more derivative function evaluations are required per step (six compared to four, Hoffman, 1992, §7.9). This can be offset by increasing the step size.

Accuracy can be improved by breaking an integration over the interval $[x_{\min}, x_{\max}]$ into n separate integrals with step size

$$h \;=\; \frac{x_{\max} - x_{\min}}{n-1}. \tag{10.36}$$

Treating the single-step errors as statistical errors, which add in quadrature, the total numerical error will be

$$\sigma_{\text{tot}} \;=\; \sqrt{\sum_{i=1}^{n} \sigma_i^2}, \tag{10.37}$$

where σ_i is the error estimate for each step. Although σ_{tot} is larger than any of the σ_i, the act of reducing the step size h reduces all of the σ_i, so the total error decreases as the step-size decreases (i.e. as the number of steps increases).

10.3 Adams-Bashforth methods

The **first-, second-, third-, fourth- and fifth-order Adams-Bashforth methods** estimate y_{n+1} to be[3]

$$y_{n+1}^{\text{AB1}} \;=\; y_n + h\,f(x_n\,,\,y_n) + \mathcal{O}(h^2), \tag{10.38}$$

$$y_{n+1}^{\text{AB2}} \;=\; y_n + \frac{h}{2}\Big[3\,f(x_n, y_n) - f(x_{n-1}, y_{n-1})\Big] = \mathcal{O}(h^3), \tag{10.39}$$

$$y_{n+1}^{\text{AB3}} \;=\; y_n + \frac{h}{12}\Big[23\,f(x_n, y_n) - 16\,f(x_{n-1}, y_{n-1}) + 5\,f(x_{n-2}, y_{n-2})\Big] + \mathcal{O}(h^4), \tag{10.40}$$

$$y_{n+1}^{\text{AB4}} \;=\; y_n + \frac{h}{24}\Big[55\,f(x_n, y_n) - 59\,f(x_{n-1}, y_{n-1}) + 37\,f(x_{n-2}, y_{n-2}) - 9\,f(x_{n-3}, y_{n-3})\Big] + \mathcal{O}(h^5), \tag{10.41}$$

$$\begin{aligned} y_{n+1}^{\text{AB5}} \;=\; y_n + \frac{h}{720}\Big[&1901\,f(x_n, y_n) - 2774\,f(x_{n-1}, y_{n-1}) + 2616\,f(x_{n-2}, y_{n-2}) \\ &- 1274\,f(x_{n-3}, y_{n-3}) + 251\,f(x_{n-4}, y_{n-4})\Big] + \mathcal{O}(h^6). \end{aligned} \tag{10.42}$$

Note that the AB1 method is equivalent to the Euler method. the k^{th}-order Adams-Bashforth method has single-step error $\mathcal{O}(h^{k+1})$ (Hairer et al., 1993). Although accuracy increases as the order increases, y_{n+1} depends on earlier values of y. This introduces two complications:

1. We must maintain copies of the last few lists of y-values used, and constantly shift them when we move to the next timestep (e.g. move all the values from y_n to y_{n-1}).

2. We cannot use higher-order methods for the first few timesteps because previous sets of y values do not yet exist.

[3] The second- and higher-order methods are usually written such that y_{n+2} and so on are calculated; that is, skipping to coordinates further ahead. However, we are only interested in calculating the next coordinate y_{n+1}, and so we rewrite the indices here to suit.

The latter step is accounted for by increasing the order every step until the desired method can be used. For example, if we wish to use the AB3 method, the process is as follows.

1. To start, create four arrays `y_nplus1`, `y_n`, `y_nminus1` and `y_nminus2`, which we will use to store the values of y at the next, current and previous two timesteps.
2. At the initial time $n = 0$, we only have the initial values y_0. Populate `y_n` with these. To calculate `y_nplus1` (which are y_1), we must use the AB1 method.
3. At the second timestep $n = 1$, copy the values from `y_n` to `y_nminus1` (which are y_0) and then copy the values from `y_nplus1` to overwrite `y_n` (which are y_1). Optionally, reset the values in `y_nplus1` (which are y_2, to be calculated) to zero, to avoid confusion or potential errors. The most accurate method we can use to calculate `y_nplus1` is the AB2 method.
4. At the third timestep $n = 2$, copy the values from `y_nminus1` to `y_nminus2` (which are y_0), copy the values from `y_n` to overwrite `y_nminus1` (which are y_1) and then copy the values from `y_nplus1` to overwrite `y_n` (which are y_2). Optionally, reset the values in `y_nplus1` (which are y_3, to be calculated) to zero. All of our arrays are populated, so we can use the AB3 method for this and all subsequent steps. Note that, for example, at the fourth timestep, we will lose y_0 when `y_nminus2` is overwritten; this is acceptable since y_0 are no longer required.

10.4 Multidimensional integrals and volume elements

The expressions coming in §10.5 may appear abstract and unwieldy, but they are highly efficient and generalisable to any coordinate system. To the best of the author's knowledge, they represent original work. Hence, it is worthwhile to spend some time developing them in detail.

Firstly, let us clearly state the notation we will use. Let (u_1, u_2, u_3) be a curvilinear coordinate system with Lamé coefficients h_1, h_2, h_3. Let $u_{1,\min}, u_{1,\max}, u_{2,\min}, u_{2,\max}, u_{3,\min}, u_{3,\max} \in \mathbb{R}$ and let $f(u_1, u_2, u_3)$ be a function defined for $u_1 \in [u_{1,\min}, u_{1,\max}]$, $u_2 \in [u_{2,\min}, u_{2,\max}]$, $u_3 \in [u_{3,\min}, u_{3,\max}]$. As in the 1D case, accuracy of numerical integration will be improved by splitting the domain:

- Split the interval $[u_{1,\min}, u_{1,\max}]$ into $N_1 \in \mathbb{N}$ values $u_{1,i}$ enumerated by $i \in \mathbb{N}_0 : 0 \leq i \leq N_1 - 1$, with $u_{1,0} = u_{1,\min}$, $u_{1,N_1-1} = u_{1,\max}$ and $\forall\, i \in \mathbb{N}_0 : 0 \leq i \leq N_1 - 2$, we have $u_{1,i} < u_{1,i+1}$.
- Split the interval $[u_{2,\min}, u_{2,\max}]$ into $N_2 \in \mathbb{N}$ values $u_{2,j}$ enumerated by $j \in \mathbb{N}_0 : 0 \leq j \leq N_2 - 1$, with $u_{2,0} = u_{2,\min}$, $u_{2,N_2-1} = u_{2,\max}$ and $\forall\, j \in \mathbb{N}_0 : 0 \leq j \leq N_2 - 2$, we have $u_{2,j} < u_{2,j+1}$.
- Split the interval $[u_{3,\min}, u_{3,\max}]$ into $N_3 \in \mathbb{N}$ values $u_{3,k}$ enumerated by $k \in \mathbb{N}_0 : 0 \leq k \leq N_3 - 1$, with $u_{3,0} = u_{3,\min}$, $u_{3,N_3-1} = u_{3,\max}$ and $\forall\, k \in \mathbb{N}_0 : 0 \leq k \leq N_3 - 2$, we have $u_{3,k} < u_{3,k+1}$.

Let us denote linear spacing as $\Delta u_1 = \frac{u_{1,\max} - u_{1,\min}}{N_1 - 1}$, and similarly $\Delta u_2 = \frac{u_{2,\max} - u_{2,\min}}{N_2 - 1}$ and $\Delta u_3 = \frac{u_{3,\max} - u_{3,\min}}{N_3 - 1}$.

Now, the 3D integral over the volume spanned by all three coordinates is

$$I_{3\mathrm{D}} \equiv \int_{u_{3,\min}}^{u_{3,\max}} \int_{u_{2,\min}}^{u_{2,\max}} \int_{u_{1,\min}}^{u_{1,\max}} f(u_1, u_2, u_3)\, h_1\, h_2\, h_3\; du_1\, du_2\, du_3. \tag{10.43}$$

Define the **finite volume element** $\Delta V_{i,j,k}$ to be the volume bounded by the eight points $(u_{1,i}, u_{2,j}, u_{3,k})$, $(u_{1,i}, u_{2,j+1}, u_{3,k})$, $(u_{1,i+1}, u_{2,j}, u_{3,k})$, $(u_{1,i+1}, u_{2,j+1}, u_{3,k})$, $(u_{1,i}, u_{2,j}, u_{3,k+1})$, $(u_{1,i}, u_{2,j+1}, u_{3,k+1})$, $(u_{1,i+1}, u_{2,j}, u_{3,k+1})$, $(u_{1,i+1}, u_{2,j+1}, u_{3,k+1})$:

$$\Delta V_{i,j,k} \equiv \int_{u_{3,k}}^{u_{3,k+1}} \int_{u_{2,j}}^{u_{2,j+1}} \int_{u_{1,i}}^{u_{1,i+1}} h_1\, h_2\, h_3\, du_1\, du_2\, du_3. \tag{10.44}$$

The 2D integral over the surface spanned by two of the coordinates (say, without loss of generality, u_1 and u_2) is

$$I_{2\mathrm{D}} \equiv \int_{u_{2,\min}}^{u_{2,\max}} \int_{u_{1,\min}}^{u_{1,\max}} f(u_1, u_2)\, h_1\, h_2\, du_1\, du_2. \tag{10.45}$$

Define the **finite area element** $\Delta A_{i,j}$ to be the area bounded by the four points $(u_{1,i}, u_{2,j})$, $(u_{1,i}, u_{2,j+1})$, $(u_{1,i+1}, u_{2,j})$, $(u_{1,i+1}, u_{2,j+1})$:

$$\Delta A_{i,j} \equiv \int_{u_{2,j}}^{u_{2,j+1}} \int_{u_{1,i}}^{u_{1,i+1}} h_1\, h_2\, du_1\, du_2. \tag{10.46}$$

The 1D integral over the line covered by one coordinate (say u_1, from which we remove the subscript since there are no other coordinates to distinguish it from) is

$$I_{\mathrm{1D}} \equiv \int_{u_{\min}}^{u_{\max}} f(u)\, h\, du. \tag{10.47}$$

Define the **finite line element** Δu_i to be the length of the straight line connecting the two points u_i and u_{i+1}:

$$\Delta u_i \equiv \int_{u_i}^{u_{i+1}} h\, du. \tag{10.48}$$

The expression for a finite line element is especially useful for integrals in which a change of variables has been performed; for example, from some variable x to some other variable t related by $x = \cos(t)$.

The finite volume, area and line elements clearly depend only on the coordinate system, not the function, so they can be calculated in advance and stored in multidimensional arrays separate to the function values. Doing so avoids repeat calculations when many numerical integrals are required over the same coordinates.

Proposition 10.3. *The finite volume elements for Cartesian, cylindrical polar and spherical polar coordinate systems are*

$$\Delta V_{i,j,k}^{\mathrm{Cartesian}} = \big[x_{i+1} - x_i\big]\,\big[y_{j+1} - y_j\big]\,\big[z_{k+1} - z_k\big], \tag{10.49}$$

$$\Delta V_{i,j,k}^{\mathrm{cylindrical}} = \frac{1}{2}\,\big[\rho_{i+1}^2 - \rho_i^2\big]\,\big[\phi_{j+1} - \phi_j\big]\,\big[z_{k+1} - z_k\big], \tag{10.50}$$

$$\Delta V_{i,j,k}^{\mathrm{spherical}} = \frac{1}{3}\,\big[r_{i+1}^3 - r_i^3\big]\,\big[\cos(\theta_j) - \cos(\theta_{j+1})\big]\,\big[\phi_{k+1} - \phi_k\big]. \tag{10.51}$$

If we use linear spacing for all coordinates, these expressions become

$$\Delta V_{i,j,k}^{\mathrm{Cartesian}} = \Delta x\, \Delta y\, \Delta z, \tag{10.52}$$

$$\Delta V_{i,j,k}^{\mathrm{cylindrical}} = \Delta\rho\, \Delta\phi\, \Delta z \left[\rho_i + \frac{1}{2}\,\Delta\rho\right], \tag{10.53}$$

$$\Delta V_{i,j,k}^{\mathrm{spherical}} = \Delta r\, \Delta\phi \left[r_i^2 + r_i\, \Delta r + \frac{1}{3}\,\Delta r^2\right] \left[\cos(\theta_j) - \cos(\theta_{j+1})\right]. \tag{10.54}$$

Proof. The process is the same in all coordinate systems; let us show the spherical polar result explicitly. We have

$$\Delta V_{i,j,k}^{\mathrm{spherical}} = \int_{\phi_k}^{\phi_{k+1}} \int_{\theta_j}^{\theta_{j+1}} \int_{r_i}^{r_{i+1}} h_r\, h_\theta\, h_\phi\, dr\, d\theta\, d\phi \tag{10.55}$$

$$= \left(\int_{r_i}^{r_{i+1}} r^2\, dr\right) \left(\int_{\theta_j}^{\theta_{j+1}} \sin(\theta)\, d\theta\right) \left(\int_{\phi_k}^{\phi_{k+1}} d\phi\right) \tag{10.56}$$

$$= \left[\frac{1}{3}\, r^3\right]_{r_i}^{r_{i+1}} \Big[-\cos(\theta)\Big]_{\theta_j}^{\theta_{j+1}} \Big[\phi\Big]_{\phi_k}^{\phi_{k+1}}, \tag{10.57}$$

which yields the given result. Now suppose that we use linear spacing for all coordinates. For the radial component, we have

$$r_{i+1}^3 = \big(r_i + \Delta r\big)^3 = r_i^3 + 3\, r_i^2\, \Delta r + 3\, r_i\, \Delta r^2 + \Delta r^3, \tag{10.58}$$

$$\Rightarrow\ \frac{1}{3}\,\big[r_{i+1}^3 - r_i^3\big] = \frac{1}{3}\Big[3\, r_i^2\, \Delta r + 3\, r_i\, \Delta r^2 + \Delta r^3\Big] = \Delta r \left[r_i^2 + r_i\, \Delta r + \frac{1}{3}\,\Delta r^2\right]. \tag{10.59}$$

For $a, b \in \mathbb{R}$, we cannot express $\cos(a+b)$ in terms of $\cos(a)$ and $\cos(b)$, so the θ term does not simplify. Clearly $\phi_{k+1} - \phi_k = \Delta\phi$ independent of k. Putting these together, we obtain the given result. □

Extensively throughout the numerical evolution, we will require the volume integral of an axisymmetric function in spherical polar coordinates, integrating over all $\phi \in [0, 2\pi]$. This is equivalent to 2π times the surface integral over r and θ, which has finite area element

$$\Delta A^{r,\theta}_{i,j} \;=\; \frac{1}{2}\left[r^2_{i+1} - r^2_i\right]\left[\theta_{j+1} - \theta_j\right], \tag{10.60}$$

and with linear coordinate spacing,

$$\Delta A^{r,\theta}_{i,j} \;=\; \Delta r\,\Delta\theta\left[r_i + \frac{1}{2}\,\delta r\right]. \tag{10.61}$$

We will also frequently use 1D integrals over the polar angle θ, for which[4]

$$\Delta\theta_j \;=\; \cos(\theta_j) - \cos(\theta_{j+1}). \tag{10.62}$$

These are obtained by the same process as in Proposition 10.3.

When comparing numerical results to known indefinite integrals, it is important to properly consider the treatment of upper and lower integration bounds.

Proposition 10.4. *Suppose that the indefinite integral of Eq. (10.43) is known to be $I(u_1, u_2, u_3)$. Then, the definite integral is*

$$\begin{aligned} I_{3D} \;=\; & I(u_{1,\max}, u_{2,\max}, u_{3,\max}) - I(u_{1,\max}, u_{2,\max}, u_{3,\min}) - I(u_{1,\max}, u_{2,\min}, u_{3,\max}) + I(u_{1,\max}, u_{2,\min}, u_{3,\min}) \\ & - I(u_{1,\min}, u_{2,\max}, u_{3,\max}) + I(u_{1,\min}, u_{2,\max}, u_{3,\min}) + I(u_{1,\min}, u_{2,\min}, u_{3,\max}) - I(u_{1,\min}, u_{2,\min}, u_{3,\min}). \end{aligned} \tag{10.63}$$

Proof. Eq. (10.43) can be separated into independent integrals over each coordinate:

$$I_{3D} \;=\; \left(\int_{u_{1,\min}}^{u_{1,\max}} f_1(u_1)\,h_1\,du_1\right)\left(\int_{u_{2,\min}}^{u_{2,\max}} f_2(u_2)\,h_2\,du_2\right)\left(\int_{u_{3,\min}}^{u_{3,\max}} f_3(u_3)\,h_3\,du_3\right), \tag{10.64}$$

where f_1, f_2, f_3 are the parts of the function that depend on u_1, u_2, u_3 respectively. Suppose that the indefinite integrals of f_1, f_2, f_3 are known to be I_1, I_2, I_3 respectively. Then, the indefinite integral of the full function $f(u_1, u_2, u_3)$ is the product

$$I(u_1, u_2, u_3) \;=\; I_1(u_1)\,I_2(u_2)\,I_3(u_3). \tag{10.65}$$

Definite integrals are obtained by

$$I_{3D} \;=\; \Big[I_1\Big]_{u_{1,\min}}^{u_{1,\max}}\Big[I_2\Big]_{u_{2,\min}}^{u_{2,\max}}\Big[I_3\Big]_{u_{3,\min}}^{u_{3,\max}} \tag{10.66}$$

$$\;=\; \Big[I_1(u_{1,\max}) - I_1(u_{1,\min})\Big]\Big[I_2(u_{2,\max}) - I_2(u_{2,\min})\Big]\Big[I_3(u_{3,\max}) - I_3(u_{3,\min})\Big] \tag{10.67}$$

$$\begin{aligned} \;=\; & \Big[I_1(u_{1,\max}) - I_1(u_{1,\min})\Big]\Big[I_2(u_{2,\max})\,I_3(u_{3,\max}) - I_2(u_{2,\max})\,I_3(u_{3,\min}) \\ & \qquad - I_2(u_{2,\min})\,I_3(u_{3,\max}) + I_2(u_{2,\min})\,I_3(u_{3,\min})\Big] \end{aligned} \tag{10.68}$$

$$\begin{aligned} \;=\; & I_1(u_{1,\max})\,I_2(u_{2,\max})\,I_3(u_{3,\max}) - I_1(u_{1,\max})\,I_2(u_{2,\max})\,I_3(u_{3,\min}) \\ & - I_1(u_{1,\max})\,I_2(u_{2,\min})\,I_3(u_{3,\max}) + I_1(u_{1,\max})\,I_2(u_{2,\min})\,I_3(u_{3,\min}) \\ & + I_1(u_{1,\min})\,I_2(u_{2,\max})\,I_3(u_{3,\max}) - I_1(u_{1,\min})\,I_2(u_{2,\max})\,I_3(u_{3,\min}) \\ & - I_1(u_{1,\min})\,I_2(u_{2,\min})\,I_3(u_{3,\max}) + I_1(u_{1,\min})\,I_2(u_{2,\min})\,I_3(u_{3,\min}), \end{aligned} \tag{10.69}$$

which combine to give the quoted result. □

[4] We will always enumerate θ with the index j in the code, so write j here for consistency.

In the same way, if the indefinite integral of Eq. (10.45) is $I(u_1, u_2)$, the definite integral is

$$I_{2\mathrm{D}} \;=\; I(u_{1,\max}, u_{2,\max}) - I(u_{1,\max}, u_{2,\min}) - I(u_{1,\min}, u_{2,\max}) + I(u_{1,\min}, u_{2,\min}). \tag{10.70}$$

Although one-dimensional indefinite integrals require just two function evaluations, i.e.

$$I_{1\mathrm{D}} \;=\; \int_{u_{\max}}^{u_{\min}} f(u)\, h_u\, du \;=\; I(u_{\max}) - I(u_{\min}), \tag{10.71}$$

this does not carry over to higher dimensions: to return the required definite integral, we must perform eight function evaluations in 3D and four in 2D. This is perhaps not appreciated when performing such calculations by hand, as the integrals are usually separated and the bounds applied during intermediate steps to simply the expression. It is a subtle point, but becomes necessary when defining indefinite integrals via a function on a computer code.

10.5 Trapezium rule

While indispensable for 1D numerical integration, the RK4, RKF and Adams-Bashforth methods have two main drawbacks:

1. The RK4 and RKF methods require function evaluations at intermediate points between x_0 and $x_0 + h$, which is impossible for discrete data without some interpolation method.

2. Neither scale easily to higher dimensions.

The trapezium rule does not share these drawbacks. Its form is well known in 1D Cartesian coordinates, but less so in general coordinate systems or higher dimensions. Let us first recap the 1D Cartesian trapezium rule and then use this as the basis for a formulation in other coordinate systems.

Definition 10.5. *Let $a, b \in \mathbb{R}$ and let $f(x)$ be a function defined on $[a, b]$ in 1D Cartesian coordinates. The* **trapezium rule***, also called the* **trapezoidal rule***, approximates the integral of $f(x)$ over $[a, b]$:*

$$\int_a^b f(x)\, dx \;\approx\; \frac{f(a) + f(b)}{2} \cdot (b - a). \tag{10.72}$$

The value above is equal to the area of the trapezium defined by the four points $\big(a, 0\big)$, $\big(b, 0\big)$, $\big(a, f(a)\big)$ and $\big(b, f(b)\big)$, hence the name; this can be seen in the left-hand graph of Figure 10.1. It can equivalently be interpreted as the average value of $f(x)$ at the two points $x = a$ and $x = b$, times the separation between them.

Accuracy is improved by splitting the interval $[a, b]$ into $N \in \mathbb{N}$ values of x with arbitrary spacing, enumerated by $i \in \mathbb{N}_0 : 0 \leq i \leq N - 1$; that is, $N - 1$ subintervals.[5] We have $x_0 = a$, $x_{N-1} = b$ and $\forall\, i \in \mathbb{N}_0 : 0 \leq i \leq N - 2$, we have $x_i < x_{i+1}$. Then, the integral is approximately

$$\int_a^b f(x)\, dx \;\approx\; \frac{1}{2} \sum_{i=0}^{N-2} \big[f(x_{i+1}) + f(x_i)\big]\, (x_{i+1} - x_i). \tag{10.73}$$

Figure 10.1 visualises this process.

[5] **Care must be taken** when comparing results in this section to the literature. Some definitions use $1 \leq i \leq N$, and some use N *subintervals* such that there are $N + 1$ values of x. We choose to "count the fence posts, not the fences" consistently throughout the project, because it will prove easier when building lists of coordinates in the main evolution code. We also enumerate i from 0 for consistency with `C++` array indexing, so that expressions in this report can be directly compared to the functions we write.

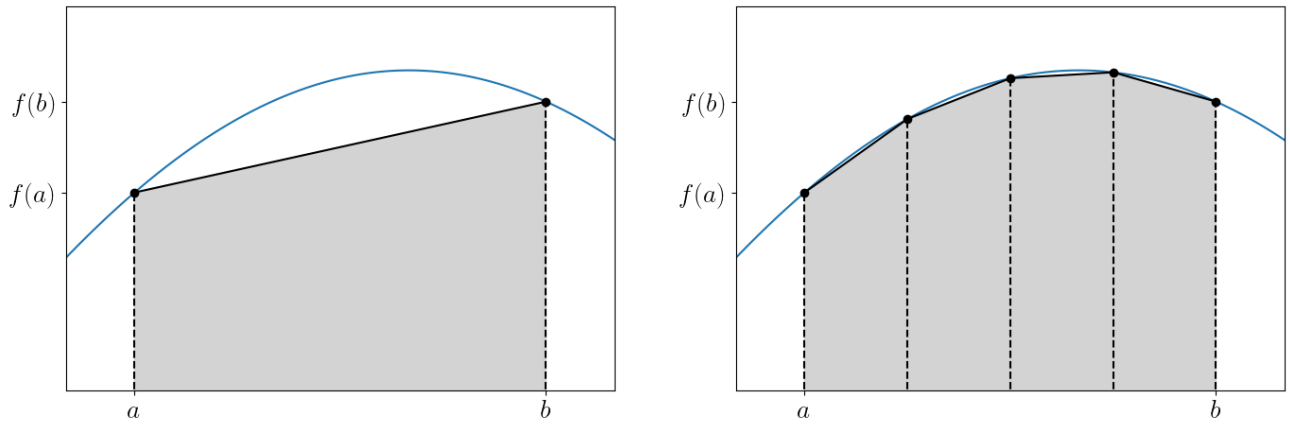

Figure 10.1: Visualisation of the integral of a function between two points by the trapezium rule. *Left:* One step. *Right:* $N = 4$ steps. The example used here is $\sin(x)$ from $0.3\,\pi$ to $0.6\,\pi$; the absolute relative error is 0.08 and 0.005 respectively.
Produced by code: `Plot_Trapezium_Rule_Visualisation.py`

Proposition 10.6. *In Cartesian coordinates, the trapezium rule with N values of x can be equivalently written*

$$\int_a^b f(x)\,dx \;\approx\; \frac{1}{2}\left(f(x_0)\,[x_1 - x_0] + f(x_{N-1})[x_{N-1} - x_{N-2}] + \sum_{i=1}^{N-2} f(x_i)[x_{i+1} - x_{i-1}] \right). \tag{10.74}$$

Writing $x_0 = a$ and $x_{N-1} = b$ slightly simplifies the notation.

Proof. Denote by I_1^N the integral defined in Eq. (10.73) with N values of x, and by I_2^N the integral defined in Eq. (10.74). We prove the statement by induction. Our base case is $N = 2$ since we need at least two values of x to integrate between. For the original expression,

$$2\,I_1^2 \;=\; \sum_{i=0}^{2-2} \big[f(x_{i+1}) + f(x_i)\big]\,(x_{i+1} - x_i) \;=\; \sum_{i=0}^{0} \big[f(x_{i+1}) + f(x_i)\big]\,(x_{i+1} - x_i) \tag{10.75}$$

$$=\; \big[f(x_{0+1}) + f(x_0)\big]\,(x_{0+1} - x_0) \;=\; \big[f(x_1) + f(x_0)\big](x_1 - x_0). \tag{10.76}$$

For the proposed expression:

$$2\,I_2^2 \;=\; f(x_0)\,[x_1 - x_0] + f(x_{2-1})[x_{2-1} - x_{2-2}] + \sum_{i=1}^{2-2} f(x_i)[x_{i+1} - x_{i-1}] \tag{10.77}$$

$$=\; f(x_0)\,[x_1 - x_0] + f(x_1)[x_1 - x_0] + \sum_{i=1}^{0} f(x_i)[x_{i+1} - x_{i-1}]. \tag{10.78}$$

Since $\nexists\, i \in \mathbb{N}_0 : (i \geq 1) \wedge (i \leq 0)$, the summation condition is never fulfilled, so the sum evaluates to zero. We have

$$2\,I_2^2 \;=\; f(x_0)\,[x_1 - x_0] + f(x_1)[x_1 - x_0] + 0 \;=\; \big[f(x_0) + f(x_1)\big]\,(x_1 - x_0) \;=\; 2\,I_1^2. \tag{10.79}$$

Assume now that the statement holds for N, that is, $I_1^N = I_2^N$. To complete the proof, we must show that this

implies that the statement holds for $N+1$. Our target value is

$$2\,I_1^{N+1} = \sum_{i=0}^{N+1-2} \big[f(x_{i+1}) + f(x_i)\big]\,(x_{i+1} - x_i) = \sum_{i=0}^{N-1} \big[f(x_{i+1}) + f(x_i)\big]\,(x_{i+1} - x_i) \tag{10.80}$$

$$= \sum_{i=0}^{N-2} \big[f(x_{i+1}) + f(x_i)\big]\,(x_{i+1} - x_i) + \big[f(x_{N-1+1}) + f(x_{N-1})\big]\,(x_{N-1+1} - x_{N-1}) \tag{10.81}$$

$$= 2\,I_1^N + \big[f(x_N) + f(x_{N-1})\big]\,(x_N - x_{N-1}). \tag{10.82}$$

We have

$$2\,I_2^{N+1} = f(x_0)\,\big[x_1 - x_0\big] + f(x_{N+1-1})\big[x_{N+1-1} - x_{N+1-2}\big] + \sum_{i=1}^{N+1-2} f(x_i)\big[x_{i+1} - x_{i-1}\big] \tag{10.83}$$

$$= f(x_0)\,\big[x_1 - x_0\big] + f(x_N)\big[x_N - x_{N-1}\big] + \sum_{i=1}^{N-1} f(x_i)\big[x_{i+1} - x_{i-1}\big]. \tag{10.84}$$

Note that

$$\sum_{i=1}^{N-1} f(x_i)\big[x_{i+1} - x_{i-1}\big] = \sum_{i=1}^{N-2} f(x_i)\big[x_{i+1} - x_{i-1}\big] + f(x_{N-1})\big[x_{N-1+1} - x_{N-1-1}\big] \tag{10.85}$$

$$= \sum_{i=1}^{N-2} f(x_i)\big[x_{i+1} - x_{i-1}\big] + f(x_{N-1})\big[x_N - x_{N-2}\big], \tag{10.86}$$

and that

$$f(x_0)\,\big[x_1 - x_0\big] + \sum_{i=1}^{N-2} f(x_i)\big[x_{i+1} - x_{i-1}\big] = 2\,I_2^N - f(x_{N-1})\big[x_{N-1} - x_{N-2}\big], \tag{10.87}$$

giving

$$2\,I_2^{N+1} = 2\,I_2^N - f(x_{N-1})\big[x_{N-1} - x_{N-2}\big] + f(x_N)\big[x_N - x_{N-1}\big] + f(x_{N-1})\big[x_N - x_{N-2}\big] \tag{10.88}$$

$$= 2\,I_2^N + f(x_{N-1})\big[-x_{N-1} + x_N\big] + f(x_N)\big[x_N - x_{N-1}\big] \tag{10.89}$$

$$= 2\,I_2^N + \big[f(x_{N-1}) + f(x_N)\big]\,\big[x_N - x_{N-1}\big] \tag{10.90}$$

$$= 2\,I_1^N + \big[f(x_{N-1}) + f(x_N)\big]\,\big[x_N - x_{N-1}\big] \tag{10.91}$$

$$= 2\,I_1^{N+1}, \tag{10.92}$$

which completes the proof. □

The expression in Eq. (10.74) comes about because terms cancel when expanding the brackets in Eq. (10.73). For $N > 2$, Eq. (10.74) is more computationally efficient than Eq. (10.73) because each term in the sum contains three terms as opposed to four. It is true that there is only one function evaluation as opposed to two, but values should be pre-calculated and stored in an array for better efficiency anyway - especially for multidimensional integration - so this alone should not offer a speed increase.

If the subintervals have equal spacing $h = \frac{b-a}{N-1}$, the integral is approximately[6]

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\sum_{i=0}^{N-2}\big[f(x_{i+1}) + f(x_i)\big] = \frac{h}{2}\big[f(x_0) + f(x_{N-1})\big] + h\sum_{i=1}^{N-2} f(x_i). \tag{10.93}$$

An estimate of the error term in the 1D trapezium rule, which scales as $\mathcal{O}(N^{-3})$, is given by Atkinson (1989). However, it requires the evaluation of $f'(x)$, which for discrete data is difficult to calculate accurately. We do not attempt to implement an error estimate, nor to extend it to higher dimensions, but highlight that error estimation is possible if required.

[6] Note that $x_{i+1} - x_{i-1} = 2h$, so the $\frac{1}{2}$ multiplying the sum is cancelled.

Definition 10.7. *Let u be a one-dimensional coordinate with Lamé coefficient h. The trapezium rule generalises to approximate $I_{1\mathrm{D}}$ as*

$$I_{1\mathrm{D}} \;=\; \int_{u_{\min}}^{u_{\max}} f(u)\,h\,du \;\approx\; \frac{f(u_{\min})+f(u_{\max})}{2}\;\cdot h\,(u_{\max}-u_{\min}). \tag{10.94}$$

Splitting the interval $[u_{\min}, u_{\max}]$ into $N \in \mathbb{N}$ values of u with arbitrary spacing as in §10.4, a general 1D integral is approximately

$$I_{1\mathrm{D}} \;\approx\; \frac{1}{2}\sum_{i=0}^{N-2}\big[f(u_{i+1})+f(u_i)\big]\,\Delta u_i. \tag{10.95}$$

Definition 10.8. *The trapezium rule generalises to approximate $I_{2\mathrm{D}}$ as the mean of the function evaluated at the four vertices of the region defined by the integration limits, multiplied by the area of the region:*

$$\begin{aligned} I_{2\mathrm{D}} \;=\; & \frac{1}{4}\Big[f(u_{1,\min},u_{2,\min})+f(u_{1,\min},u_{2,\max})+f(u_{1,\max},u_{2,\min})+f(u_{1,\max},u_{2,\max})\Big] \\ & \cdot h_1\,h_2\,(u_{1,\max}-u_{1,\min})\,(u_{2,\max}-u_{2,\min}). \end{aligned} \tag{10.96}$$

Splitting the domain as in §10.4, a general 2D integral is approximately

$$I_{2\mathrm{D}} \;\approx\; \frac{1}{4}\sum_{j=0}^{N_2-2}\sum_{i=0}^{N_1-2}\Big[f(u_{1,i},u_{2,j})+f(u_{1,i},u_{2,j+1})+f(u_{1,i+1},u_{2,j})+f(u_{1,i+1},u_{2,j+1})\Big]\,\Delta A_{i,j}. \tag{10.97}$$

Definition 10.9. *The trapezium rule generalises to approximate $I_{3\mathrm{D}}$ as the mean of the function evaluated at the eight vertices of the region defined by the integration limits, multiplied by the volume of the region:*

$$\begin{aligned} I_{3\mathrm{D}} \;=\; & \frac{1}{8}\Big[f(u_{1,\min},u_{2,\min},u_{3,\min})+f(u_{1,\min},u_{2,\min},u_{3,\max})+f(u_{1,\min},u_{2,\max},u_{3,\min})+f(u_{1,\min},u_{2,\max},u_{3,\max}) \\ & +f(u_{1,\max},u_{2,\min},u_{3,\min})+f(u_{1,\max},u_{2,\min},u_{3,\max})+f(u_{1,\max},u_{2,\max},u_{3,\min})+f(u_{1,\max},u_{2,\max},u_{3,\max})\Big] \\ & \cdot h_1\,h_2\,h_3\,(u_{1,\max}-u_{1,\min})\,(u_{2,\max}-u_{2,\min})\,(u_{3,\max}-u_{3,\min}). \end{aligned} \tag{10.98}$$

Splitting the domain as in §10.4, a general 3D integral is approximately

$$\begin{aligned} I_{3\mathrm{D}} \;\approx\; & \frac{1}{8}\sum_{k=0}^{N_3-2}\sum_{j=0}^{N_2-2}\sum_{i=0}^{N_1-2}\Big[f(u_{1,i},u_{2,j},u_{3,k})+f(u_{1,i},u_{2,j},u_{3,k+1})+f(u_{1,i},u_{2,j+1},u_{3,k})+f(u_{1,i},u_{2,j+1},u_{3,k+1}) \\ & +f(u_{1,i+1},u_{2,j},u_{3,k})+f(u_{1,i+1},u_{2,j},u_{3,k+1})+f(u_{1,i+1},u_{2,j+1},u_{3,k})+f(u_{1,i+1},u_{2,j+1},u_{3,k+1})\Big]\,\Delta V_{i,j,k}. \end{aligned} \tag{10.99}$$

10.6 Testing implementation

To test our trapezium method implementation, we define the following functions within the header file `Trapezium_rule.h`:

1. `double integral_trapezium_1D_Cartesian( std::vector<double> f, std::vector<double> du )`
2. `double integral_trapezium_2D( std::vector< std::vector<double> > f, std::vector< std::vector<double> > dA )`
3. `double integral_trapezium_3D( std::vector< std::vector< std::vector<double> > > f, std::vector< std::vector< std::vector<double> > > dV )`

We then write the following `C++` codes to test the implementation in different coordinate systems.

1. To test the 1D Cartesian implementation, we write `Test_trapezium_rule_1D_Cartesian.cpp`, a code that numerically integrates the function $f(x) = x^2$ between $x_{\min} = 1$ and $x_{\max} = 2$ with $N = 10$ linearly spaced values of x. The exact result is

$$\int_{x_{\min}}^{x_{\max}} f(x)\,dx \;=\; \left[\frac{1}{3}\,x^3\right]_1^3 \;=\; \frac{7}{3}, \tag{10.100}$$

   which is implemented by defining the indefinite integral as a function to be evaluated at the upper and lower bounds. The output is

```
Numerical: 2.33539
Exact    : 2.33333
```

   We also test the efficient expression in Proposition 10.74 and the expressions for linear spacing in Eq. (10.93), which return the same result.

2. To test the 2D Cartesian implementation, we write `Test_trapezium_rule_2D_Cartesian.cpp` to integrate $f(x, y) = x^2 + y^2$ from $x_{\min} = 1$ to $x_{\max} = 2$, and $y_{\min} = 3.5$ to $y_{\max} = 5$, with $N_x = 10$ values of x and $N_y = 20$ values of y, all equally spaced. The exact result is

$$\int_{y_{\min}}^{y_{\max}} \int_{x_{\min}}^{x_{\max}} f(x,y)\,dx\,dy \;=\; \left[\frac{1}{3}\,(x^3\,y + x\,y^3)\right]_{(1,3.5)}^{(2,5)} \;=\; 30.875, \tag{10.101}$$

   and the output is

```
Numerical: 30.8796
Exact    : 30.875
```

3. To test the 2D polar implementation, we write `Test_trapezium_rule_2D_polar.cpp` to integrate $f(r, \theta) = r\,\cos(\theta)$ from $r_{\min} = 2$ to $r_{\max} = 3$, and $\theta_{\min} = \pi/4$ to $\theta_{\max} = \pi$, with $N_r = 10$ values of r and $N_\theta = 20$ values of θ, all equally spaced. The exact result is

$$\int_{\theta_{\min}}^{\theta_{\max}} \int_{r_{\min}}^{r_{\max}} f(r,\theta)\,r\,dr\,d\theta \;=\; \left[\frac{1}{3}\,r^3\,\sin(\theta)\right]_{(2,\,\pi/4)}^{(3,\,\pi)} \;=\; \frac{19}{3\sqrt{2}} \;\approx\; 4.4783429, \tag{10.102}$$

   and the output is

```
Numerical: -4.47188
Exact    : -4.47834
```

4. To test the 3D Cartesian implementation, we write `Test_trapezium_rule_3D_Cartesian.cpp` to integrate $f(x,y,z) = x^2 + y^2 + z^2$ from $x_{\rm min} = 1$ to $x_{\rm max} = 2$, and $y_{\rm min} = 3.5$ to $y_{\rm max} = 5$, and $z_{\rm min} = 2.5$ to $y_{\rm max} = 6$, with $N_x = 10$ values of x, $N_y = 20$ values of y and $N_z = 30$ values of z, all equally spaced. The exact result is

$$\int_{z_{\rm min}}^{z_{\rm max}} \int_{y_{\rm min}}^{y_{\rm max}} \int_{x_{\rm min}}^{x_{\rm max}} f(x,y,z)\, dx\, dy\, dz \;=\; \left[\frac{1}{3} \left(x^3\, y\, z + x\, y^3\, z + x\, y\, z^3 \right) \right]_{(1,\,3.5,\,2.5)}^{(2,\,5,\,6)} \;=\; 208.25, \qquad (10.103)$$

and the output is

```
Numerical: 208.279
Exact    : 208.25
```

5. To test the 3D polar implementation, we write `Test_trapezium_rule_3D_spherical.cpp` to integrate $f(r,\theta,\phi) = r\, \cos(\theta)\, \sin(\phi)$ from $r_{\rm min} = 2$ to $r_{\rm max} = 3$, and $\theta_{\rm min} = \pi/4$ to $\theta_{\rm max} = \pi$, and $\phi_{\rm min} = \pi/3$ to $\phi_{\rm max} = \pi/2$, with $N_r = 10$ values of r, $N_\theta = 20$ values of θ and $N_\phi = 30$ values of ϕ, all equally spaced. The exact result is

$$\int_{\phi_{\rm min}}^{\phi_{\rm max}} \int_{\theta_{\rm min}}^{\theta_{\rm max}} \int_{r_{\rm min}}^{r_{\rm max}} f(r,\theta,\phi)\, r^2\, \sin(\theta)\, dr\, d\theta\, d\phi \;=\; \left[\frac{1}{16}\, r^4\, \cos(2\theta)\, \cos(\phi) \right]_{(2,\,\pi/4,\,\pi/3)}^{(3,\,\pi,\,\pi/2)} \;=\; \frac{65}{32} \;=\; 2.03125, \qquad (10.104)$$

and the output is

```
Numerical: -2.03055
Exact    : -2.03125
```

In all cases, we get agreement within at least three significant figures even for a modest number of gridpoints. Execution time scales disproportionately with dimensionality, but our numerical simulation will only require 1D and 2D integration, somewhat mitigating this. Further, only one 2D integral is required per timestep, as opposed to larger numbers of 1D integrals. Thus, we can increase grid sizes for better integration accuracy without slowing the code down too much.

11 Numerical differentiation by spline fitting

Suppose that we have a function $f(x)$ whose closed analytical form is unknown, and which is evaluated at a set of points $\{x_i\}$ to give values $\{f_i\}$. Suppose that we wish to calculate the first and second derivatives of the function at each gridpoint, i.e. $\{f_i'\}$ and $\{f_i''\}$.

One method is to fit a polynomial to a small number of gridpoints surrounding x_i. This is attractive because a well-known result in linear algebra is that an $(n-1)^{\text{th}}$ order polynomial can always be found to pass *exactly* through n points. One might then naïvely assume that, since polynomials are straightforward to differentiate, we immediately obtain an exact expression for the derivatives at each gridpoint.

However, the logical fallacy is that although the fitted function passes exactly *through* each gridpoint, its behaviour *between* them is not accounted for. In fact, the higher-order polynomial that is fitted to, the more the function is prone to spurious oscillations between the gridpoints that may even become very large.

Now, the derivative of a function $f(x)$ is given by the rate of change with respect to x. As we will see in Chapter 12, this means that the derivative depends on function values either side of the gridpoint, and its accuracy may be improved by taking into account the effect of increasingly further gridpoints. In the limit of infinite accuracy, we can interpret this as saying that we need the function values all possible points in order to obtain an accurate value of the derivative at x_i. But if our fitting function is prone to spurious oscillations between gridpoints, clearly there will be infinitely many sub-gridpoint values of f that will spuriously influence the value we calculate.

In short, even though we can obtain a polynomial which *exactly* passes through a set of gridpoints, and even though we can differentiate this polynomial *exactly*, it doesn't necessarily give an exact value of the derivative at the gridpoints. Hence, polynomial fitting is not a valid method of numerically differentiating a function.

Proposition 11.1. *Let $x_0, x_1, x_2, y_0, y_1, y_2, a, b, c \in \mathbb{R}$. Suppose that we have three pairs of gridpoints (x_0, y_0), (x_1, y_1) and (x_2, y_2). The y-values may be exactly reproduced from the x-values, by the quadratic $f(x) = ax^2 + bx + c$, with coefficients*

$$a = \frac{1}{\det(A)}\Big[(x_1 - x_2)y_0 + (x_2 - x_0)y_1 + (x_0 - x_1)y_2\Big], \tag{11.1}$$

$$b = \frac{1}{\det(A)}\Big[(x_2^2 - x_1^2)y_0 + (x_0^2 - x_2^2)y_1 + (x_1^2 - x_0^2)y_2\Big], \tag{11.2}$$

$$c = \frac{1}{\det(A)}\Big[x_1x_2(x_1 - x_2)y_0 + x_0x_2(x_2 - x_0)y_1 + x_0x_1(x_0 - x_1)y_2\Big], \tag{11.3}$$

where $\det(A)$ *is the matrix determinant*

$$\det(A) = \det\begin{pmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{pmatrix} = \frac{1}{x_0^2(x_1 - x_2) + x_0(x_2^2 - x_1^2) + x_1x_2(x_1 - x_2)}. \tag{11.4}$$

That is, $y_0 = f(x_0) = ax_0^2 + bx_0 + c$ and so on.

Proof. We have the system of three simultaneous equations

$$f(x_0) = y_0 = ax_0^2 + bx_0 + c, \tag{11.5}$$

$$f(x_1) = y_1 = ax_1^2 + bx_1 + c, \tag{11.6}$$

$$f(x_2) = y_2 = ax_2^2 + bx_2 + c, \tag{11.7}$$

with coefficients a, b, c to be found. This can equivalently be written as a matrix equation

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \tag{11.8}$$

and so the matrix of coefficients is given by

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}. \tag{11.9}$$

Now, for a general 3×3 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \tag{11.10}$$

its inverse is

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} a_{22}a_{33} - a_{23}a_{32} & a_{13}a_{32} - a_{12}a_{33} & a_{12}a_{23} - a_{13}a_{22} \\ a_{23}a_{31} - a_{21}a_{33} & a_{11}a_{33} - a_{13}a_{31} & a_{13}a_{21} - a_{11}a_{23} \\ a_{21}a_{32} - a_{22}a_{31} & a_{12}a_{31} - a_{11}a_{32} & a_{11}a_{22} - a_{12}a_{21} \end{pmatrix}, \tag{11.11}$$

where

$$\det(A) = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(a_{23}a_{31} - a_{21}a_{33}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \tag{11.12}$$

is the determinant of A. Substituting this, we obtain the given result. □

Let us illustrate our arguments by fitting a quadratic to three gridpoints. First, we can produce a random set of three gridpoints by producing six random numbers, and plot a parabola through them. Figure 11.1 demonstrates this. Note that the parabola passes through all gridpoints within the resolution of the figure, demonstrating the validity of using a parabola to approximate a function between three gridpoints.

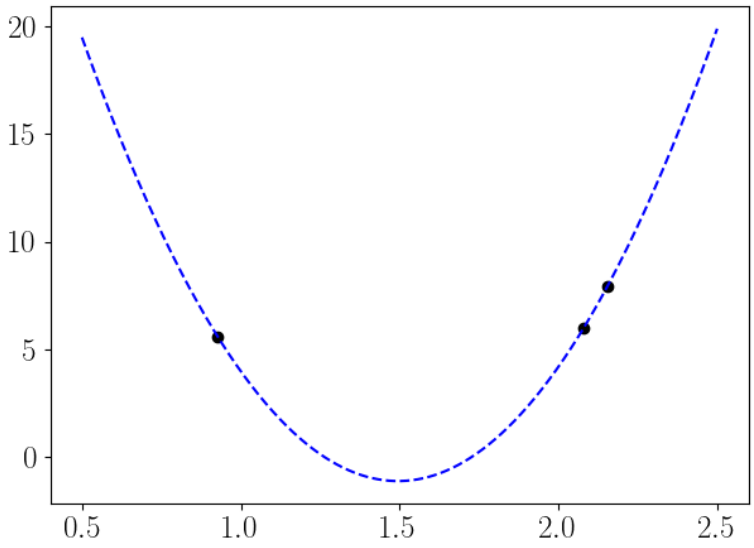

Figure 11.1: Three gridpoints chosen at random, and the quadratic that passes exactly through them. Produced by code: `Plot_Parabola_From_3_Points_Random.py`

However, let us now use the parabola to estimate the first and second derivative of a function at the central gridpoint (x_1, y_1). Suppose that the gridpoints are produced by a function $f(x) = \sin(x)$, a particularly "forgiving" choice because sinusoidal functions behave similarly to quadratics within small regions around a given gridpoint. Figure 11.2 shows the result given three randomly chosen gridpoints. The tangent line at (x_1, y_1) is also plotted, using both the gradient obtained from the parabola and the exact gradient of the known function. The first and second derivatives are

```
f'(x_1) exact          : 0.5069758539747738
f'(x_1) from parabola : 0.4883380482031725
f''(x_1) exact         : -0.8619602563265598
f''(x_1) from parabola: -0.9053190730467202
```

We see that the derivatives are not exact, but accurate in this example to around one significant figure. The two tangent lines are reasonably well aligned, but accuracy could be improved. This motivates us to develop a more accurate method, which we shall do by finite differencing in the following chapter. In so doing, we will obtain generalisable results that can be extended to arbitrary accuracy, greatly improving upon the most widely known finite difference methods.

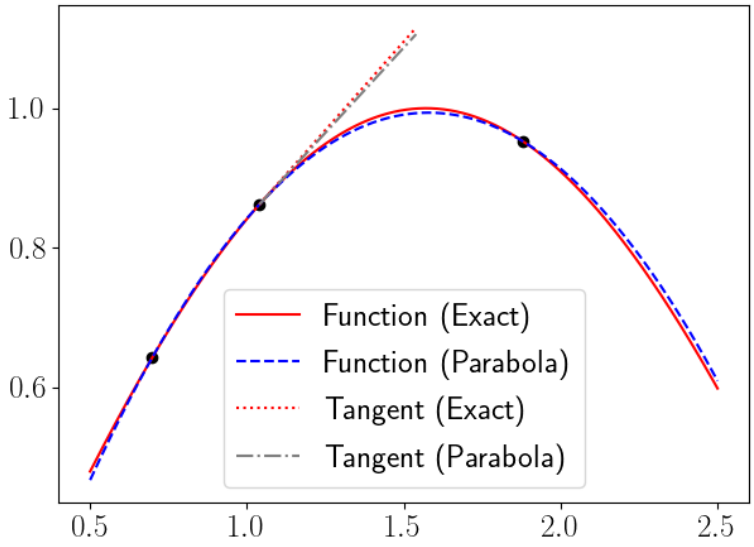

Figure 11.2: Three values of the function $f(x) = \sin(x)$ chosen at random, and the quadratic that passes exactly through them.
Produced by code: `Plot_Parabola_From_3_Points_and_Derivatives.py`

12 Numerical differentiation by finite differencing

In this chapter, we develop accurate finite difference schemes for the first and second derivative of a function evaluated at several equally spaced points. We only consider one-dimensional functions, but all results extend trivially to partial derivatives of multivariate functions if we hold the other variables constant.

12.1 Derivation from Taylor series

Let $x \in \mathbb{R}$ be a fixed point, let $h \in \mathbb{R} : |h| \ll 1$ be a small variable and let $f : \mathbb{R} \to \mathbb{R}$ be a "smooth" function, such that it can be differentiated infinitely many times (even if these eventually become zero). **Taylor's theorem** states that we can express $f(x+h)$ as

$$\begin{aligned} f(x+h) &= \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(x)\, h^k && (12.1) \\ &= f(x) + f'(x)\, h + \frac{1}{2} f''(x)\, h^2 + \frac{1}{6} f^{(3)}(x)\, h^3 + \frac{1}{24} f^{(4)}(x)\, h^4 + \cdots, && (12.2) \end{aligned}$$

which is a polynomial in h known as a **Taylor series**.[1] We can assume that the terms become smaller as k increases, which becomes increasingly accurate as $|h| \to 0$. Then, we can truncate the sum at some maximum index K and use the next term as an estimate of the error between our approximation and the true value of $f(x+h)$:

$$f(x+h) = \sum_{k=0}^{K} \frac{1}{k!} f^{(k)}(x)\, h^k + \frac{1}{(K+1)!} f^{(K+1)}(\xi)\, h^{K+1}, \tag{12.3}$$

where $\xi \in (x, x+h)$ is whichever value of x gives the largest value of $|f^{(K+1)}|$ on the interval $(x, x+h)$. Clearly this isn't known in general, and if we had a closed expression for $f^{(K+1)}(x)$ for which to check all values of ξ then we'd probably have enough knowledge of the original function not to need its Taylor series. But we can treat $f^{(K+1)}(\xi)$ as some unknown number; what is important is the scaling of this error term with h, so we typically write simply

$$f(x+h) = \sum_{k=0}^{K} \frac{1}{k!} f^{(k)}(x)\, h^k + \mathcal{O}(h^{K+1}). \tag{12.4}$$

Taylor series are most often used as a way to approximate a function in a small region around a gridpoint x, but we can also work the other way around and use the expression to estimate the derivatives of $f(x)$. The expressions we obtain are known as **finite difference approximations** because they are usually taught by first invoking the definition of the derivative,

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \tag{12.5}$$

and relaxing the limit to allow the expression to be evaluated for small but finite h. This yields perhaps the most famous finite difference expression,

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{12.6}$$

which we shall call the **order-h forward difference expression** for $f'(x)$. Forward difference expressions are so-called because they consider only the gridpoint x and those with larger coordinates, i.e. "going forward".

[1] Note that since x is fixed, the derivatives are evaluated at x and are simply real numbers.

These are vital for the innermost point of a finite domain, where there are no points with smaller values of x to use. We can also derive **backward difference expressions** using gridpoints such as $x - h$, which are vital for the outermost point of the domain, and **symmetric derivatives** using gridpoints either side, which are in general far more accurate and so should be used for intermediate points. Forward and backward expressions may be collectively referred to as **one-sided expressions**. We may also require **offset one-sided expressions** for points near the edges of the domain where not enough neighbours exist for a symmetric derivative of the required accuracy. Figure 12.1 visualises the need for various types of finite difference expression.

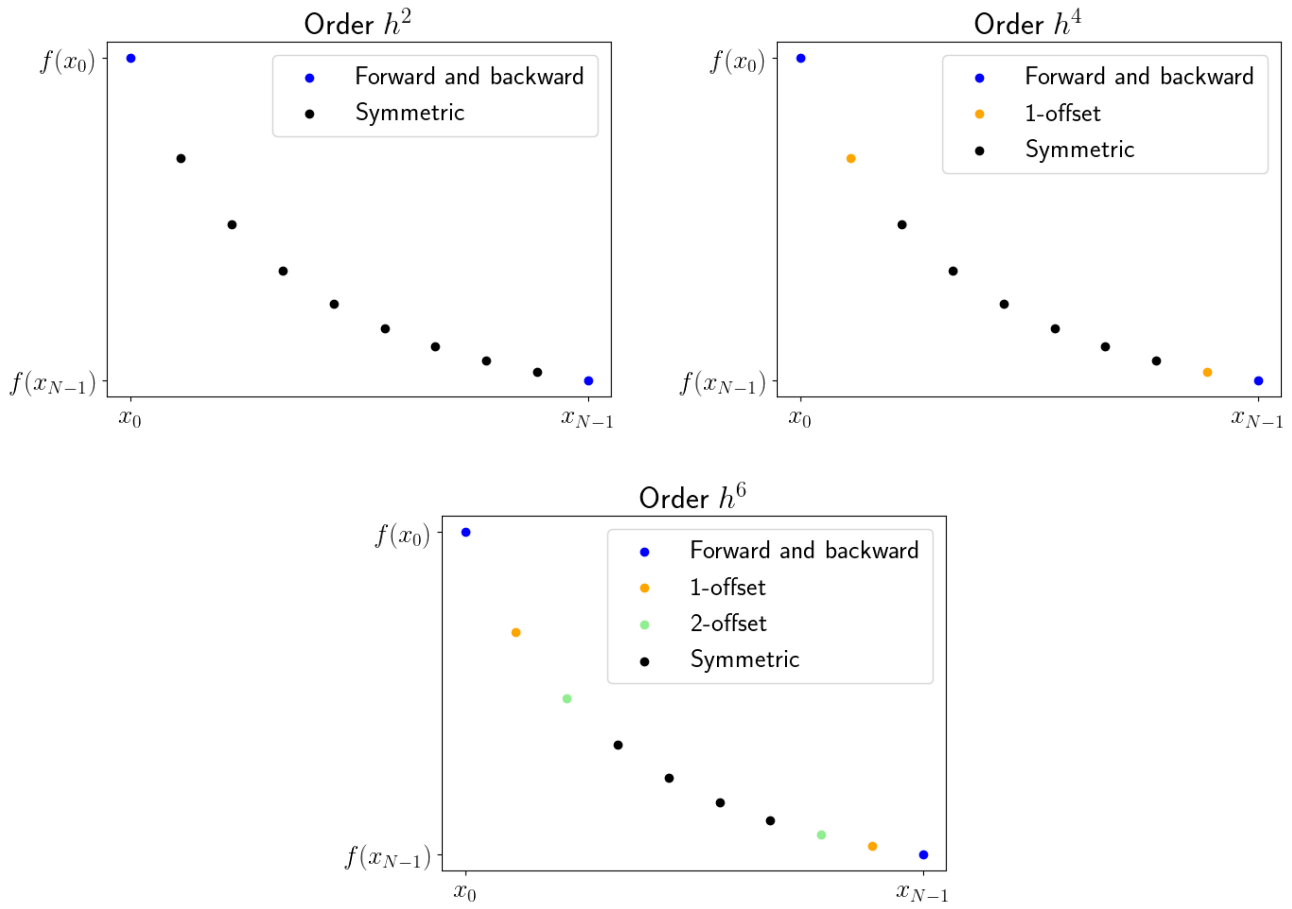

Figure 12.1: Visualisation of the use of one-sided, offset and symmetric finite difference expressions when calculating the first derivative of a function up to increasing accuracy. For higher derivatives, the gridpoint distribution is the same but the error estimation is different. The function plotted is $f(x) = \frac{1}{x^3}$, similar to the radial dependence that we expect to encounter in the simulation, with a small number $N = 10$ gridpoints to emphasise the end regions.
Produced by code: `Plot_Finite_Difference_Visualisation.py`

Let us evaluate the Taylor series at n gridpoints equally spaced either side of our reference point x:

$$f(x \pm nh) = \sum_{k=0}^{\infty} \frac{(\pm 1)^k n^k}{k!} f^{(k)}(x) h^k \tag{12.7}$$

$$= f(x) \pm n f'(x) h + \frac{n^2}{2} f''(x) h^2 \pm \frac{n^3}{6} f^{(3)}(x) h^3 + \frac{n^4}{24} f^{(4)}(x) h^3 \pm \cdots . \tag{12.8}$$

Adding and subtracting the Taylor series at various gridpoints yields expressions that can easily be rearranged for $f^{(k)}(x)$ involving as many gridpoints (hence as high accuracy) as required. The weights of each $f(x \pm nh)$ are chosen so as to cancel out some of the higher derivatives, improving accuracy; the more gridpoints we include, the more derivatives we can cancel out.

In principle, the same process can be applied for non-uniform gridpoints. However, the expression in Eq. (12.7) will in general require specific versions for each displaced gridpoint and so we will need to perform the analysis again specifically for that grid spacing, arriving at different finite-difference expressions. In this report, we consider only constant grid spacing.

12.2 First derivative

12.2.1 General expression

Proposition 12.1. *Suppose that we have a set of N_B gridpoints below x ("backward") and N_F gridpoints above x ("forward"), all equally spaced by h, for a total $N_B + N_R + 1$ gridpoints:*

$$\big\{\, x - N_B h\,,\, x-(N_B-1)h\,,\, \cdots\,,\, x-1\,,\, x\,,\, x+1\,,\, \cdots\,,\, x+(N_F-1)h\,,\, x+N_F h\,\big\}. \tag{12.9}$$

Then, the optimum finite difference expression for $f'(x)$ evaluated at x is

$$f'(x) \;=\; \frac{\displaystyle\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh) - \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\right) f(x)}{\left(\displaystyle\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n\right) h} + \mathcal{O}(h^{N_B+N_F}), \tag{12.10}$$

with $N_B + N_F$ coefficients a_n. The first $N_B + N_F - 1$ of these are obtained by first defining normalised versions with respect to $a_{N_B+N_F}$ so that $\tilde{a}_n = \frac{a_n}{a_{N_B+N_F}}$, and then solving the matrix equation

$$M\,\tilde{\mathbf{a}} \;=\; \mathbf{N}, \tag{12.11}$$

with matrix and vector elements

$$M_{i,j} \;=\; \begin{cases} (-N_B+j-1)^{i+1} & j < N_B, \\ (-N_B+j)^{i+1} & j \geq N_B, \end{cases} \tag{12.12}$$

$$(\tilde{\mathbf{a}})_i \;=\; \begin{cases} \tilde{a}_{-N_B+i-1} & i < N_B, \\ \tilde{a}_{-N_B+i} & i \geq N_B, \end{cases} \tag{12.13}$$

$$(\mathbf{N})_i \;=\; -N_F^{i+1}, \tag{12.14}$$

where $i, j \in \mathbb{N} : i, j \leq N_B + N_F$, to obtain $\tilde{a}_n$ for $n \in [1, N_B + N_F - 1]$. The choice of $a_{N_B+N_F}$ is free.

Proof. Let us take a linear combination of the Taylor series at these points except for x, with coefficients a_n to be found:

$$\sum_{n=-N_B}^{-1} a_n\, f(x+nh) + \sum_{n=1}^{N_F} a_n\, f(x+nh) \;=\; \sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh). \tag{12.15}$$

Use Eq. (12.7) for $f(x+nh)$:

$$\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh) \;=\; \sum_{k=0}^{\infty}\left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n^k\right)\frac{1}{k!}\, f^{(k)}(x)\, h^k. \tag{12.16}$$

The sums over k and n are independent of each other, so we can freely switch their order:

$$\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, f(x+nh) \;=\; \sum_{k=0}^{\infty}\left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, n^k\right)\frac{1}{k!}\, f^{(k)}(x)\, h^k \tag{12.17}$$

$$= \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\right) f(x) + \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, n\right) f'(x)\, h + \sum_{k=2}^{\infty}\left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, n^k\right)\frac{1}{k!}\, f^{(k)}(x)\, h^k, \tag{12.18}$$

where we explicitly stated the $k = 0$ and $k = 1$ terms for clarity. To improve the accuracy of our estimate of $f'(x)$, we should choose a_n so as to eliminate as many of the higher-order derivatives as possible. This means setting as many of the sums over $a_n\, n^k$ to zero as possible. We ignore $f^{(0)}(x) = f(x)$ because it is known exactly (so setting its sum to zero would remove one opportunity to cancel an unknown higher derivative); we ignore $f^{(1)}(x) = f'(x)$ because it is the target. Our $N_B + N_F$ coefficients a_n give $N_B + N_F - 1$ degrees of freedom; setting to zero the first $N_B + N_F - 1$ sums from $k = 2$ yields up to and including $k = N_B + N_F$. This results in

$$\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, f(x+nh) \;=\; \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\right) f(x) + \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, n\right) f'(x)\, h + \mathcal{O}(h^{N_B+N_F+1}), \tag{12.19}$$

which rearranges to the given expression. Let us now find the coefficients a_n. We have

$$\forall\, k \in \mathbb{N} : 2 \leq k \leq N_B + N_F, \qquad \sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} a_n\, n^k \;=\; 0, \tag{12.20}$$

a set of $N_B + N_F - 1$ equations for $N_B + N_F$ unknowns, which is underdetermined. To get around this, we can divide all of the coefficients by a_{N_F} such that $\tilde{a}_n = \frac{a_n}{a_{N_F}}$. This would cause issues if a_n were zero, but then we would effectively have a set of $N - 1$ gridpoints up to $x + (N-1)h$ and could proceed as normal for that case. Now we can separate $\tilde{a}_{N_F}$ from each sum,

$$\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F} \tilde{a}_n\, n^k \;=\; \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F-1} \tilde{a}_n\, n^k\right) + \tilde{a}_{N_F}\, N_F^k \;=\; \left(\sum_{\substack{n=-N_B\\n\neq 0}}^{N_F-1} \tilde{a}_n\, n^k\right) + N_F^k \;=\; \frac{0}{a_{N_F}} \;=\; 0, \tag{12.21}$$

and move it to the RHS:

$$\forall\, k \in \mathbb{N} : 2 \leq k \leq N_B + N_F, \qquad \sum_{\substack{n=-N_B\\n\neq 0}}^{N_F-1} \tilde{a}_n\, n^k \;=\; -N_F^k. \tag{12.22}$$

This is equivalent to the matrix equation

$$\begin{pmatrix} (-N_B)^2 & (-N_B+1)^2 & \cdots & (-2)^2 & (-1)^2 & 1^2 & 2^2 & \cdots & (N_F-1)^2 \\ (-N_B)^3 & (-N_B+1)^3 & \cdots & (-2)^3 & (-1)^3 & 1^3 & 2^3 & \cdots & (N_F-1)^3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (-N_B)^{N_B+N_F} & (-N_B+1)^{N_B+N_F} & \cdots & (-2)^{N_B+N_F} & (-1)^{N_B+N_F} & 1^{N_B+N_F} & 2^{N_B+N_F} & \cdots & (N_F-1)^{N_B+N_F} \end{pmatrix} \cdot \begin{pmatrix} \tilde{a}_{-N_B} \\ \tilde{a}_{-N_B+1} \\ \vdots \\ \tilde{a}_{-2} \\ \tilde{a}_{-1} \\ \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_{N_F-1} \end{pmatrix} = \begin{pmatrix} -N_F^2 \\ -N_F^3 \\ \vdots \\ -N_F^{N_B+N_F} \end{pmatrix}, \tag{12.23}$$

where the "central values" of each row of the matrix, and the "central values" of the vector of unknown coefficients, are shown explicitly to highlight the absence of terms related to $k = 0$. This can be written in compact form as given in the proposition. □

The matrix equation can be solved with standard methods such as Gaussian elimination. There is freedom when deciding on a value of a_N to return to the original coefficients. When encoding an automatic derivative calculation on a computer, it may be preferable to set $a_n = 1$. In this report, we seek mathematical expressions for the first derivative that we can quote, so we choose a_N to be the smallest value such that all a_n are integers.

The sums in the numerator in Eq. (12.10) may be combined when encoding onto a computer; if one seeks a mathematical expression, it is best to leave them separate.

Example 12.2 (1-offset order-h^5 forward difference expression)**.** *Suppose that we have the points*

$$\{\, x-h\,,\, x\,,\, x+h\,,\, x+2h\,,\, x+3h\,,\, x+4h\,\}, \tag{12.24}$$

and want to find the first derivative of a function at x. Our linear combination of Taylor series is

$$a_{-1}\, f(x-h) + a_1\, f(x+h) + a_2\, f(x+2h) + a_3\, f(x+3h) + a_4\, f(x+4h). \tag{12.25}$$

Continuing with the above process, we will arrive at the matrix equation

$$\begin{pmatrix} 1 & 1 & 4 & 9 \\ -1 & 1 & 8 & 27 \\ 1 & 1 & 16 & 81 \\ -1 & 1 & 32 & 243 \end{pmatrix} \begin{pmatrix} \tilde{a}_{-1} \\ \tilde{a}_1 \\ \tilde{a}_2 \\ \tilde{a}_3 \end{pmatrix} = \begin{pmatrix} -16 \\ -64 \\ -256 \\ -1024 \end{pmatrix}, \tag{12.26}$$

whose solution is

$$\begin{pmatrix} \tilde{a}_{-1} \\ \tilde{a}_1 \\ \tilde{a}_2 \\ \tilde{a}_3 \end{pmatrix} = \begin{pmatrix} 4 \\ -40 \\ 20 \\ \frac{-20}{3} \end{pmatrix}. \tag{12.27}$$

We choose $a_4 = 3$ so that a_3 becomes an integer, giving

$$\begin{pmatrix} a_{-1} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 12 \\ -120 \\ 60 \\ -20 \\ 3 \end{pmatrix}. \tag{12.28}$$

The sums in our expression for $f'(x)$ are then

$$\sum_{\substack{n=-N_B \\ n\neq 0}}^{N_F} a_n\, f(x+nh) = a_{-1}\, f(x-h) + a_1\, f(x+h) + a_2\, f(x+2h) + a_3\, f(x+3h) + a_4\, f(x+4h) \tag{12.29}$$

$$= 12\, f(x-h) - 120\, f(x+h) + 60\, f(x+2h) - 20\, f(x+3h) - 3\, f(x+4h), \tag{12.30}$$

$$\sum_{\substack{n=-N_B \\ n\neq 0}}^{N_F} a_n = a_{-1} + a_1 + a_2 + a_3 + a_4 = -65, \tag{12.31}$$

$$\sum_{\substack{n=-N_B \\ n\neq 0}}^{N_F} a_n\, n = a_{-1}\cdot(-1) + a_1\cdot(1) + a_2\cdot(2) + a_3\cdot(3) + a_4\cdot(4) = -60. \tag{12.32}$$

Absorbing the minus sign from the denominator, our expression for $f'(x)$ is

$$f'(x) = \frac{-3\, f(x+4h) + 20\, f(x+3h) - 60\, f(x+2h) + 120\, f(x+h) - 65\, f(x) - 12\, f(x-h)}{60h} + \mathcal{O}(h^5). \tag{12.33}$$

12.2.2 Forward expressions

Corollary 12.3. *Given $N+1$ forward-facing gridpoints $\{x, x+h, x+2h, ..., x+Nh\}$, the optimum finite difference expression for $f'(x)$ evaluated at x is*

$$f'(x) \;=\; \frac{\left(\sum_{n=1}^{N} a_n\, f(x+nh)\right) - \left(\sum_{n=1}^{N} a_n\right) f(x)}{\left(\sum_{n=1}^{N} a_n\, n\right) h} + \mathcal{O}(h^N), \tag{12.34}$$

with N coefficients a_n. The first $N-1$ of these are obtained by first defining normalised versions with respect to a_N so that $\tilde{a}_n = \frac{a_n}{a_N}$, and then solving the matrix equation

$$M\,\tilde{\mathbf{a}} \;=\; \mathbf{N}, \tag{12.35}$$

with matrix elements

$$M_{i,j} \;=\; j^{i+1}, \tag{12.36}$$
$$(\tilde{\mathbf{a}})_i \;=\; \tilde{a}_i, \tag{12.37}$$
$$(\mathbf{N})_i \;=\; -N^{i+1}, \tag{12.38}$$

to obtain $\tilde{a}_n$ for $n \in [1, N-1]$. The choice of a_N is free.

Proof. This follows immediately from Proposition 12.1 with $N_B = 0$ and $N_F = N$. □

The first few forward expressions for $f'(x)$ are

$$f'(x) \;=\; \frac{-f(x) + f(x+h)}{h} + \mathcal{O}(h) \tag{12.39}$$
$$\;=\; \frac{-3\,f(x) + 4\,f(x+h) - f(x+2h)}{2h} + \mathcal{O}(h^2) \tag{12.40}$$
$$\;=\; \frac{-11\,f(x) + 18\,f(x+h) - 9\,f(x+2h) + 2\,f(x+3h)}{6h} + \mathcal{O}(h^3) \tag{12.41}$$
$$\;=\; \frac{-25\,f(x) + 48\,f(x+h) - 36\,f(x+2h) + 16\,f(x+3h) - 3\,f(x+4h)}{12h} + \mathcal{O}(h^4). \tag{12.42}$$

Note that the order-h expression is the fiducial example we gave in Eq. (12.6).

12.2.3 Backward expressions

Corollary 12.4. *Given $N+1$ backward-facing gridpoints $\{x, x-h, x-2h, ..., x-Nh\}$, the optimum finite difference expression for $f'(x)$ evaluated at x is*

$$f'(x) = \frac{\left(\sum_{n=1}^{N} a_n\right) f(x) - \left(\sum_{n=1}^{N} a_n\, f(x+nh)\right)}{\left(\sum_{n=1}^{N} a_n\, n\right) h} + \mathcal{O}(h^N), \tag{12.43}$$

with N coefficients a_n equal to those found in Corollary 12.3.

Proof. This follows immediately from Corollary 12.3 with $h \to -h$. □

Although we have shown that the backward expression immediately follows from the forward expression, and hence from the general expression, we find that computer codes yield inaccurate results for this case. It may be safer to directly encode the expression in Corollary 12.3 and use this for backward expressions.

The first few backward expressions for $f'(x)$ are

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h) \tag{12.44}$$

$$= \frac{3\,f(x) - 4\,f(x-h) + f(x-2h)}{2h} + \mathcal{O}(h^2) \tag{12.45}$$

$$= \frac{11\,f(x) - 18\,f(x-h) + 9\,f(x-2h) - 2\,f(x-3h)}{6h} + \mathcal{O}(h^3) \tag{12.46}$$

$$= \frac{25\,f(x) - 48\,f(x-h) + 36\,f(x-2h) - 16\,f(x-3h) + 3\,f(x-4h)}{12h} + \mathcal{O}(h^4). \tag{12.47}$$

12.2.4 Symmetric expressions

Suppose now that we have N gridpoints symmetrically distributed either side of x. The general expression in Proposition 12.1 still holds, but we can derive a simpler expression which results in the solution of an $(N-1)\times(N-1)$ matrix equation instead of a $(2N-1)\times(2N-1)$ matrix equation.

In Eq. (12.7), the even-powered terms are always positive while the odd-powered terms have the same sign as $\pm nh$. Then, we can cancel alternate terms by adding or subtracting equally spaced gridpoints:

$$f(x+nh)+f(x-nh) \;=\; 2\sum_{\substack{k=0\\ k\text{ even}}}^{\infty}\frac{1}{k!}\,f^{(k)}(x)\,n^k\,h^k, \tag{12.48}$$

$$f(x+nh)-f(x-nh) \;=\; 2\sum_{\substack{k=1\\ k\text{ odd}}}^{\infty}\frac{1}{k!}\,f^{(k)}(x)\,n^k\,h^k. \tag{12.49}$$

Proposition 12.5. *Given $2N+1$ symmetrically distributed points $\{x-Nh, x-(N-1)h, ..., x-h, x, x+h, ..., x+(N-1)h, x+Nh\}$, the optimum finite difference expression for $f'(x)$ evaluated at x is*

$$f'(x) \;=\; \frac{\displaystyle\sum_{n=1}^{N} a_n\Big[f(x+nh)-f(x-nh)\Big]}{\displaystyle\Big(\sum_{n=1}^{N} a_n\,n\Big)\,2h}+\mathcal{O}(h^{2N}), \tag{12.50}$$

with N coefficients a_n. The first $N-1$ of these are obtained by first defining normalised versions with respect to a_N so that $\tilde{a}_n=\frac{a_n}{a_N}$, and then solving the matrix equation

$$M\,\tilde{\mathbf{a}} \;=\; \mathbf{N}, \tag{12.51}$$

with matrix elements

$$M_{i,j} \;=\; j^{2i+1}, \tag{12.52}$$

$$(\tilde{\mathbf{a}})_i \;=\; \tilde{a}_i, \tag{12.53}$$

$$(\mathbf{N})_i \;=\; -N^{2i+1}, \tag{12.54}$$

to obtain $\tilde{a}_n$ for $n\in[1,N-1]$. The choice of a_N is free.

Proof. Create a linear combination of the differences in Taylor series evaluated at symmetrically opposite gridpoints from x:

$$\sum_{n=1}^{N} a_n\Big[f(x+nh)-f(x-nh)\Big] \;=\; \sum_{n=1}^{N} a_n\Bigg[\sum_{\substack{k=1\\ k\text{ odd}}}^{\infty}\frac{2\,n^k}{k!}\,f^{(k)}(x)\,h^k\Bigg]. \tag{12.55}$$

Switching the order of the sums gives

$$\sum_{n=1}^{N} a_n\Big[f(x+nh)-f(x-nh)\Big] \;=\; \sum_{\substack{k=1\\ k\text{ odd}}}^{\infty}\Big(\sum_{n=1}^{N} a_n\,n^k\Big)\,\frac{2}{k!}\,f^{(k)}(x)\,h^k \tag{12.56}$$

$$=\; \Big(\sum_{n=1}^{N} a_n\,n\Big)\,2\,f'(x)\,h+\Big(\sum_{n=1}^{N} a_n\,n^3\Big)\,\frac{2}{6}\,f^{(3)}(x)\,h^3 + \Big(\sum_{n=1}^{N} a_n\,n^{2N-1}\Big)\,\frac{2}{(2N-1)!}\,f^{(2N-1)}(x)\,h^{2N-1}+\mathcal{O}(h^{2N+1}). \tag{12.57}$$

As in the proof of Proposition 12.1, our N coefficients give us $N-1$ degrees of freedom. We can set the sums multiplying $f(3)(x)$ up to $f^{(N)}(x)$ to zero. As before, we have $N-1$ equations for N unknowns, so divide the coefficients by a_N such that $\tilde{a}_n = \frac{a_n}{a_N}$ and separate the $n = N$ terms from the sums, again obtaining Eq. (12.21). This time, the index k begins at 3 and increments by 2 each time, so the system of equations in matrix form is

$$\begin{pmatrix} 1 & 8 & \cdots & (N-1)^3 \\ 1 & 32 & \cdots & (N-1)^5 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{2N-1} & \cdots & (N-1)^{2N-1} \end{pmatrix} \begin{pmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_N \end{pmatrix} = \begin{pmatrix} -N^3 \\ -N^5 \\ \vdots \\ -N^{2N-1} \end{pmatrix}. \tag{12.58}$$

Once a_n have been found, substitute them back into Eq. (12.56), which has now become

$$\sum_{n=1}^{N} a_n \Big[f(x+nh) - f(x-nh) \Big] = \left(\sum_{n=1}^{N} a_n \, n \right) 2 \, f'(x) \, h + \mathcal{O}(h^{2N+1}), \tag{12.59}$$

and rearrange for $f'(x)$. □

The first few symmetric expressions for $f'(x)$ are

$$f'(x) = \frac{f(x+h) - f(x-h)}{2\,h} + \mathcal{O}(h^2) \tag{12.60}$$

$$= \frac{-f(x+2h) + f(x-2h) + 8\left[f(x+h) - f(x-h)\right]}{12\,h} + \mathcal{O}(h^4) \tag{12.61}$$

$$= \frac{f(x+3h) - f(x-3h) - 9\left[f(x+2h) - f(x-2h)\right] + 45\left[f(x+h) - f(x-h)\right]}{60\,h} + \mathcal{O}(h^6). \tag{12.62}$$

12.2.5 Combining expressions

Below, we list systems of finite-difference expressions to use across an entire list of datapoints $f_n \equiv f(x_n)$ evaluated for N gridpoints $x_n = x_0 + nh$, with $n \in \mathbb{Z} : 0 \leq n < N$. That is, we use the same enumeration as computer codes and start from $n = 0$.

For a maximum error $\mathcal{O}(h)$, we only require forward and backward expressions:

$$f'_n = \frac{\text{(numerator)}}{h} + \mathcal{O}(h), \tag{12.63}$$

where

$$\text{(numerator)} = \begin{cases} f_{n+1} - f_n & n \leq N-2, \\ \\ f_{N-1} - f_{N-2} & n = N-1. \end{cases} \tag{12.64}$$

For a maximum error $\mathcal{O}(h^2)$ we use a symmetric expression with separate forward/backward expressions at the endpoints:

$$f'_n = \frac{\text{(numerator)}}{2\,h} + \mathcal{O}(h^2), \tag{12.65}$$

where

$$\text{(numerator)} = \begin{cases} -f_2 + 4\,f_1 - 3\,f_0 & n = 0, \\ \\ f_{n+1} - f_{n-1} & 1 \leq N \leq N-2, \\ \\ 3\,f_{N-1} - 4\,f_{N-2} + f_{N-3} & n = N-1. \end{cases} \tag{12.66}$$

For a maximum error $\mathcal{O}(h^4)$, we require offset expressions at $n = 1$ and $n = N-2$:

$$f'_n = \frac{\text{(numerator)}}{12\,h} + \mathcal{O}(h^4), \tag{12.67}$$

where

$$\text{(numerator)} = \begin{cases} -3\,f_4 + 16\,f_3 - 36\,f_2 + 48\,f_1 - 25\,f_0 & n = 0, \\ \\ f_4 - 6\,f_3 + 18\,f_2 - 10\,f_1 - 3\,f_0 & n = 1, \\ \\ -f_{n+2} + f_{n-2} + 8\,(f_{n+1} - f_{n-1}) & n \leq n \leq N-3, \\ \\ 3\,f_{N-1} + 10\,f_{N-2} - 18\,f_{N-3} + 6\,f_{N-4} - f_{N-5} & n = N-2, \\ \\ 25\,f_{N-1} - 48\,f_{N-2} + 36\,f_{N-3} - 16\,f_{N-4} + 3\,f_{N-5} & n = N-1. \end{cases} \tag{12.68}$$

For a maximum error $\mathcal{O}(h^6)$,

$$f'_n = \frac{\text{(numerator)}}{60\,h} + \mathcal{O}(h^6), \tag{12.69}$$

where

$$\text{(numerator)} = \begin{cases} -10\,f_6 + 72\,f_5 - 225\,f_4 + 400\,f_3 - 450\,f_2 + 360\,f_1 - 147\,f_0 & n = 0, \\ 2\,f_6 - 15\,f_5 + 50\,f_4 - 100\,f_3 + 150\,f_2 - 77\,f_1 - 10\,f_0 & n = 1, \\ -f_6 + 8\,f_5 - 30\,f_4 + 80\,f_3 - 35\,f_2 - 24\,f_1 + 2\,f_0 & n = 2, \\ +f_{N-7} - 8\,f_{N-6} + 30\,f_{N-5} - 80\,f_{N-4} + 35\,f_{N-3} + 24\,f_{N-2} - 2\,f_{N-1} & n = N-3, \\ f_{n+3} - f_{n-3} - 9\,(f_{n+2} - f_{n-2}) + 45\,(f_{n+1} - f_{n-1}) & 3 \leq n \leq N-4, \\ -2\,f_{N-7} + 15\,f_{N-6} - 50\,f_{N-5} + 100\,f_{N-4} - 150\,f_{N-3} + 77\,f_{N-2} + 10\,f_{N-1} & n = N-2, \\ 10\,f_{N-7} - 72\,f_{N-6} + 225\,f_{N-5} - 400\,f_{N-4} + 450\,f_{N-3} - 360\,f_{N-2} + 147\,f_{N-1} & n = N-1. \end{cases} \tag{12.70}$$

Accuracy at or near the endpoints, where symmetric expressions cannot be used, can be slightly lower than at the intermediate points. Higher-order expressions can be substituted for these points to mitigate this affect; for example, using the $\mathcal{O}(h^5)$ forward and backward expressions in the above for $n = 0$ and $n = N - 1$.

12.3 Second derivative

12.3.1 General expression

Proposition 12.6. *For the same set of gridpoints mentioned in Proposition 12.1, the optimum finite difference expression for $f''(x)$ evaluated at x is*

$$f''(x) \;=\; \frac{2\sum\limits_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh) - 2\left(\sum\limits_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\right) f(x)}{\left(\sum\limits_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n^2\right) h^2} + \mathcal{O}(h^{N_B+N_F-1}), \tag{12.71}$$

with $N_B + N_F$ coefficients a_n. The first $N_B + N_F - 1$ of these are obtained by first defining normalised versions with respect to $a_{N_B+N_F}$ so that $\tilde{a}_n = \frac{a_n}{a_{N_B+N_F}}$, and then solving the matrix equation

$$M\,\tilde{\mathbf{a}} \;=\; \mathbf{N}, \tag{12.72}$$

with matrix and vector elements

$$M_{i,j} \;=\; \begin{cases} -N_B + j - 1 & i = 1 \text{ and } j < N_B, \\ -N_B + j & i = 1 \text{ and } j \geq N_B, \\ (-N_B + j - 1)^{i+1} & i > 1 \text{ and } j < N_B, \\ (-N_B + j)^{i+1} & i > 1 \text{ and } j \geq N_B, \end{cases} \tag{12.73}$$

$$(\tilde{\mathbf{a}})_i \;=\; \begin{cases} \tilde{a}_{i-N_B-1} & i < N_B, \\ \tilde{a}_i & i \geq N_B, \end{cases} \tag{12.74}$$

$$(\mathbf{N})_i \;=\; \begin{cases} -N_F & i = 1, \\ -N_F^{i+1} & i > 1, \end{cases} \tag{12.75}$$

where $i, j \in \mathbb{N} : i, j \leq N_B + N_F$, to obtain $\tilde{a}_n$ for $n \in [1, N_B + N_F - 1]$. The choice of $a_{N_B+N_F}$ is free.

Proof. The proof is similar to that of Proposition 12.1. The same linear combination of Taylor series leads to

$$\begin{aligned} \sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh) \;=\; & \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\right) f(x) + \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n\right) f'(x)\, h + \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n^2\right) \frac{1}{2}\, f''(x)\, h^2 \\ & + \sum_{k=3}^{\infty} \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n^k\right) \frac{1}{k!}\, f^{(k)}(x)\, h^k, \end{aligned} \tag{12.76}$$

which is the same as Eq. (12.18) but with the $k = 2$ term also stated for clarity. Again, we have $N_B + N_F - 1$ degrees of freedom. We set to zero the sum for $k = 1$ (first derivative) and the $N_B + N_F - 2$ sums from $k = 3$ up to and including $k = N_B + N_F$. This results in

$$\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, f(x+nh) \;=\; \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\right) f(x) + \left(\sum_{\substack{n=-N_B\\ n\neq 0}}^{N_F} a_n\, n^2\right) \frac{1}{2}\, f''(x)\, h^2 + \mathcal{O}(h^{N_B+N_F+1}), \tag{12.77}$$

which rearranges to the given expression. Let us now find the coefficients a_n. We have

$$\forall\, k \in \mathbb{N} : (1 \leq k \leq N_B + N_F) \wedge (k \neq 2), \qquad \sum_{\substack{n=-N_B \\ n\neq 0}}^{N_F} a_n\, n^k \;=\; 0, \tag{12.78}$$

which as before is an undetermined system of $N_B + N_F - 1$ equations for $N_B + N_F$ coefficients. Define $\tilde{a}_n = \frac{a_n}{a_{N_F}}$, separate the last term and move it to the RHS:

$$\forall\, k \in \mathbb{N} : (1 \leq k \leq N_B + N_F) \wedge (k \neq 2), \qquad \sum_{\substack{n=-N_B \\ n\neq 0}}^{N_F-1} \tilde{a}_n\, n^k \;=\; -N_F^k. \tag{12.79}$$

This is equivalent to the matrix equation

$$\begin{pmatrix} -N_B & -N_B+1 & \cdots & -2 & -1 & 1 & 2 & \cdots & N_F-1 \\ (-N_B)^3 & (-N_B+1)^3 & \cdots & (-2)^3 & (-1)^3 & 1^3 & 2^3 & \cdots & (N_F-1)^3 \\ (-N_B)^4 & (-N_B+1)^4 & \cdots & (-2)^4 & (-1)^4 & 1^4 & 2^4 & \cdots & (N_F-1)^4 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (-N_B)^{N_B+N_F} & (-N_B+1)^{N_B+N_F} & \cdots & (-2)^{N_B+N_F} & (-1)^{N_B+N_F} & 1^{N_B+N_F} & 2^{N_B+N_F} & \cdots & (N_F-1)^{N_B+N_F} \end{pmatrix}$$

$$\cdot \begin{pmatrix} \tilde{a}_{-N_B} \\ \tilde{a}_{-N_B+1} \\ \vdots \\ \tilde{a}_{-2} \\ \tilde{a}_{-1} \\ \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_{N_F-1} \end{pmatrix} = \begin{pmatrix} -N_F \\ -N_F^3 \\ -N_F^4 \\ \vdots \\ -N_F^{N_B+N_F} \end{pmatrix}. \tag{12.80}$$

This can be written in compact form as given in the proposition. □

In testing, this provides accurate derivatives for the cases $N_B = 0$ and $N_B = N_F$, that is, the forward and symmetric derivatives. Since the backward derivatives immediately follow from the forward derivatives, we can obtain those too. However, the expression appears to break when calculating intermediate expressions. Due to lack of necessity, this issue was not pursued further. The reader may find applications of the above expression for the second derivative in their own work, but care should be taken with intermediate points. As discussed above, one could use the forward and backward methods for all intermediate points as a somewhat crude workaround to achieve the desired accuracy.

12.3.2 Forward and backward expressions

The first few forward expressions for $f''(x)$ are

$$f''(x) = \frac{f(x+2h) - 2\,f(x+h) + f(x)}{h^2} + \mathcal{O}(h) \tag{12.81}$$

$$= \frac{-f(x+3h) + 4\,f(x+2h) - 5\,f(x+h) + 2\,f(x)}{h^2} + \mathcal{O}(h^2) \tag{12.82}$$

$$= \frac{11\,f(x+4h) - 56\,f(x+3h) + 114\,f(x+2h) - 104\,f(x+h) + 35\,f(x)}{12\,h^2} + \mathcal{O}(h^3) \tag{12.83}$$

$$= \frac{-10\,f(x+5h) + 61\,f(x+4h) - 156\,f(x+3h) + 214\,f(x+2h) - 154\,f(x+h) + 45\,f(x)}{12\,h^2} + \mathcal{O}(h^4) \tag{12.84}$$

$$= \frac{1}{180\,h^2}\Big[137\,f(x+6h) - 972\,f(x+5h) + 2970\,f(x+4h) - 5080\,f(x+3h) + 5265\,f(x+2h) - 3132\,f(x+h) + 812\,f(x)\Big] + \mathcal{O}(h^5), \tag{12.85}$$

$$= \frac{1}{180\,h^2}\Big[-126\,f(x+7h) + 1019\,f(x+6h) - 3618\,f(x+5h) + 7380\,f(x+4h) - 9490\,f(x+3h) + 7911\,f(x+2h) - 4014\,f(x+h) + 938\,f(x)\Big] + \mathcal{O}(h^6). \tag{12.86}$$

The backward expressions for $f''(x)$ are the same as the forward expressions, but with $f(x+nh)$ replaced by $f(x-nh)$. Note that the order-h forward expression is equal to the order-h symmetric expression for $f''(x+nh)$, and hence the order-h backward expression is equal to the order-h symmetric expression for $f''(x-nh)$, so these in particular may not be especially accurate. In this case, we recommend using the order-h^2 forward/backward methods instead.

12.3.3 Symmetric expressions

As we found for the first derivative, the general expression in Proposition 12.6 holds for symmetric derivatives, but we can obtain a simpler expression with a fresh derivation. In so doing, we will find that the error term is improved by a factor of h.

Proposition 12.7. *Given $2N+1$ symmetrically distributed points $\{x-Nh, x-(N-1)h, ..., x-h, x, x+h, ..., x+(N-1)h, x+Nh\}$, the optimum finite difference expression for $f''(x)$ evaluated at x is*

$$f''(x) \;=\; \frac{\displaystyle\sum_{n=1}^{N} a_n \Big[f(x+nh)+f(x-nh)\Big] - 2\Big(\sum_{n=1}^{N} a_n\Big) f(x)}{\displaystyle\Big(\sum_{n=1}^{N} a_n\, n^2\Big) h} + \mathcal{O}(h^{2N}), \tag{12.87}$$

with N coefficients a_n. The first $N-1$ of these are obtained by first defining normalised versions with respect to a_N so that $\tilde{a}_n = \frac{a_n}{a_N}$, and then solving the matrix equation

$$M\,\tilde{\mathbf{a}} \;=\; \mathbf{N}, \tag{12.88}$$

with matrix elements

$$M_{i,j} \;=\; j^{2i+2}, \tag{12.89}$$
$$(\tilde{\mathbf{a}})_i \;=\; \tilde{a}_i, \tag{12.90}$$
$$(\mathbf{N})_i \;=\; -N^{2i+2}, \tag{12.91}$$

to obtain $\tilde{a}_n$ for $n \in [1, N-1]$. The choice of a_N is free.

Proof. Recalling Eq. (12.48), we can eliminate all of the odd derivatives by restricting $a_{-n} = a_n$. Doing this uses up half of the available degrees of freedom, but it means we no longer need to explicitly set the sum for the first derivative to zero. We have the linear combination

$$\sum_{n=1}^{N} a_n\big[f(x+nh)+f(x-nh)\big] \;=\; \sum_{n=1}^{N} a_n\Bigg(2\sum_{\substack{k=0\\ k\text{ even}}}^{\infty} \frac{1}{k!}\, f^{(k)}(x)\, n^k\, h^k\Bigg) \tag{12.92}$$

$$=\; \sum_{\substack{k=0\\ k\text{ even}}}^{\infty} \Bigg(\sum_{n=1}^{N} a_n\, n^k\Bigg)\, 2\,\frac{1}{k!}\, f^{(k)}(x)\, h^k \tag{12.93}$$

$$=\; \Bigg(\sum_{n=1}^{N} a_n\Bigg)\, 2\, f(x) + \Bigg(\sum_{n=1}^{N} a_n\, n^2\Bigg)\, f''(x)\, h^2 + \sum_{\substack{k=4\\ k\text{ even}}}^{\infty} \Bigg(\sum_{n=1}^{N} a_n\, n^k\Bigg)\, 2\,\frac{1}{k!}\, f^{(k)}(x)\, h^k. \tag{12.94}$$

Our N coefficients give us $N-1$ degrees of freedom, so let us set the first $N-1$ sums over $a_n\, n^k$ to zero, from $k=4$ up to and including $k = 2N+2$. This leaves

$$\sum_{n=1}^{N} a_n\big[f(x+nh)+f(x-nh)\big] \;=\; \Bigg(\sum_{n=1}^{N} a_n\Bigg)\, 2\, f(x) + \Bigg(\sum_{n=1}^{N} a_n\, n^2\Bigg)\, f''(x)\, h^2 + \mathcal{O}(h^{2N+2}), \tag{12.95}$$

which rearranges to the given expression. Let us now find the coefficients a_n. We have

$$\forall\, k \in \mathbb{N} : (k \text{ even}) \wedge (4 \le k \le 2N), \qquad \sum_{n=1}^{N} a_n\, n^k \;=\; 0, \tag{12.96}$$

a system of $N-1$ equations for N unknowns. Define $\tilde{a}_n \equiv \frac{a_n}{a_N}$ and separate the $\tilde{a}_n$ term to give

$$\forall\, k \in \mathbb{N} : (k \text{ even}) \wedge (4 \le k \le 2N), \qquad \sum_{n=1}^{N} a_n\, n^k \;=\; 0, \tag{12.97}$$

which is equivalent to the matrix equation

$$\begin{pmatrix} 1^4 & 2^4 & \cdots & N^4 \\ 1^6 & 2^6 & \cdots & N^6 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{2N} & 2^{2N} & \cdots & N^{2N} \end{pmatrix} \begin{pmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_3 \end{pmatrix} \;=\; \begin{pmatrix} -N^4 \\ -N^6 \\ \vdots \\ -N^{2N} \end{pmatrix}. \tag{12.98}$$

This can be written in compact form as given in the proposition. □

The first few symmetric expressions for $f''(x)$ are

$$f''(x) \;=\; \frac{f(x+h) - 2\,f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2) \tag{12.99}$$

$$\;=\; \frac{-\big[f(x+2h) + f(x-2h)\big] + 16\,\big[f(x+h) + f(x-h)\big] - 30\,f(x)}{12\,h^2} + \mathcal{O}(h^4) \tag{12.100}$$

$$\;=\; \frac{2\,\big[f(x+3h) + f(x-3h)\big] - 27\,\big[f(x+2h) + f(x-2h)\big] + 270\,\big[f(x+h) - f(x-h)\big] - 490\,f(x)}{180\,h^2} + \mathcal{O}(h^6). \tag{12.101}$$

12.3.4 Combining expressions

Below, we list systems of finite-difference expressions to use across an entire list of datapoints $f_n \equiv f(x_n)$ evaluated for N gridpoints $x_n = x_0 + nh$, with $n \in \mathbb{Z} : 0 \leq n < N$. That is, we use the same enumeration as computer codes and start from $n = 0$. Recall that we are unable to find expressions at intermediate gridpoints, so use the forward and backward expressions at these points.

For maximum error $\mathcal{O}(h^2)$,

$$f''_n = \frac{\text{(numerator)}}{h^2} + \mathcal{O}(h^2), \tag{12.102}$$

where

$$\text{(numerator)} = \begin{cases} -f_3 + 4\,f_2 - 5\,f_1 + 2\,f_0 & n = 0, \\ f_{n+1} - 2\,f_n + f_{n-1} & 1 \leq n \leq N-2, \\ -f_{N-4} + 4\,f_{N-3} - 5\,f_{N-2} + 2\,f_{N-2} + 2\,f_{N-1} & n = N-1. \end{cases} \tag{12.103}$$

For maximum error $\mathcal{O}(h^4)$,

$$f''_n = \frac{\text{(numerator)}}{12\,h^2} + \mathcal{O}(h^4), \tag{12.104}$$

where

$$\text{(numerator)} = \begin{cases} -10\,f_{n+5} + 61\,f_{n+4} - 156\,f_{n+3} + 214\,f_{n+2} - 154\,f_{n+1} + 45\,f_n & 0 \leq n \leq 1, \\ -(f_{n+2} + f_{n-2}) + 16\,(f_{n+1} + f_{n-1}) - 30\,f_n & 2 \leq n \leq N-3, \\ -10\,f_{n-5} + 61\,f_{n-4} - 156\,f_{n-3} + 214\,f_{n-2} - 154\,f_{n-1} + 45\,f_n & N-2 \leq n \leq N-1. \end{cases} \tag{12.105}$$

For maximum error $\mathcal{O}(h^6)$,

$$f''_n = \frac{\text{(numerator)}}{180\,h^2} + \mathcal{O}(h^6), \tag{12.106}$$

where

$$\text{(numerator)} = \begin{cases} -126\,f_{n+7} + 1019\,f_{n+6} - 3618\,f_{n+5} + 7380\,f_{n+4} \\ \qquad -9490\,f_{n+3} + 7911\,f_{n+2} - 4014\,f_{n+1} + 938\,f_n & 0 \leq n \leq 2, \\ 2\,(f_{n+3} + f_{n-3}) - 27\,(f_{n+2} + f_{n-2}) + 270\,(f_{n+1} + f_{n-1}) - 490\,f_n & 3 \leq n \leq N-4, \\ -126\,f_{n-7} + 1019\,f_{n-6} - 3618\,f_{n-5} + 7380\,f_{n-4} \\ \qquad -9490\,f_{n-3} + 7911\,f_{n-2} - 4014\,f_{n-1} + 938\,f_n & N-3 \leq n \leq N-1. \end{cases} \tag{12.107}$$

13 Chebyshev polynomials

The reader solely interested in functions used within the final code may skip this and the following chapter.

We will now discuss the Chebyshev polynomials and their application for approximating 1D functions and their derivatives. The aim is to use these functions to calculate the radial derivatives of our vector fields in the time-evolution code.

We will ultimately conclude after side-by-side testing in Chapter 24 that our finite-difference methods from Chapter 12 prove more accurate than Chebyshev decomposition in our code (which perhaps should not be surprising given our brief discussion of spline fitting in Chapter 11). However, the results derived in these chapters represent original work that may well find applications in other projects, so we include them in full.

13.1 Chebyshev polynomials on $[-1, 1]$

13.1.1 Definition and examples

Definition 13.1. *The* **Chebyshev polynomials** *are two sequences of polynomials in $x \equiv \cos(\theta)$, defined on the interval $x \in [-1, 1]$, or the interval $\theta \in [0, \pi]$ because $\cos(\theta)$ is one-to-one. There exist* **Chebyshev polynomials of the first kind** *$T_n[\cos(\theta)] = T_n(x)$ and* **Chebyshev polynomials of the second kind** *$U_n[\cos(\theta)] = U_n(x)$, respectively defined in terms of θ as*

$$T_n[\cos(\theta)] = \cos(n\theta), \tag{13.1}$$

$$U_n[\cos(\theta)] = \frac{\sin[(n+1)\theta]}{\sin(\theta)}, \tag{13.2}$$

and in terms of x as

$$T_n(x) = \cos\left[n \cos^{-1}(x)\right], \tag{13.3}$$

$$U_n(x) = \frac{\sin\left[(n+1)\cos^{-1}(x)\right]}{\sin\left[\cos^{-1}(x)\right]} = \frac{\sin\left[(n+1)\cos^{-1}(x)\right]}{\sqrt{1-x^2}}. \tag{13.4}$$

While it is not immediately apparent that these are polynomials in $x = \cos(\theta)$, we prove this fact in §13.3. Note that we will mostly be using Chebyshev polynomials of the first kind during this text, and for brevity we will refer to these simply as "Chebyshev polynomials". Let us also define $T_n = U_n = 0$ for $n < 0$, which will prove useful when considering derivatives. The first few Chebyshev polynomials of the first kind are

$$T_0(x) = 1, \tag{13.5}$$

$$T_1(x) = x, \tag{13.6}$$

$$T_2(x) = 2x^2 - 1, \tag{13.7}$$

$$T_3(x) = 4x^3 - 3x, \tag{13.8}$$

$$T_4(x) = 8x^4 - 8x^2 + 1, \tag{13.9}$$

and the first few Chebyshev polynomials of the second kind are

$$U_0(x) = 1, \tag{13.10}$$

$$U_1(x) = 2x, \tag{13.11}$$

$$U_2(x) = 4x^2 - 1, \tag{13.12}$$

$$U_3(x) = 8x^3 - 4x, \tag{13.13}$$

$$U_4(x) = 16x^4 - 12x^2 + 1. \tag{13.14}$$

13.1.2 Properties and relations

The T_n and U_n both follow the same recursion relation:

$$T_n(x) \;=\; 2x\,T_{n-1}(x) - T_{n-2}(x), \tag{13.15}$$
$$U_n(x) \;=\; 2x\,U_{n-1}(x) - U_{n-2}(x). \tag{13.16}$$

The T_n and U_n of even/odd order are even/odd functions:

$$T_n(-x) \;=\; (-1)^n\,T_n(x), \tag{13.17}$$
$$U_n(-x) \;=\; (-1)^n\,U_n(x). \tag{13.18}$$

Proposition 13.2 (Values at the endpoints)**.** *$\forall\, n \in \mathbb{N}_0$, we have*

$$T_n(1) \;=\; T_n\big[\cos(2n\pi)\big] \;=\; 1, \tag{13.19}$$
$$U_n(1) \;=\; U_n\big[\cos(2n\pi)\big] \;=\; n+1, \tag{13.20}$$

and

$$T_n(-1) \;=\; T_n\big[\cos((2n+1)\pi)\big] \;=\; (-1)^n, \tag{13.21}$$
$$U_n(-1) \;=\; U_n\big[\cos((2n+1)\pi)\big] \;=\; (-1)^n\,(n+1), \tag{13.22}$$

and

$$T_n(0) \;=\; T_n\left[\cos\left(\frac{\pi}{2} + 2n\pi\right)\right] \;=\; \begin{cases} 0 & n \text{ odd}, \\ -1 & n \equiv 0 \,(\mathrm{mod}\, 4), \\ 1 & n \equiv 2 \,(\mathrm{mod}\, 4). \end{cases} \tag{13.23}$$

Proof. Since $x = \cos(\theta)$ is one-to-one on $\theta \in (0, \pi)$, then $\cos(\theta) = 1$ iff $\theta = 0$. Then,

$$T_n(1) \;=\; T_n\big[\cos(0)\big] \;=\; \cos(n \cdot 0) \;=\; \cos(0) = 1. \tag{13.24}$$

For U_n, we have a $\frac{0}{0}$ indeterminate form and must invoke L'Hôpital's rule:

$$\lim_{x\to 1} U_n(x) \;=\; \lim_{\theta\to 0} \frac{\sin\big[(n+1)\theta\big]}{\sin(\theta)} \;=\; \lim_{\theta\to 0} \frac{(n+1)\cos\big[(n+1)\theta\big)}{\cos(\theta)} \tag{13.25}$$
$$=\; (n+1)\lim_{\theta\to 0} \frac{\cos\big[(n+1)\theta\big]}{\cos(\theta)} \;=\; (n+1)\cdot\frac{1}{1} \;=\; n+1. \tag{13.26}$$

Similarly, $\cos(\theta) = -1$ iff $\theta = \pi$. Then,

$$T_n(-1) \;=\; T_n\big[\cos(\pi)\big] \;=\; \cos(n\pi) \;=\; (-1)^n, \tag{13.27}$$

and

$$\lim_{x\to -1} U_n(x) \;=\; (n+1)\lim_{\theta\to \pi} \frac{\cos\big[(n+1)\theta\big]}{\cos(\theta)} \;=\; (n+1)\cdot\frac{(-1)^n}{-1} \;=\; -(n+1)\,(-1)^n \;=\; (n+1)\,(-1)^{n+1}. \tag{13.28}$$

For $T_n(0)$, make use of the recursion relation. Clearly $T_0(0) = 1$ and $T_1(0) = 0$. Then, for $n > 2$, $T_n(0) = 2 \cdot 0 \cdot T_{n-1}(0) - T_{n-2}(0) = -T_{n-2}(0)$, so $T_2(0) = -T_0(0) = -1$ and so on. □

Proposition 13.3 (First derivatives)**.**

$$\frac{d}{dx}\,T_n(x) \;=\; n\,U_{n-1}(x) \tag{13.29}$$

Proof. Differentiate the definition of T_n, Eq. (13.3), using the chain rule:

$$\frac{d}{dx}\,T_n(x) \;=\; \frac{d}{dx}\,\cos\left[n\,\cos^{-1}(x)\right] \;=\; -\sin\left[n\,\cos^{-1}(x)\right]\frac{d}{dx}\,n\,\cos^{-1}(x) \tag{13.30}$$

$$\;=\; -\sin\left[n\,\cos^{-1}(x)\right] n\,\cdot -\frac{1}{\sqrt{1-x^2}} \;=\; n\,\frac{\sin\left[n\,\cos^{-1}(x)\right]}{\sqrt{1-x^2}} \;=\; n\,U_{n-1}(x), \tag{13.31}$$

where we used that $\frac{d}{dx}\,\cos^{-1}(x) = -\frac{1}{\sqrt{1-x^2}}$ and recognised the definition of U_{n-1}, Eq. (13.4). □

The T_n obey the following orthogonality relation:

$$\int_{-1}^{1} T_n(x)\,T_m(x)\,\frac{1}{\sqrt{1-x^2}}\,dx \;=\; \int_0^{\pi} T_n\big[\cos(\theta)\big]\,T_m\big[\cos(\theta)\big]\,d\theta \;=\; \begin{cases} \pi & n=m=0, \\ \frac{\pi}{2} & n=m \text{ and } n\neq 0, \\ 0 & n\neq m, \end{cases} \;=\; \begin{cases} \pi\,\delta_{n,m} & n=0, \\ \frac{\pi}{2}\,\delta_{n,m} & n\neq 0. \end{cases} \tag{13.32}$$

A useful expression is (e.g. Bateman, 1953, §10.11, Eq. (3))

$$U_{n+1}(x) \;=\; x\,U_n(x) + T_{n+1}(x). \tag{13.33}$$

Proposition 13.4. $T_n(x) \;=\; \frac{1}{2}\left[U_n(x) - U_{n-2}(x)\right]$

Proof. By the recursion relation Eq. (13.16), $2x\,U_{n-1} = U_n + U_{n-2}$ and so $x\,U_n = \frac{1}{2}(U_{n+1} + U_{n-1})$. Then, Eq. (13.33) gives

$$T_{n+1}(x) \;=\; U_{n+1}(x) - x\,U_n(x) \;=\; U_{n+1}(x) - \frac{1}{2}\Big(U_{n+1}(x) + U_{n-1}(x)\Big) \;=\; \frac{1}{2}\,U_{n+1}(x) - \frac{1}{2}\,U_{n-1}(x), \tag{13.34}$$

which is the given result if we relabel $n+1 \to n$. □

Proposition 13.5. *The Chebyshev polynomials of the second kind can be written in terms of those of the first kind by*

$$U_n \;=\; 2\sum_{\substack{k=0 \\ k+n \text{ even}}}^{n} h_k\,T_k \;=\; 2\sum_{\substack{k=n\,(\mathrm{mod}\,2) \\ k+=2}}^{n} h_k\,T_k, \tag{13.35}$$

where we define

$$h_n \;=\; 1 - \frac{1}{2}\,\delta_{n,0} \;=\; \begin{cases} \frac{1}{2} & n=0, \\ 1 & n\neq 0, \end{cases} \tag{13.36}$$

for notational convenience, and where the second summation gives a faster way to encode the result without having to check whether $k+n$ is even at each step. We borrow the coding notation += to mean "increment by 2 in the summation".

Verification. `Check mathematical expressions/20230624_Chebyshev_polynomial_second_kind_in_terms_of_first_kind.cpp` □

Proof. We prove the relation by induction. Due to the modulo 2, we must consider the first two base cases:

$$U_0 \;=\; = \;2 \sum_{\substack{k=0 \,(\mathrm{mod}\, 2) \\ k+=2}}^{0} h_k\, T_k \;=\; 2\left(h_0\, T_0\right) \;=\; 2 \cdot \frac{1}{2} \cdot 1 \;=\; 1, \tag{13.37}$$

$$U_1 \;=\; 2 \sum_{\substack{k=1 \,(\mathrm{mod}\, 2) \\ k+=2}}^{1} h_k\, T_k \;=\; 2\big(h_1\, T_1\big) \;=\; 2 \cdot 1 \cdot x \;=\; 2x. \tag{13.38}$$

Assume that the relation holds for n. Then, for $n+2$,

$$U_{n+2} \;=\; 2 \sum_{\substack{k=n+2 \,(\mathrm{mod}\, 2) \\ k+=2}}^{n+2} h_k\, T_k \;=\; 2 \sum_{\substack{k=n \,(\mathrm{mod}\, 2) \\ k+=2}}^{n+2} h_k\, T_k \;=\; 2\, h_{n+2}\, T_{n+2} + 2 \sum_{\substack{k=n \,(\mathrm{mod}\, 2) \\ k+=2}}^{n} h_k\, T_k. \tag{13.39}$$

Since $n+2 \neq 0$, we have $h_{n+2} = 1$. We recognise the second term as U_n. Then, we have

$$U_{n+2} \;=\; 2\, T_{n+2} + U_n, \tag{13.40}$$

which is Proposition 13.4. We have shown that the case n implies the case $n+2$. Coupled with our two consecutive base cases $n = 0$ and $n = 1$, this completes the proof. □

Corollary 13.6 (First derivatives)**.** *$\forall\, n \in \mathbb{N}_0$, we have*

$$\frac{dT_n}{dx} \;=\; n\, U_{n-1}(x) \;=\; \sum_{\substack{k=0 \\ k+n \text{ odd}}}^{n-1} 2n\, h_k\, T_k. \tag{13.41}$$

Had we not defined $U_{-1} = 0$, we would need a separate case to state that $\frac{dT_0}{dx} = 0$.

We can freely switch the order of a double sum. Let $i, j \in \mathbb{N}$. The sum of a quantity $a_{i,j}$ over both indices, with no restrictions on the indices chosen, is independent of the order in which the sums are taken:

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} a_{i,j} \;=\; \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} a_{i,j}. \tag{13.42}$$

If there are conditions on the indices, we can rewrite these as multiplication factors that are unity if the condition is met and zero otherwise. For example,

$$\sum_{\substack{i=1 \\ i \text{ even}}}^{\infty} a_{i,j} \;=\; \sum_{i=1}^{\infty} a_{i,j}\, g(i), \tag{13.43}$$

where we define

$$g(i) \;=\; \begin{cases} 1 & i \text{ even}, \\ 0 & i \text{ odd}. \end{cases} \tag{13.44}$$

Then, it is easy to switch the order of summation between two indices with restrictions. Once the order of summation has been switched, the conditions within g can be relegated to labels on the summations. Typically, it is useful to convert conditions on one index to conditions on the other.

Proposition 13.7 (Second derivatives). *$\forall\, n \in \mathbb{N}_0$, we have*

$$\frac{d^2T_n}{dx^2} \;=\; n(n+1)\,\frac{T_n}{1-x^2} + n\,\frac{U_n}{1-x^2} \;=\; \sum_{\substack{k=0\\ k+n \text{ even}}}^{n-2} \sum_{\substack{\ell=k+1\\ \ell+n \text{ odd}}}^{n-1} 4n\,h_k\,\ell\,T_k. \tag{13.45}$$

For the first expression, the values at the endpoints can be hard-coded to avoid division by zero as

$$\left.\frac{d^2T_n}{dx^2}\right|_{1} \;=\; \frac{n^4-n^2}{3}, \tag{13.46}$$

$$\left.\frac{d^2T_n}{dx^2}\right|_{-1} \;=\; (-1)^n\,\frac{n^4-n^2}{3}. \tag{13.47}$$

For the second expression, the initial value of k can be implemented in numerical codes as $k = n \pmod 2$, and both k and ℓ can be incremented by 2.

Proof. For the first expression, use L'Hôpital's rule to obtain the expressions for $x = \pm 1$.
For the second expression, separate the $k = 0$ term in our summation formula for U_n and use that $T_0 = 1$ to give

$$U_n \;=\; 2\sum_{k=0}^{\infty} h_k\,T_k\,g(k,n) \;=\; g(0,n) + 2\sum_{k=1}^{\infty} T_k\,g(k,n), \tag{13.48}$$

where

$$g(k,n) \;=\; \begin{cases} 1 & (k \le n) \wedge (k+n \text{ even}),\\ 0 & \text{otherwise.}\end{cases} \tag{13.49}$$

Then,

$$\frac{dT_n}{dx} \;=\; n\,U_{n-1} \;=\; n\,g(0,n-1) + 2n\sum_{k=1}^{\infty} T_k\,g(k,n-1). \tag{13.50}$$

We have eliminated h_k. Differentiating a second time, we find

$$\frac{d^2T_n}{dx^2} \;=\; 2n\sum_{k=1}^{\infty} \frac{dT_k}{dx}\,g(k,n-1) \tag{13.51}$$

$$\;=\; 2n\sum_{k=1}^{\infty} 2k\sum_{\ell=0}^{\infty} h_\ell\,T_\ell\,g(\ell,k-1)\,g(k,n-1) \tag{13.52}$$

$$\;=\; 4n\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} k\,h_\ell\,T_\ell\,g(\ell,k-1)\,g(k,n-1) \tag{13.53}$$

$$\;=\; 4n\sum_{\ell=0}^{\infty}\sum_{k=0}^{\infty} k\,h_\ell\,T_\ell\,g(\ell,k-1)\,g(k,n-1) \tag{13.54}$$

$$\;=\; 4n\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} \ell\,h_k\,T_k\,g(k,\ell-1)\,g(\ell,n-1), \tag{13.55}$$

where we added the $k = 0$ term (which is zero) to put the sums into the same form and guarantee that we can interchange their order, and in the final step relabelled $\ell \leftrightarrow k$. Now, $g(k,\ell-1) = 1$ iff $k \le \ell-1$, i.e. $\ell \ge k+1$, and $k+\ell-1$ is even. Similarly, $g(\ell,n-1) = 1$ iff $\ell \le n-1$ and $\ell+n-1$ is even. We have $k+1 \le \ell \le n-1$, which gives us bounds for the sum over ℓ, and moreover $k \le \ell-1 \le n-2$ so that $k \le n-2$, which gives us an

upper bound for the sum over k. Finally, add the two "even" conditions together to give that $k+2\ell+n-2$ is even, so $k+n$ is even. This can be relegated to the sum over k, while $\ell+n-1$ is even, i.e. $\ell+n$ is odd, can be relegated to the sum over ℓ. This yields the given result. Finally, the term $\ell=k+1$ is always considered because then $\ell+n=k+1+n$, but since $k+n$ is always even, this is always odd. Hence, we can start the summation at $\ell=k+2$ and increment by 2 in the "code expression". □

13.2 Chebyshev polynomials on arbitrary interval

Proposition 13.8 (Affine linear transformation)**.** *Let $A,B\in\mathbb{R}:A<B$, let $x\in[-1,1]$ and let $X\in[A,B]$. Then, $\Lambda_{A,B}:[-1,1]\to[A,B]$, given by*

$$\Lambda_{A,B}(x) \;=\; \frac{B-A}{2}\,x+\frac{A+B}{2}, \tag{13.56}$$

describes the transformation of an element $x\in[-1,1]$ to an element $X\in[A,B]$. The transformation back to $[-1,1]$ is described by the inverse function $\Lambda^{-1}_{A,B}:[A,B]\to[-1,1]$, given by

$$\Lambda^{-1}_{A,B}(X) \;=\; \frac{2}{B-A}\,X-\frac{B+A}{B-A}. \tag{13.57}$$

Proof. We require $\Lambda_{A,B}$ to be a linear mapping, so it must have the form $\Lambda_{A,B}(x)=ax+b$, with $a,b\in\mathbb{R}$ to find. The transformation must satisfy $\Lambda_{A,B}(-1)=A$ and $\Lambda_{A,B}(1)=B$. We thus have two simultaneous equations which are easily solved for a,b. The inverse function $\Lambda^{-1}_{A,B}$ is found in the usual way, that is, by setting $X=\Lambda(x)$ and solving for x. □

Corollary 13.9. $\forall\,k\in\mathbb{Z},\ \big[\Lambda_{A,B}(x)\big]^k=\frac{(B-A)^k}{2^k}\big[x+\rho\big]^k$, *where we define* $\rho\equiv\frac{A+B}{A-B}$.

So far, we have defined the Chebyshev polynomials on an interval $x\in[-1,1]$. It will prove useful to define versions of them on an arbitrary interval $X\in[A,B]$, where $A,B\in\mathbb{R}$ and $A<B$. We saw in Definition 13.8 that the two intervals can be mapped to each other by $X=\Lambda_{A,B}(x)$ and $x=\Lambda^{-1}_{A,B}(X)$. Then, our original change of variables $x=\cos(\theta)$ becomes $X=\Lambda_{A,B}\big[\cos(\theta)\big]$ and we can generalise Definition 13.1 by simply replacing x by $\Lambda^{-1}_{A,B}(X)$ and θ by $\cos^{-1}\big[\Lambda^{-1}_{A,B}(X)\big]$:

$$T_{n,A,B}(X) \;=\; T_n\big[\Lambda^{-1}_{A,B}(X)\big] \;=\; \cos\Big[n\,\cos^{-1}\big[\Lambda^{-1}_{A,B}(X)\big]\Big], \tag{13.58}$$

$$U_{n,A,N}(X) \;=\; U_n\big[\Lambda^{-1}_{A,B}(X)\big] \;=\; \frac{\sin\Big[(n+1)\,\cos^{-1}\big[\Lambda^{-1}_{A,B}(X)\big]\Big]}{\sqrt{1-\big[\Lambda^{-1}_{A,B}(X)\big]^2}}. \tag{13.59}$$

As before, let us define that $\forall\,n<0,\ T_n(X)=U_n(X)\equiv 0$. Although n is outside the range of validity, defining the polynomials to be zero allows us to avoid awkward casework when handling derivatives or other relations which decrease the index.

We do not list the first few examples, since they are equal to those on $[-1,1]$, Eqs. (13.5) to (13.14), but with x replaced by $\Lambda^{-1}_{A,B}(X)$. There is no other simplification. For example, $T_2(X)=2\big[\Lambda^{-1}_{A,B}(X)\big]^2-1$.

The recursion relations, Eqs. (13.15) and (13.16), become

$$T_n(X) \;=\; 2\,\Lambda^{-1}_{A,B}(X)\,T_{n-1}(X)-T_{n-2}(X), \tag{13.60}$$

$$U_n(X) \;=\; 2\,\Lambda^{-1}_{A,B}(X)\,U_{n-1}(X)-U_{n-2}(X). \tag{13.61}$$

Proposition 13.10 (Values at the endpoints). $\forall\, n \in \mathbb{N}_0$, *we have*

$$T_n(B) = 1, \tag{13.62}$$
$$U_n(B) = n+1, \tag{13.63}$$

and

$$T_n(A) = (-1)^n, \tag{13.64}$$
$$U_n(A) = (-1)^n\,(n+1). \tag{13.65}$$

Proof. This follows immediately from Proposition 13.2 with $B = \Lambda_{A,B}(1)$ and $A = \Lambda_{A,B}(-1)$. □

Proposition 13.11 (Orthogonality). $\forall\, n, m \in \mathbb{N}_0$, *we have*

$$\frac{2}{B-A}\int_A^B T_n\big[\Lambda_{A,B}^{-1}(X)\big]\,T_m\big[\Lambda_{A,B}^{-1}(X)\big]\,\frac{1}{\sqrt{1-\big[\Lambda_{A,B}^{-1}(X)\big]^2}}\,dX$$
$$= \int_0^\pi T_n\big[\cos(\theta)\big]\,T_m\big[\cos(\theta)\big] = \begin{cases}\pi\,\delta_{n,m} & n=0,\\ \frac{\pi}{2}\,\delta_{n,m} & n\neq 0.\end{cases} \tag{13.66}$$

Proof. Take the orthogonality relation for $[-1,1]$, Eq. (13.32) and perform the change of variables $X = \Lambda_{A,B}(x)$. Then, $dX/dx = \frac{B-A}{2}$ and so $dx = \frac{2}{B-A}\,dX$. The bounds of integration become $-1 \to \Lambda_{A,B}(-1) = A$ and $1 \to \Lambda_{A,B}(1) = B$. This yields the result above. If we were to perform another change of variables $X = \Lambda_{A,B}\big[\cos(\theta)\big]$, we would simply recover the integral in terms of θ in Eq. (13.32). □

Proposition 13.12 (First derivatives). $\forall\, n \in \mathbb{N}_0$, *we have*

$$\frac{dT_n}{dx} = \frac{2}{B-A}\,n\,U_{n-1}\big[\Lambda_{A,B}^{-1}(x)\big] = \sum_{\substack{k=0\\ k+n\ \text{odd}}}^{n-1}\frac{4}{B-A}\,n\,h_k\,T_k\big[\Lambda_{A,B}^{-1}(x)\big]. \tag{13.67}$$

Proof. The chain rule gives $\frac{dT_n}{dx} = \frac{dT_n}{d\Lambda_{A,B}^{-1}}\,\frac{d\Lambda_{A,B}^{-1}}{dx}$. We have that $\frac{dT_n}{d\Lambda_{A,B}^{-1}} = n\,U_{n-1}(\Lambda_{A,B}^{-1})$ and from Proposition 13.8 we have $\frac{d\Lambda_{A,B}^{-1}}{dx} = \frac{2}{B-A}$. □

13.3 Proof that the Chebyshev polynomials are indeed polynomials

Lemma 13.13. *A sum of powers of ix, where $i = \sqrt{-1}$, each with coefficients $b_k \in \mathbb{R}$, can be written*

$$\sum_{k=0}^{n} b_k i^k x^k \;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} x^{4k}\Big[b_{4k} - b_{4k+2}x^2 + ix(b_{4k+1} - b_{4k+3}x^2)\Big] + r(n), \tag{13.68}$$

where $r(n)$ is a "remainder" that accounts for the fact that n might not be a multiple of 4, and contains the leftover terms from the highest value of k:

$$r(n) \;=\; \begin{cases} b_n x^n & n \equiv 0 \;(\mod 4), \\ b_{n-1}x^{n-1} + ib_n x^n & n \equiv 1 \;(\mod 4), \\ x^{n-2}(b_{n-2} - b_n x^2) + ib_{n-1}x^{n-1} & n \equiv 2 \;(\mod 4), \\ 0 & n \equiv 3 \;(\mod 4). \end{cases} \tag{13.69}$$

Proof. The powers of i follow a cycle of length 4, which means that for integer k we have

$$i^{4k} \;=\; i^0 \;=\; 1, \qquad i^{4k+1} \;=\; i^1 \;=\; i, \qquad i^{4k+2} \;=\; i^2 \;=\; -1, \qquad i^{4k+3} \;=\; i^3 \;=\; -i. \tag{13.70}$$

Then, a sum of powers of i, each with coefficients a_k, can easily be split into real and imaginary parts:

$$\sum_{k=0}^{n} a_k i^k \;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} \left[i^0 a_{4k} + i^1 a_{4k+1} + i^2 a_{4k+2} + i^3 a_{4k+3}\right] + r(n) \;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} \left[a_{4k} - a_{4k+2} + i(a_{4k+1} - a_{4k+3})\right] + r(n), \tag{13.71}$$

with "remainder" $r(n)$ given by

$$r(n) \;=\; \begin{cases} a_n & n \equiv 0 \;(\mod 4), \\ a_{n-1} + ia_n & n \equiv 1 \;(\mod 4), \\ a_{n-2} - a_n + ia_{n-1} & n \equiv 2 \;(\mod 4), \\ 0 & n \equiv 3 \;(\mod 4). \end{cases} \tag{13.72}$$

If the coefficients contain increasing powers of x, $a_k = b_k x^k$, where b_k is a general coefficient, these simplify to the given expressions. □

Lemma 13.14. $\forall\, n, x, a \in \mathbb{N},\ \binom{n}{x+a} = \binom{x+a}{a}^{-1}\binom{n-x}{a}\binom{n}{x}$.

Proof. We have

$$\binom{n}{x+1} \;=\; \frac{n!}{(x+1)!(n-x-1)!} \;=\; \frac{n!}{(x+1)x!\,\frac{(n-x)!}{n-x}} \;=\; \frac{n-x}{x+1}\binom{n}{x}, \tag{13.73}$$

$$\binom{n}{x+2} \;=\; \frac{n!}{(x+2)(x+1)x!\,\frac{(n-x)!}{(n-x)(n-x-1)}} \;=\; \frac{(n-x)(n-x-1)}{(x+2)(x+1)}\binom{n}{x} \;=\; \frac{(n-x)!\,x!}{(n-x-2)!\,(x+2)!}\binom{n}{x}, \tag{13.74}$$

so by extension the general rule is the stated expression, where $\binom{n-x}{a} = \frac{(n-x)!}{a!\,(n-x-a)!}$ and $\frac{a!\,x!}{a!\,(x+a)!} = \binom{x+a}{a}^{-1}$. □

Proposition 13.15. *$\forall\, n \in \mathbb{N}$, we can express*

$$\cos(n\theta) \;=\; P(x), \qquad \sin(n\theta) \;=\; \tilde{P}(x)\,\sin(\theta), \tag{13.75}$$

where $P(x), \tilde{P}(x)$ are general polynomials in $\cos(\theta)$.

Proof. Combine de Moivre's theorem $\big[\cos(x) + i\sin(\theta)\big]^n = \cos(nx) + i\sin(\theta)$ with the binomial theorem $(a+b)^n \;=\; \sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^k$. Using the notational shorthand $c \equiv \cos(\theta)$ and $s \equiv \sin(\theta)$, this gives

$$\cos(n\theta) + i\sin(n\theta) \;=\; \sum_{k=0}^{n}\binom{n}{k}c^{n-k}(is)^k \;=\; \sum_{k=0}^{n}\binom{n}{k}c^{n-k}s^k i^k. \tag{13.76}$$

This is an example of our sum (13.68) with coefficients $a_k = \binom{n}{k}c^{n-k}s^k$, so we can write it as

$$\begin{aligned}
\cos(n\theta) + i\sin(n\theta) \;=\; & \sum_{k=0}^{\lfloor n/4\rfloor}\Bigg[\binom{n}{4k}c^{n-4k}s^{4k} - \binom{n}{4k+2}c^{n-4k-2}s^{4k+2} \\
& \qquad + i\Bigg\{\binom{n}{4k+1}c^{n-4k-1}s^{4k+1} - \binom{n}{4k+3}c^{n-4k-3}s^{4k+3}\Bigg\}\Bigg] + r(n) \qquad (13.77)\\
\;=\; & \sum_{k=0}^{\lfloor n/4\rfloor} c^{n-4k-3}s^{4k}\Bigg[\binom{n}{4k}c^3 - \binom{n}{4k+2}cs^2 + i\Bigg\{\binom{n}{4k+1}c^2 s - \binom{n}{4k+3}s^3\Bigg\}\Bigg] + r(n) \qquad (13.78)\\
\;=\; & \sum_{k=0}^{\lfloor n/4\rfloor} c^{n-4k-3}(1-c^2)^{2k}\Bigg[\binom{n}{4k}c^3 - \binom{n}{4k+2}c(1-c^2) \\
& \qquad + is\Bigg\{\binom{n}{4k+1}c^2 - \binom{n}{4k+3}(1-c^2)\Bigg\}\Bigg] + r(n), \qquad (13.79)
\end{aligned}$$

where the remainder is

$$r(n) \;=\; \begin{cases} s^n & n \equiv 0 \;(\mod 4\;), \\ (n-1)cs^{n-1} + is^n & n \equiv 1 \;(\mod 4\;), \\ s^{n-2}\big[\frac{1}{2}n(n-1)c^2 - s^2\big] + i(n-1)cs^{n-1} & n \equiv 2 \;(\mod 4\;), \\ 0 & n \equiv 3 \;(\mod 4\;). \end{cases} \tag{13.80}$$

We could use Lemma 13.14 to handle the binomial coefficients, but the simplification is only numerical and not aesthetical; it will actually make the expression look messier. Further, we can appreciate that the binomial coefficient is just an integer, so it makes sense to leave it untouched.
The sum follows the required rules, namely that its real part is a polynomial in $\cos(\theta)$ and its imaginary part is $\sin(\theta)$ times a polynomial in $\cos(\theta)$. The remainder contains mixed terms, but we can put it in the required form as follows. If n is even, we can write $n = 2m$ where k is an integer, so that

$$r(n \equiv 0 \bmod 4) \;=\; s^{2m} \;=\; (1-c^2)^m \;=\; (1-c^2)^{n/2}, \tag{13.81}$$

and

$$\begin{aligned}
r(n \equiv 2 \bmod 4) \;=\; & s^{2m-2}\big[m(2m-1)c^2 - s^2\big] + i(2m-1)cs^{2m-1} & (13.82)\\
\;=\; & (1-c^2)^{m-1}\big[m(2m-1)c^2 + 1 - c^2\big] + i(2m-1)cs^{2m-2}s & (13.83)\\
\;=\; & (1-c^2)^{m-1}\big[(2m+1)(m-1)c^2 + 1\big] + i(2m-1)c(1-c^2)^{m-1}s, & (13.84)\\
\;=\; & (1-c^2)^{(n-2)/2}\left[\frac{1}{2}\,n(n-3)c^2 + 1\right] + i(n-2)c(1-c^2)^{(n-2)/2}s. & (13.85)
\end{aligned}$$

If n is odd, we can write $n = 2m+1$ where k is an integer, so that

$$r(n \equiv 1 \bmod 4) \;=\; 2mcs^{2m} + is^{2m+1} \;=\; 2mc(1-c^2)^m + i(1-c^2)s \;=\; (n-1)c(1-c^2)^{(n-1)/2} + i(1-c^2)s, \tag{13.86}$$

and still $r(\equiv 3 \bmod 4) = 0$. To summarise,

$$r(n) \;=\; \begin{cases} (1-c^2)^{n/2} & n \equiv 0 \pmod 4, \\ (n-1)c(1-c^2)^{(n-1)/2} + i(1-c^2)s & n \equiv 1 \pmod 4, \\ (1-c^2)^{(n-2)/2}\left[\frac{1}{2}n(n-3)c^2+1\right] + i(n-2)c(1-c^2)^{(n-2)/2}s & n \equiv 2 \pmod 4, \\ 0 & n \equiv 3 \pmod 4. \end{cases} \tag{13.87}$$

In all cases, the real part of the remainder is a polynomial in $\cos(\theta)$ and the imaginary part is $\sin(\theta)$ times a polynomial in $\cos(\theta)$. Polynomials of any order can be added to give another polynomial, so the entire RHS of our expression (13.77) follows this rule. Equating real and imaginary parts, the proof is complete. □

Corollary 13.16. *$\forall\,\theta \in \mathbb{R}$ and $\forall\, n \in \mathbb{N}$, we have*

$$\cos(n\theta) \;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} c^{n-4k-3}(1-c^2)^{2k}\left[\binom{n}{4k}c^3 - \binom{n}{4k+2}c(1-c^2)\right] + \mathrm{Re}\big[r(n)\big] \tag{13.88}$$

$$\;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} \binom{n}{4k} c^{n-4k-2}(1-c^2)^{2k}\left[c^2 - \frac{(n-4k)(n-4k-1)}{(4k+2)(4k+1)}(1-c^2)\right] + \mathrm{Re}\big[r(n)\big], \tag{13.89}$$

where

$$\mathrm{Re}\big[r(n)\big] \;=\; \begin{cases} (1-c^2)^{n/2} & n \equiv 0 \pmod 4, \\ (n-1)c(1-c^2)^{(n-1)/2} & n \equiv 1 \pmod 4, \\ (1-c^2)^{(n-2)/2}\left[\frac{1}{2}n(n-3)c^2+1\right] & n \equiv 2 \pmod 4, \\ 0 & n \equiv 3 \pmod 4, \end{cases} \tag{13.90}$$

and

$$\sin(n\theta) \;=\; \sum_{k=0}^{\lfloor n/4 \rfloor} c^{n-4k-3}(1-c^2)^{2k}s\left[\binom{n}{4k+1}c^2 - \binom{n}{4k+3}(1-c^2)\right] + \mathrm{Im}\big[r(n)\big] \tag{13.91}$$

$$\;=\; s\sum_{k=0}^{\lfloor n/4 \rfloor} \binom{n}{4k+1} c^{n-4k-3}(1-c^2)^{2k}\left[c^2 - \frac{(n-4k-1)(n-4k-2)}{4k+3)(4k+1)}(1-c^2)\right] + \mathrm{Im}\big[r(n)\big], \tag{13.92}$$

where

$$\mathrm{Im}\big[r(n)\big] \;=\; \begin{cases} 0 & n \equiv 0 \pmod 4 \text{ or } n \equiv 3 \pmod 4, \\ (1-c^2)s & n \equiv 1 \pmod 4, \\ (n-2)c(1-c^2)^{(n-2)/2}s & n \equiv 2 \pmod 4. \end{cases} \tag{13.93}$$

14 Chebyshev series

14.1 Chebyshev series on $[-1, 1]$

Proposition 14.1 (Chebyshev series on $[-1,1]$)**.** *Let $f(x)$ be a function defined on $[-1,1]$. Then, the function can be written as an infinite sum of Chebyshev polynomials $T_n(x)$:*

$$f(x) \;=\; \sum_{n=0}^{\infty} a_n\, T_n(x), \tag{14.1}$$

with coefficients

$$a_n \;=\; \frac{2\,h_n}{\pi}\int_{-1}^{1} f(x)\,T_n(x)\,\frac{dx}{\sqrt{1-x^2}} \;=\; \frac{2\,h_n}{\pi}\int_{0}^{\pi} f\big[\cos(\theta)\big]\,\cos(n\theta)\,d\theta, \tag{14.2}$$

and in particular

$$a_0 \;=\; \frac{1}{\pi}\int_{-1}^{1} f(x)\,\frac{dx}{\sqrt{1-x^2}} \;=\; \frac{1}{\pi}\int_{0}^{\pi} f\big[\cos(\theta)\big]\,d\theta, \tag{14.3}$$

where we define for notational convenience that

$$h_n \;\equiv\; 1-\frac{1}{2}\,\delta_{n,0} \;=\; \begin{cases} \frac{1}{2} & n=0, \\ 1 & n\neq 0. \end{cases} \tag{14.4}$$

Proof. Multiply $f(x)$ by $T_n(x)$, integrate over the domain and substitute the desired expression for $f(x)$ as a sum of coefficients multiplied by Chebyshev polynomials:

$$\int_{-1}^{1} f(x)\,T_n(x)\,\frac{dx}{\sqrt{1-x^2}} \;=\; \int_{-1}^{1}\sum_{m=0}^{\infty} a_m\,T_m(x)\,T_n(x)\,\frac{dx}{\sqrt{1-x^2}} \;=\; \sum_{m=0}^{\infty} a_m\int_{-1}^{1} T_m(x)\,T_n(x)\,\frac{dx}{\sqrt{1-x^2}}. \tag{14.5}$$

Substitute the orthogonality condition (13.32). Only the $n=m$ term in the sum yields a nonzero integral: if $n=0$, the integral is π, otherwise the integral is $\pi/2$. For the $n\neq 0$ case, we then have

$$\int_{-1}^{1} f(x)\,T_n(x)\,\frac{dx}{\sqrt{1-x^2}} \;=\; a_n\,\frac{\pi}{2}, \tag{14.6}$$

which rearranges to the first given expression for a_n. For $n=0$, the prefactor will become $1/\pi$ instead of $2/\pi$. We can retain the same expression and account for this by halving a_0. When performing numerical integration, the division by $\sqrt{1-x^2}$ will cause issues at the endpoints ± 1, so it is necessary to perform a change of variables. Substitute $x=\cos(\theta)$ so that $dx/d\theta = -\sin(\theta)$ and $dx/\sqrt{1-x^2} = dx/\sin(\theta) = -d\theta$. The limits become $-1 \to \cos^{-1}(-1) = 0$ and $1 \to \cos^{-1}(0) = \pi$, and we swap the limits to absorb the minus sign. We also note that $T_n[\cos(\theta)] = \cos(n\theta)$ by definition. This gives

$$a_n \;=\; \frac{2}{\pi}\int_{\pi}^{0} f\big[\cos(\theta)\big]\,T_n\big[\cos(\theta)\big]\,\cdot -d\theta \;=\; \frac{2}{\pi}\int_{0}^{\pi} f\big[\cos(\theta)\big]\,\cos(n\theta)\,d\theta. \tag{14.7}$$

The proof is now complete. □

Proposition 14.2 (First derivative of Chebyshev series on $[-1,1]$)**.** *Let $f(x)$ be a function defined on $[-1,1]$ whose Chebyshev series has been found. Then,*

$$\frac{df}{dx} = \sum_{n=1}^{\infty} a_n\, n\, U_{n-1}(x) \tag{14.8}$$

$$= \sum_{n=0}^{\infty} \Bigg(\sum_{\substack{k=n+1 \\ k+n \text{ odd}}}^{\infty} 2h_n\, k\, a_k \Bigg) T_n. \tag{14.9}$$

The second expression is the Chebyshev series expansion of $\frac{df}{dx}$.

Proof. To obtain the first expression, differentiate the expression in Proposition 14.1 term-by-term, using the result of Proposition 13.6. To obtain the second expression, substitute U_{n-1} for a sum over T_k by Proposition 13.5:

$$\frac{df}{dx} = \sum_{n=0}^{\infty} a_n\, n \sum_{k=0}^{\infty} 2h_k\, T_k(x)\, g(k, n-1) \tag{14.10}$$

$$= 2\sum_{n=0}^{\infty}\sum_{k=0}^{\infty} a_n\, n\, h_k\, T_k(x)\, g(k, n-1), \tag{14.11}$$

$$= 2\sum_{k=0}^{\infty}\sum_{n=0}^{\infty} a_n\, n\, h_k\, T_k(x)\, g(k, n-1), \tag{14.12}$$

where

$$g(k, n-1) = \begin{cases} 1 & (k \leq n-1) \wedge (k+n-1 \text{ even}), \\ 0 & \text{otherwise}, \end{cases} \tag{14.13}$$

$$= \begin{cases} 1 & (k \leq n-1) \wedge (k+n \text{ odd}), \\ 0 & \text{otherwise}, \end{cases} \tag{14.14}$$

$$= \begin{cases} 1 & (n \geq k+1) \wedge (k+n \text{ odd}), \\ 0 & \text{otherwise}. \end{cases} \tag{14.15}$$

Having rewritten the conditions within g explicitly as conditions on n, we can relegate them to labels in the sum over n:

$$\frac{df}{dx} = 2\sum_{k=0}^{\infty} \sum_{\substack{n=k+1 \\ n+k \text{ odd}}}^{\infty} a_n\, n\, h_k\, T_k(x). \tag{14.16}$$

We obtain the final result by relabelling $k \leftrightarrow n$. □

Verification. `Chebyshev series/Chebyshev_series_02.cpp` □

Proposition 14.3 (Second derivative of Chebyshev series on $[-1,1]$)**.** *Let $f(x)$ be a function defined on $[-1,1]$ whose Chebyshev series has been found. Then, the Chebyshev series expansion of $\frac{d^2f}{dx^2}$ is*

$$\frac{d^2f}{dx^2} = \sum_{n=0}^{\infty}\Bigg(\sum_{\substack{k=n+2\\ n+k \text{ even}}}^{\infty} h_n\, k(k^2-n^2)\, a_k\Bigg) T_n. \tag{14.17}$$

Proof. We have

$$\frac{d^2f}{dx^2} = \sum_{n=0}^{\infty} a_n \frac{d^2T_n}{dx^2} \tag{14.18}$$

$$= \sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} 4n\ell\, a_n\, h_k\, T_k\, g(k,\ell-1)\, g(\ell,n-1) \tag{14.19}$$

$$= \sum_{k=0}^{\infty}\sum_{n=0}^{\infty}\sum_{\ell=0}^{\infty} 4n\ell\, a_n\, h_k\, T_k\, g(k,\ell-1)\, g(\ell,n-1) \tag{14.20}$$

$$= \sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} 4k\ell\, a_k\, h_n\, T_n\, g(n,\ell-1)\, g(\ell,k-1), \tag{14.21}$$

where in the penultimate step we switched the order of summation over n and k, and in the final step we relabelled $n \leftrightarrow k$. We have that $g(n,\ell-1)=1$ iff $n \le \ell-1$, i.e. $\ell \ge n+1$, and $n+\ell-1$ is even, i.e. $n+\ell$ is odd. Similarly, $g(\ell,k-1)=1$ iff $\ell \le k-1$ and $\ell+k-1$ is even. Adding the two "even" conditions, we find $n+2\ell+k-2$ is even, i.e. $n+k$ is even. The bounds on ℓ give $n+1 \le \ell \le k-1$, i.e. $n+2 \le \ell+1 \le k$, so $k \ge n+2$. Relegating the conditions to the summation indices, we obtain the given result. Finally, if $\ell = n+1$ then $n+\ell-1 = 2n-2$ is always even, and if $k = n+2$ then $n+k = 2n+2$ is always even, so the lower bounds of $\ell = k+1$ and $k = n+2$ are always triggered. We will arrive at

$$\frac{d^2f}{dx^2} = \sum_{n=0}^{\infty}\Bigg(\sum_{\substack{k=n+2\\ n+k \text{ even}}}^{\infty}\ \sum_{\substack{q=n+1\\ n+q \text{ odd}}}^{k-1} 4k\, q\, a_k\, h_n\Bigg) T_n. \tag{14.22}$$

It can be shown that

$$\sum_{\substack{q=n+1\\ n+\ell \text{ odd}}}^{k-1} q = \frac{k^2-n^2}{4}, \tag{14.23}$$

from which the given result follows. □

Verification. `Chebyshev series/20230627_sum_over_q.cpp` □

Proposition 14.4 (Chebyshev series of product of two functions on $[-1,1]$)**.** *Let $f(x)$ and $g(x)$ two functions whose Chebyshev series are known. Then, the Chebyshev series coefficients of $f(x)\,g(x)$ are*

$$(fg)_0 \;=\; f_0\,g_0 + \frac{1}{2}\sum_{n_1=1}^{\infty} f_{n_1}\,g_{n_1}, \tag{14.24}$$

$$(fg)_{n>0} \;=\; \frac{1}{2}\sum_{n_1=0}^{n} f_{n_1}\,g_{n-n_1} + \frac{1}{2}\sum_{n_1=n}^{\infty}\big[f_{n_1}\,g_{n_1-n} + f_{n_1-n}\,g_{n_1}\big]. \tag{14.25}$$

Proof. Suppose that the Chebyshev series coefficients of $f(x)$ are f_n and those of $g(x)$ are g_n. Then,

$$f(x)\,g(x) \;=\; \Big(\sum_{n=0}^{\infty} f_n\,T_n(x)\Big)\Big(\sum_{n=0}^{\infty} g_n\,T_n(x)\Big) \tag{14.26}$$

$$\;=\; \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\,T_{n_1}(x)\,T_{n_2}(x) \tag{14.27}$$

$$\;=\; \frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\big[T_{n_1+n_2}(x) + T_{n_1-n_2}(x)\big], \tag{14.28}$$

where we used that $T_n(x)\,T_m(x) = \frac{1}{2}\big[T_{n+m}(x) + T_{n-m}(x)\big]$, and where $T_{-n}(x) = -T_n(x)$. This is a function of x, so it must possess a Chebyshev series with coefficients

$$(fg)_n \;=\; \frac{h_n}{\pi}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\bigg(\int_{-1}^{1} T_{n_1+n_2}(x)\,\frac{dx}{\sqrt{1-x^2}} + \int_{-1}^{1} T_{n_1-n_2}(x)\,\frac{dx}{\sqrt{1-x^2}}\bigg). \tag{14.29}$$

For $n = 0$,

$$(fg)_0 \;=\; \frac{1}{2\pi}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\big(\pi\,\delta_{n_1+n_2,0} + \pi\,\delta_{n_1-n_2,0}\big) \;=\; \frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\big(\delta_{n_1+n_2,0} + \delta_{n_1-n_2,0}\big). \tag{14.30}$$

We have that $n_1 \in [0,\infty]$ and $n_2 \in [0,\infty]$. Then, $n_1 + n_2 = 0$ iff $n_1 = n_2 = 0$, so the first of the double-sums collapses to leave only the $(n_1,n_2) = (0,0)$ terms. Further, $n_1 - n_2 = 0$ iff $n_2 = n_1$, so the second sum over n_2 collapses to only the $n_2 = n_1$ term. We have

$$(fg)_0 \;=\; \frac{1}{2}\,f_0\,g_0 + \frac{1}{2}\sum_{n_1=0}^{\infty} f_{n_1}\,g_{n_1} \;=\; \frac{1}{2}\,f_0\,g_0 + \bigg(\frac{1}{2}\,f_0\,g_0 + \frac{1}{2}\sum_{n_1=1}^{\infty} f_{n_1}\,g_{n_1}\bigg) \;=\; f_0\,g_0 + \frac{1}{2}\sum_{n_1=1}^{\infty} f_{n_1}\,g_{n_1}. \tag{14.31}$$

For $n > 0$, due to our double sum, we must be sure to consider the cases where $n_1 - n_2 = n$ and where $n_2 - n_1 = n$:

$$(fg)_{n>0} \;=\; \frac{1}{\pi}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\bigg(\frac{\pi}{2}\,\delta_{n_1+n_2,n} + \frac{\pi}{2}\,\delta_{n_1-n_2,n} + \frac{\pi}{2}\,\delta_{n_2-n_1,n}\bigg) \tag{14.32}$$

$$\;=\; \frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\,g_{n_2}\big(\delta_{n_1+n_2,n} + \delta_{n_1-n_2,n} + \delta_{n_2-n_1,n}\big). \tag{14.33}$$

Now, $n_1 + n_2 = n$ iff $n_2 = n - n_1$, so

$$\sum_{n_2=0}^{\infty} g_{n_2}\,\delta_{n_1+n_2,n} \;=\; g_{n-n_1}. \tag{14.34}$$

Since $\forall\, m < 0,\ g_m = 0$, we may enforce $n - n_1 \geq 0$ and so $n_1 \leq n$ to obtain an upper bound on the summation over n_1, giving

$$\frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\, g_{n_2}\, \delta_{n_1+n_2,n} \;=\; \frac{1}{2}\sum_{n_1=0}^{n} f_{n_1}\, g_{n-n_1}. \tag{14.35}$$

Similarly, $n_1 - n_2 = n$ iff $n_2 = n_1 - n$, so

$$\sum_{n_2=0}^{\infty} g_{n_2}\, \delta_{n_1-n_2,n} \;=\; g_{n_1-n}, \tag{14.36}$$

and we may enforce $n_1 - n \geq 0$ so that $n_1 \geq n$ to obtain a lower bound on the summation over n_1, giving

$$\frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\, g_{n_2}\, \delta_{n_1-n_2,n} \;=\; \frac{1}{2}\sum_{n_1=n}^{\infty} f_{n_1}\, g_{n_1-n}. \tag{14.37}$$

Finally, $n_2 - n_1 = n$ iff $n_1 = n_2 - n$ and we can make the same argument, leading to

$$\frac{1}{2}\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} f_{n_1}\, g_{n_2}\, \delta_{n_2-n_1,n} \;=\; \frac{1}{2}\sum_{n_1=n}^{\infty} f_{n_1-n}\, g_{n_1}. \tag{14.38}$$

We are free to relabel $n_2 \to n_1$. Combining the three sums, we obtain the given result. □

Verification. `Chebyshev series/Chebyshev_series_f_times_g_05.cpp` □

Example 14.5. *The Chebyshev series coefficients of $x\, f(x)$ are*

$$(xf)_n \;=\; \begin{cases} \dfrac{1}{2}\, f_1 & n = 0, \\[2ex] f_0 + \dfrac{1}{2}\, f_2 & n = 1, \\[2ex] \dfrac{1}{2}\, f_{n-1} + \dfrac{1}{2}\, f_{n+1} & n > 1. \end{cases} \tag{14.39}$$

Proof. Use the result of Proposition 14.4 with $f(x) = x$ and $g(x) = 1$. The Chebyshev series coefficients of $f(x)$ are simply $f_n = \delta_{n,1}$, and those of $g(x)$ are unknown g_n. Then,

$$(fg)_0 \;=\; \delta_{0,1}\, g_0 + \frac{1}{2}\sum_{n_1=1}^{\infty} \delta_{n_1,1}\, g_{n_1}. \tag{14.40}$$

The first term is zero; the only nonzero term in the sum over n_1 is that with $n_1 = 1$, so it collapses to $(fg)_0 = \frac{1}{2}\, g_1$. Next, it is instructive to consider the $n = 1$ coefficient separately:

$$(fg)_1 \;=\; \frac{1}{2}\sum_{n_1=0}^{1} \delta_{n_1,1}\, g_{1-n_1} + \frac{1}{2}\sum_{n_1=1}^{\infty} \delta_{n_1,1}\, g_{n_1-1} + \frac{1}{2}\sum_{n_1=1}^{\infty} \delta_{n_1-1,1}\, g_{n_1}. \tag{14.41}$$

The first and second sums collapse to the $n_1 = 1$ term so that $1 - n_1 = n_1 - 1 = 0$. The third sum collapses to $n_1 - 1 = 1$, or $n_1 = 2$. This gives $(fg)_1 = g_0 + \frac{1}{2}\, g_2$. Finally, consider the coefficients with $n > 1$:

$$(fg)_{n>1} \;=\; \frac{1}{2}\sum_{n_1=0}^{n} \delta_{n_1,1}\, g_{n-n_1} + \frac{1}{2}\sum_{n_1=n}^{\infty} \delta_{n_1,1}\, g_{n_1-n} + \frac{1}{2}\sum_{n_1=n}^{\infty} \delta_{n_1-n,1}\, g_{n_1}. \tag{14.42}$$

The first sum collapses to the $n_1 = 1$ term so that $n - n_1 = n - 1$; the second sum collapses to zero because $n_1 \neq 1$; the third sum collapses to the $n_1 - n = 1$ term so that $n_1 = n + 1$. This gives $(fg)_{n>1} = \frac{1}{2}\, g_{n-1} + \frac{1}{2}\, g_{n+1}$. □

14.2 Chebyshev series on arbitrary interval

Proposition 14.6 (Chebyshev series on arbitrary interval)**.** *Let $A, B \in \mathbb{R} : A < B$ and let $f(X)$ be a function defined on $X \in [A, B]$. Then, the function can be written as an infinite sum of linearly transformed Chebyshev polynomials $T_n\big[\Lambda^{-1}_{A,B}(X)\big]$:*

$$f(X) \;=\; \sum_{n=0}^{\infty} a_n \, T_n\big[\Lambda^{-1}_{A,B}(X)\big], \tag{14.43}$$

with coefficients

$$a_n \;=\; \frac{2\,h_n}{\pi}\,\frac{2}{B-A}\int_A^B f(X)\,T_n\big[\Lambda^{-1}_{A,B}(X)\big]\,\frac{1}{\sqrt{1-\big[\Lambda^{-1}_{A,B}(X)\big]^2}}\,dX \tag{14.44}$$

$$\;=\; \frac{2\,h_n}{\pi}\int_0^{\pi} f\Big(\Lambda_{A,B}\big[\cos(\theta)\big]\Big)\,\cos(n\theta)\,d\theta, \tag{14.45}$$

and in particular

$$a_0 \;=\; \frac{1}{\pi}\int_0^{\pi} f\Big(\Lambda_{A,B}\big[\cos(\theta)\big]\Big)\,d\theta, \tag{14.46}$$

where h_n is a notational shorthand introduced in Eq. (14.4) and the linear transformation $\Lambda_{A,B}$ and its inverse $\Lambda^{-1}_{A,B}$ are described in Proposition 13.8.

Proof. The proof proceeds in the same way as for Proposition 14.1. First, take $f(X)$, multiply it by $T_n\big[\Lambda^{-1}_{A,B}(X)\big]$ and integrate, labelling by I_n:

$$I_n \;=\; \frac{2}{B-A}\int_A^B f(X)\,T_n\big[\Lambda^{-1}_{A,B}(X)\big]\,\frac{1}{\sqrt{1-\big[\Lambda^{-1}_{A,B}(X)\big]^2}}\,dX \tag{14.47}$$

$$\;=\; \frac{2}{B-A}\int_A^B \sum_{m=0}^{\infty} a_m\,T_m\big[\Lambda^{-1}_{A,B}(X)\big]\,T_n\big[\Lambda^{-1}_{A,B}(X)\big]\,\frac{1}{\sqrt{1-\big[\Lambda^{-1}_{A,B}(X)\big]^2}}\,dX \tag{14.48}$$

$$\;=\; \sum_{m=0}^{\infty} a_m\,\frac{2}{B-A}\int_A^B T_m\big[\Lambda^{-1}_{A,B}(X)\big]\,T_n\big[\Lambda^{-1}_{A,B}(X)\big]\,\frac{1}{\sqrt{1-\big[\Lambda^{-1}_{A,B}(X)\big]^2}}\,dX. \tag{14.49}$$

If $n = 0$, this yields

$$I_0 \;=\; \sum_{m=0}^{\infty} a_m\,\pi\,\delta_{0,m} \;=\; a_0\,\pi. \tag{14.50}$$

If $n \neq 0$, this yields

$$I_n \;=\; \sum_{m=0}^{\infty} a_m\,\frac{\pi}{2}\,\delta_{0,m} \;=\; a_n\,\frac{\pi}{2}, \qquad n \neq 0. \tag{14.51}$$

Rearranging the expressions for I_n, and using the definition of h_n from Eq. (14.4) to combine the two cases, we obtain the given result as an integral over X. Now, perform the change of variables $X = \Lambda_{A,B}\big[\cos(\theta)\big]$ so that

$$\frac{dX}{d\theta} \;=\; \frac{dX}{d\Lambda_{A,B}}\,\frac{d\Lambda_{A,B}}{d\cos(\theta)}\,\frac{d\cos(\theta)}{d\theta} \;=\; 1\cdot\frac{B-A}{2}\cdot -\sin(\theta, \tag{14.52}$$

$$\Rightarrow\; dX \;=\; -\frac{(B-A)}{2}\,\sin(\theta), \tag{14.53}$$

the limits become

$$B \;\to\; \cos^{-1}\left[\Lambda_{A,B}^{-1}(B)\right] \;=\; \cos^{-1}(1) \;=\; 0, \tag{14.54}$$

$$A \;\to\; \cos^{-1}\left[\Lambda_{A,B}^{-1}(A)\right] \;=\; \cos^{-1}(-1) \;=\; \pi, \tag{14.55}$$

$$\tag{14.56}$$

and we note that

$$\frac{1}{\sqrt{1-\left[\Lambda_{A,B}^{-1}(X)\right]^2}} \;=\; \frac{1}{\sqrt{1-\cos^2(\theta)}} \;=\; \frac{1}{\sin(\theta)}. \tag{14.57}$$

Substituting all of these to the integral in terms of X, and swapping the limits to absorb the minus sign, we obtain the integral in terms of θ. □

Example 14.7. *The Chebyshev series coefficients of* $f(x) = \frac{1}{x}$ *on an arbitrary interval* $[A, B]$ *are*

$$a_n \;=\; \frac{4\,h_n}{\pi(B-A)} \int_0^\pi \frac{\cos(n\theta)}{\cos(\theta)+\rho}\,d\theta, \tag{14.58}$$

where $\rho = \frac{A+B}{A-B}$.

Proof. Note that $\lim_{x\to 0^\pm} \frac{1}{x} = \pm\infty$ and so $\lim_{x\to 0} \frac{1}{x}$ does not exist and a Chebyshev series cannot be found in intervals containing $x = 0$. We have

$$a_n \;=\; \frac{2\,h_n}{\pi} \int_0^\pi \frac{1}{\Lambda\big[\cos(\theta)\big]} \cos(n\theta)\,d\theta \;=\; \frac{2\,h_n}{\pi} \int_0^\pi \frac{1}{\frac{B-A}{2}\cos(\theta)+\frac{B+A}{2}} \cos(n\theta)\,d\theta \tag{14.59}$$

$$\;=\; \frac{2\,h_n}{\pi} \int_0^\pi \frac{1}{\frac{B-A}{2}\left[\cos(\theta)+\frac{B+A}{2}\frac{2}{B-A}\right]} \cos(n\theta)\,d\theta \;=\; \frac{4\,h_n}{\pi(B-A)} \int_0^\pi \frac{1}{\cos(\theta)+\frac{B+A}{B-A}} \cos(n\theta)\,d\theta. \tag{14.60}$$

Defining $\rho = \frac{B+A}{B-A}$ to absorb the constants, the result follows. □

From Example 14.7, the Chebyshev series of $f(x) = \frac{1}{x}$ on an arbitrary interval consists of infinitely many terms. In reality we must truncate at some $n_{\max}$, limiting the accuracy of a Chevyshev series decomposition for modelling this function. Typically we have $\lim_{n\to\infty} |a_n| = 0$, so for sufficient $n_{\max}$ the truncation error will be negligible.

However, the integral for a_n in Example 14.7 can only be calculated numerically, not analytically. This means that even low-n coefficients will be subject to numerical error. Further, there will be some $n_{\max}$ beyond which the total error by using more coefficients outweighs the accuracy gained, adding an upper limit to the possible accuracy. We will experience this during testing in §24.2.

The accuracy of a Chebyshev representation of $\frac{1}{x}$ is limited not only by our choice of $n_{\max}$, but also by our coordinate spacing. we appreciate that the same issue will occur with $f(x) = \frac{1}{x^3}$, which is the typical radial behaviour we will expect in our simulations (Chapter 14.7).

Lemma 14.8. $\forall\, a, b \in \mathbb{R}$,

$$\int \cos(ax)\,\cos(bx)\,dx \;=\; \frac{1}{2}\left[\frac{\sin\big[(a+b)x\big]}{a+b} + \frac{\sin\big[(a-b)x\big]}{a-b}\right], \qquad |a| \neq b, \tag{14.61}$$

$$\int \cos(ax)\,\cos(ax)\,dx \;=\; \int \cos(ax)\,\cos(-ax)\,dx \;=\; \frac{x}{2} + \frac{1}{4a}\,\sin(2ax), \tag{14.62}$$

where the addition of an arbitrary constant is implied.

Proof. For the case $a \neq b$, write $\cos(x)$ in terms of exponentials. For the case $a = b$, we have a standard integral. □

Lemma 14.9. $\forall\, k \in \mathbb{N}$ *and* $\forall\, n \in \mathbb{N}$,

$$\int \cos^k(x)\cos(nx)\,dx \;=$$

$$\begin{cases} \dfrac{1}{2^{k+1}} \displaystyle\sum_{\substack{j=0 \\ j\neq(k\pm n)/2}}^{k} \binom{k}{j}\left[\frac{\sin\big[(2j-k+n)x\big]}{2j-k+n} + \frac{\sin\big[(2j-k-n)x\big]}{2j-k-n}\right] \\ \qquad\qquad + \dfrac{1}{2^k}\binom{k}{\frac{k-n}{2}}\left[x + \frac{1}{2n}\,\sin(2nx)\right] & k+n \text{ even and } k \geq n, \\ \dfrac{1}{2^{k+1}} \displaystyle\sum_{j=0}^{k} \binom{k}{j}\left[\frac{\sin\big[(2j-k+n)x\big]}{2j-k+n} + \frac{\sin\big[(2j-k-n)x\big]}{2j-k-n}\right] & \text{otherwise.} \end{cases}$$

Proof. We can simplify $\cos^k(\theta) \in \mathbb{R}$ by rewriting in terms of exponentials, applying the binomial theorem and taking the real part:

$$\cos^k(x) \;=\; \left(\frac{e^{ix}+e^{-ix}}{2}\right)^k \;=\; \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\left[e^{ix}\right]^k\left[e^{-ix}\right]^{k-j} \;=\; \mathrm{Re}\left(\frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}e^{i(2j-k)x}\right) \tag{14.63}$$

$$=\; \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\cos\big[(2j-k)x\big]. \tag{14.64}$$

so the integral becomes

$$I \;=\; \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\int \cos\big[(2j-k)x\big]\,\cos(nx)\,dx \tag{14.65}$$

$$=\; \frac{1}{2^k}\sum_{\substack{j=0 \\ |2j-k|\neq n}}^{k}\binom{k}{j}\frac{1}{2}\left[\frac{\sin\big[(2j-k+n)x\big]}{2j-k+n} + \frac{\sin\big[(2j-k-n)x\big]}{2j-k-n}\right]$$

$$+\,\frac{1}{2^k}\binom{k}{\frac{k+n}{2}}\left[\frac{x}{2} + \frac{1}{4n}\,\sin\big[2nx\big]\right] + \frac{1}{2^k}\binom{k}{\frac{k-n}{2}}\left[\frac{x}{2} + \frac{1}{4(-n)}\,\sin\big[2(-n)x\big]\right], \tag{14.66}$$

where we used Lemma 14.8 and allowed the sum over j to contain terms with $|2j-k| = n$, i.e. $j = (k\pm n)/2$. The condition in the sum simplifies to $j \neq (k \pm n)/2$. Depending on the values of k, n, there are several possibilities:

1. $k+n$ is even, and $k \geq n$. In this case, the sum over j will hit both $(k+n)/2$ and $(k-n)/2$, so both these extra terms need to be considered. Notice that the binomial coefficients are equal,

$$\binom{k}{\frac{k+n}{2}} \;=\; \frac{k!}{\big(\frac{k+n}{2}\big)!\,\big(k-\big[\frac{k+n}{2}\big]\big)!} \;=\; \frac{k!}{\big(\frac{k+n}{2}\big)!\,\big(\frac{k-n}{2}\big)!} \;=\; \binom{k}{\frac{k-n}{2}}, \tag{14.67}$$

and the minus signs in $\sin(-x)/(-n)$ cancel, so the two terms are equal.

2. $k+n$ is even, and $k<n$. In this case, the lower value is $(k-n)/2<0$ so is never hit. The upper value is $(k+n)/2\in(k,n)$, but since $j\in[0,k]$, this is never hit either. Hence, neither extra term is included.

3. $k+n$ is odd. In this case, $(k+n)/2\notin\mathbb{Z}$, so neither extra term is included.

4. $k=0$ and $n\neq 0$. In this case, the sum over j has one term:

$$\frac{1}{2^{0+1}}\binom{0}{0}\left[\frac{\sin\left[(2\cdot 0-0+n)x\right]}{2\cdot 0-0+n}+\frac{\sin\left[(2\cdot 0-0-n)x\right]}{2\cdot 0-0-n}\right] \;=\; \frac{1}{2}\cdot 1\cdot\left[\frac{\sin(nx)}{n}+\frac{\sin(-nx)}{-n}\right] \;=\; \frac{\sin(nx)}{n}, \tag{14.68}$$

and $|2j-k|=0\neq n$, so the extra terms are never hit. This result agrees with the expected value

$$\int\cos(ax+b)\,dx \;=\; \frac{1}{a}\sin(ax+b)+c. \tag{14.69}$$

It is consistent with cases 1, 2 and 3 above.

5. $n=0$ and $k\neq 0$. In this case, we recover the integral

$$\int\cos^n(x)\,dx \;=\; \frac{1}{n}\cos^{n-1}(x)\,\sin(x)+\frac{n-1}{n}\int\cos^{n-2}(x)\,dx, \tag{14.70}$$

which is easily proven by writing the integrand as $\cos^{n-1}(x)\,\cos(x)\,dx$ and integrating by parts with $u(x)=\cos^{n-1}(x)$ and $dv=\cos(x)\,dx$. We have not checked whether the final result is consistent with any of the cases above.

6. $k=n=0$. In this case, the sum over j would only contain one term, but this is the term with $|2j-k|=0=\pm n$, so is excluded. The expression for the two extra terms is invalid due to the $1/n$ factor. We know that the integral reduces to $\int dx=x$, and we see that this is not compatible with any of the cases above.

□

Lemma 14.10. $\forall\, n \in \mathbb{N}_0$,

$$\int_0^\pi \cos^k(x)\cos(nx)\,dx \;=\; \begin{cases} \dfrac{\pi}{2^k}\dbinom{k}{\frac{k-n}{2}} & k+n \text{ even and } k \geq n, \\ 0 & \text{otherwise.} \end{cases} \tag{14.71}$$

Note that this definite integral holds for $n = 0$, even though the indefinite integral in Lemma 14.9 does not.

Proof. The case $n \neq 0$ follows immediately from Lemma 14.9 and that $\sin(m\pi) = 0\ \forall\, m \in \mathbb{Z}$. We prove the case $n = 0$ by induction:

$$\int_0^\pi \cos^k(x)\,dx \;=\; \begin{cases} \frac{\pi}{2^k}\binom{k}{\frac{k}{2}} & k+2 \text{ even and } k \geq n, \\ 0 & \text{otherwise.} \end{cases} \tag{14.72}$$

We require two base cases. For $k = 0$,

$$\text{LHS} \;=\; \int_0^\pi \cos^0(x)\,dx \;=\; \int_0^\pi dx \;=\; \big[x\big]_0^\pi \;=\; \pi, \tag{14.73}$$

and

$$\text{RHS} \;=\; \frac{\pi}{2^0}\binom{0}{\frac{0}{2}} \;=\; \pi. \tag{14.74}$$

For $k = 1$,

$$\int_0^\pi \cos^1(x)\,dx \;=\; \big[\sin(x)\big]_0^\pi \;=\; 0. \tag{14.75}$$

Suppose that the relation holds for k. Then, for $k+2$, the integral I_{k+2} is

$$I_{k+2} \;=\; \int_0^\pi \cos^{k+2}(x)\,dx \;=\; \int_0^\pi \cos^k(x)\,\cos(x)\,dx. \tag{14.76}$$

Integrate by parts with $u = \cos^{k+1}(x)$ so $du = -(k+1)\,\cos^k(x)\,\sin(x)\,dx$ and $dv = \cos(x)$ so $v = \sin(x)$:

$$I_{k+2} \;=\; \Big[\cos^{k+1}k(x)\,\sin(x)\Big]_0^\pi + (k+1)\int_0^\pi \cos^k(x)\,\sin^2(x)\,dx. \tag{14.77}$$

The boundary term is zero. In the integral on the RHS, write $\sin^2(x) = 1 - \cos^2(x)$ and multiply out the brackets:

$$I_{k+2} \;=\; (k+1)\int_0^\pi \cos^k(x)\,dx - (k+1)\int_0^\pi \cos^{k+2}(x)\,dx \;=\; (k+1)\,I_k - (k+1)\,I_{k+2}, \tag{14.78}$$

which rearranges to

$$I_{k+2} \;=\; \frac{k+1}{k+2}\,I_k. \tag{14.79}$$

Now, if the induction hypothesis is correct, we also have

$$I_k \;=\; \frac{\pi}{2^k}\binom{k}{\frac{k}{2}} \;=\; \frac{\pi}{2^k}\,\frac{k!}{(\frac{k}{2})!\,(k-\frac{k}{2})!} \;=\; \frac{\pi}{2^k}\,\frac{k!}{(\frac{k}{2})!^2}, \tag{14.80}$$

so that

$$I_{k+2} \;=\; \frac{\pi}{2^{k+2}}\binom{k+2}{\frac{k+2}{2}} \;=\; \frac{1}{4}\,\frac{\pi}{2^k}\,\frac{(k+2)!}{(\frac{k+2}{2})!2} \;=\; \frac{1}{4}\,\frac{\pi}{2^k}\,\frac{(k+2)!}{(\frac{k}{2}+1)!^2} \;=\; \frac{1}{4}\,\frac{\pi}{2^k}\,\frac{(k+2)(k+1)k!}{\left[(\frac{k}{2}+1)(\frac{k}{2})\right]^2} \tag{14.81}$$

$$=\; \frac{1}{4}\,\frac{\pi}{2^k}\,\frac{(k+2)(k+1)}{(\frac{k+2}{2})^2}\,\frac{k!}{(\frac{k}{2})!^2} \;=\; \frac{1}{4}\,\frac{\pi}{2^k}\,4\,\frac{k+1}{k+2}\binom{k}{\frac{k}{2}} \;=\; \frac{k+1}{k+2}\,I_k. \tag{14.82}$$

Then, Eq. (14.79) showed that the case k implies the case $k+2$. Coupled with our two consecutive bases cases, this completes the proof. □

Lemma 14.11. $\forall\, n \in \mathbb{N}_0$,

$$\int_0^\pi \cos(nx)\,dx \;=\; \delta_{n,0}. \tag{14.83}$$

Proof. For $n = 0$, we have

$$\int_0^\pi \cos(0)\,dx \;=\; \int_0^\pi dx \;=\; \pi. \tag{14.84}$$

For $n \neq 0$, we have

$$\int_0^\pi \cos(nx)\,dx \;=\; \Big[\frac{1}{n}\,\sin(nx)\Big]_0^\pi \;=\; 0. \tag{14.85}$$

Combining these, the proof is complete. Alternatively, we could have recognised that $\cos(n\theta) = T_n\big[\cos(\theta)\big]$ and used the orthgonality relation for Chebyshev polynomials of the first kind, Eq. (13.32), with $T_0\big[\cos(\theta)\big] = 1$:

$$\int_0^\pi \cos(n\theta)\,d\theta \;=\; \int_0^\pi T_n\big[\cos(\theta)\big]\,d\theta \;=\; \int_0^\pi T_n\big[\cos(\theta)\big]\,T_0\big[\cos(\theta)\big]\,d\theta \;=\; \pi\,\delta_{n,0}. \tag{14.86}$$

□

Example 14.12. *The Chebyshev series coefficients of $f(x) = x^2$ on an arbitrary interval $[A, B]$ are*

$$a_0 = \frac{1}{8}\big(3\,B^2 + 3\,A^2 + 2\,A\,B\big), \tag{14.87}$$

$$a_1 = \frac{1}{2}\big(B + A\big)\big(B - A\big), \tag{14.88}$$

$$a_2 = \frac{1}{8}\big(B - A\big)^2, \tag{14.89}$$

with $a_n = 0$ for $n > 2$.

Proof. By Proposition 13.8, we have

$$f\Big(\Lambda_{A,B}\big[\cos(\theta)\big]\Big) = \frac{1}{4}\big(B - A\big)^2\,\cos^2(\theta) + \frac{1}{2}\big(B - A\big)\big(B + A\big)\,\cos(\theta) + \frac{1}{4}\big(B + A\big)^2, \tag{14.90}$$

so Eq. (14.45) gives

$$\begin{aligned} a_n = {} & \frac{h_n}{2\pi}\big(B - A\big)^2 \int_0^\pi \cos^2(\theta)\,\cos(n\theta)\,d\theta + \frac{h_n}{\pi}\big(B - A\big)\big(B + A\big)\int_0^\pi \cos(\theta)\,\cos(n\theta)\,d\theta \\ & + \frac{h_n}{2\pi}\big(B + A\big)^2 \int_0^\pi \cos(n\theta)\,d\theta. \end{aligned} \tag{14.91}$$

Lemma 14.10 yields

$$\int_0^\pi \cos^2(\theta)\,\cos(n\theta)\,d\theta = \frac{\pi}{2}\,\delta_{n,0} + \frac{\pi}{4}\,\delta_{n,2}, \tag{14.92}$$

$$\int_0^\pi \cos(\theta)\,\cos(n\theta)\,d\theta = \frac{\pi}{2}\,\delta_{n,1}. \tag{14.93}$$

Substituting these plus Lemma 14.11 and using the expression for h_n in Eq. (14.4), we obtain the given result. □

Proposition 14.13 (First derivative of Chebyshev series on arbitrary interval)**.** *Let $f(x)$ be a function defined on $[A, B]$ whose Chebyshev series has been found on this interval. Then,*

$$\frac{df}{dx} = \frac{2}{B - A}\sum_{n=1}^{\infty} a_n\,n\,U_{n-1}\big[\Lambda_{A,B}^{-1}(x)\big] \tag{14.94}$$

$$= \sum_{n=0}^{\infty}\left(\frac{4}{B - A}\,h_n \sum_{\substack{k=n+1 \\ n+k\ \text{odd}}}^{\infty} k\,a_k\right) T_n\big[\Lambda_{A,B}^{-1}(x)\big]. \tag{14.95}$$

Proof. Differentiate the expression in Proposition 14.6 term-by-term, using the result of Proposition 13.12. □

Proposition 14.14 (Second derivative of Chebyshev series on arbitrary interval)**.** *Let $f(x)$ be a function defined on $[A, B]$ whose Chebyshev series has been found on this interval. Then,*

$$\frac{d^2 f}{dx^2} = \sum_{n=0}^{\infty}\left(\frac{4}{(B - A)^2}\,h_n \sum_{\substack{k=n+2 \\ n+k\ \text{even}}}^{\infty} k(k^2 - n^2)\,a_k\right) T_n\big[\Lambda_{A,B}^{-1}(x)\big]. \tag{14.96}$$

Proof. Two applications of the chain rule yield

$$\frac{d^2 T_n}{dx^2} = \frac{d^2 T_n}{d(\Lambda_{A,B}^{-1})^2}\left(\frac{d\Lambda_{A,B}^{-1}}{dx}\right)^2 + \frac{dT_n}{d\Lambda_{A,B}^{-1}}\,\frac{d^2(\Lambda_{A,B}^{-1})}{dx^2} = \frac{4}{(B - A)^2}\,\frac{d^2 T_n}{d(\Lambda_{A,B}^{-1})^2}. \tag{14.97}$$

The rest follows in the same way as for $[-1, 1]$. □

15 Legendre polynomials

The **spherical harmonics** (Definition 17.1) form a complete orthonormal basis over the θ and ϕ directions, which means that a general two-dimensional angular function $f(\theta, \phi)$ can be expressed exactly as a linear combination of them. This makes approximation of such a function straightforward. Making a simple extension to the **vector spherical harmonics** (Definition 18.1), and using the previously mentioned 1D methods to handle radial dependence, it is possible to model the behaviour of any vector function in spherical coordinates. Further, vector spherical harmonics have straightforward behaviour under divergence and curl operations, facilitating numerical vector calculus.

Over the next few chapters, we build up to a method of expressing the angular part of a vector function in terms of vector spherical harmonics. We begin from their one-dimensional counterparts, the Legendre polynomials and associated Legendre functions, since many intermediary results will prove useful.

Ultimately, our project will not extend beyond axisymmetry, so angular dependence can be modelled using only Legendre polynomials and associated Legendre functions. The chapters on spherical harmonics, vector spherical harmonics may be skipped, but we retain them because they represent original work that will prove useful should the project be expanded to 3D in the future, or for other applications.

The statements in this chapter are true $\forall\, x \in [-1, 1]$ and $\forall\, \ell \in \mathbb{N}_0$, unless otherwise stated.

15.1 Definition and examples

Definition 15.1. *Let $x \in [-1, 1]$ and $\ell \in \mathbb{N}_0$. Then,* **Legendre's differential equation** *states that*

$$(1 - x^2)\,\frac{d^2 P_\ell}{dx^2} - 2x\,\frac{dP_\ell}{dx} + \ell(\ell + 1)\,P_\ell(x) \;=\; 0. \tag{15.1}$$

The **Legendre polynomials** *are the unique polynomials $P_\ell : [-1, 1] \to \mathbb{R}$ with degree ℓ which are solutions to this equation.*

The first few Legendre polynomials are

$$P_0(x) \;=\; 1, \tag{15.2}$$
$$P_1(x) \;=\; x, \tag{15.3}$$
$$P_2(x) \;=\; \frac{1}{2}\big(3x^2 - 1\big), \tag{15.4}$$
$$P_3(x) \;=\; \frac{1}{2}\big(5x^3 - 3x\big), \tag{15.5}$$
$$P_4(x) \;=\; \frac{1}{8}\big(35x^4 - 30x^2 + 3\big). \tag{15.6}$$

15.2 Properties and relations

The values at the endpoints are (e.g. Arfken & Weber, 2005, §12.2)

$$P_\ell(-1) \;=\; P_\ell\big[\cos((2\ell + 1)\pi)\big] \;=\; (-1)^\ell, \tag{15.7}$$
$$P_\ell(1) \;=\; P_\ell\big[\cos(2\ell\pi)\big] \;=\; 1. \tag{15.8}$$

For negative values, we have

$$P_\ell(-x) \;=\; (-1)^\ell\, P_\ell(x). \tag{15.9}$$

That is, the Legendre polynomials of even order are even, and those of odd order are odd.

The Legendre polynomials follow **Bonnet's recursion relation**. $\forall\, \ell \in \mathbb{N} : \ell > 1$,

$$P_\ell(x) \;=\; \frac{1}{\ell}\Big[(2\ell-1)\,x\,P_{\ell-1}(x) - (\ell-1)\,P_{\ell-2}(x)\Big], \tag{15.10}$$

This gives a way to generate all the Legendre polynomials up to some cutoff $\ell_{\max}$, evaluated at a single point x, given two "seed" values $P_0(x)$ and $P_1(x)$. We apply this relation in §23.3.

15.3 Derivatives

The first derivatives of the Legendre polynomials are best expressed in terms of the associated Legendre functions (Proposition 16.41), but an expression staying with Legendre polynomials does exist:

$$\frac{d}{dx}P_\ell(x) \;=\; \begin{cases} \dfrac{\ell}{1-x^2}\Big[-x\,P_\ell(x) + P_{\ell-1}(x)\Big] & |x| \neq 1, \\[2ex] \dfrac{\ell(\ell+1)}{2} & x = 1, \\[2ex] (-1)^{\ell+1}\,\dfrac{\ell(\ell+1)}{2} & x = -1. \end{cases} \tag{15.11}$$

The expression for $|x| \neq 1$ can be found in e.g. Eq. (12.26) of Arfken & Weber (2005). It breaks down at the endpoints due to the division by $1 - x^2$; we add those expressions as separate cases.

Proposition 15.2. *The first derivative in terms of θ is*

$$\frac{d}{d\theta}P_\ell\big[\cos(\theta)\big] \;=\; \frac{\ell}{\sin(\theta)}\Big[\cos(\theta)\,P_\ell\big[\cos(\theta)\big] - P_{\ell-1}\big[\cos(\theta)\big]\Big]. \tag{15.12}$$

In particular, $\frac{d}{d\theta}P_\ell\big[\cos(\theta)\big] = 0$ *if* $\theta \in \{0, \pi\}$.

Proof. For $\theta \in (0,\pi)$, the chain rule gives $\frac{d}{d\theta}P_\ell\big[\cos(\theta)\big] = \frac{dP_\ell}{d\cos(\theta)}\frac{d\cos(\theta)}{d\theta} = -\sin(\theta)\frac{dP_\ell}{d\cos(\theta)} = -\sqrt{1-x^2}\,\frac{dP_\ell}{dx}$. Substituting the expression for $\frac{dP_\ell}{dx}$ gives the result in terms of θ. We claim that the expression also holds in the limit as $\theta \to 0$ and $\theta \to 2\pi$, and that it yields zero in both limits. □

It may be surprising to see that $\frac{dP_\ell}{dx}$ cannot be expressed purely in terms of Legendre polynomials, especially since a well-known result is that they form a complete orthonormal basis such that any general function can be expressed on $(-1, 1)$ as an infinite sum of Legendre polynomials multiplied by constant coefficients. Since $\frac{dP_\ell}{dx}$ is clearly a function, it should then be expressable as a sum of Legendre polynomials. However, for this case it can only be produced by *infinitely many* such polynomials.

One way to explain this is that the Legendre polynomials are polynomials in x, but x is a trigonometric function of θ, so taking a derivative introduces an extra function of θ such that the result is no longer a polynomial in x. If it is not a polynomial, we cannot expect finitely many Legendre polynomials to reproduce it exactly.

We will face this issue with all of the basis functions that we introduce in the proceeding chapters. It is unfortunate because our evolutionary equations will require us to take spatial derivatives in the radial and polar directions (r, θ), so it would save much time and computational expense if we found an immediate way to obtain derivatives of functions whose series expansions we had already calculated. We will instead search for expressions for the derivatives that are as close to being straightforward as possible.

15.4 Integrals

The Legendre polynomials obey the orthogonality relation

$$\int_{-1}^{1} P_\ell(x)\, P_m(x)\, dx \;=\; \int_0^{\pi} P_\ell\big[\cos(\theta)\big]\, P_m\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \;=\; \frac{2}{2\ell+1}\,\delta_{\ell,m}, \tag{15.13}$$

The expression in terms of θ is given by a change of variables $x = \cos(\theta)$.

A useful expression is that (e.g. Arfken & Weber, 2005, §12.2)

$$(2\ell+1)\, P_\ell(x) \;=\; \frac{d}{dx}\big[P_{\ell+1}(x) - P_{\ell-1}(x)\big], \tag{15.14}$$

from which it immediately follows that we can express the integral of a Legendre polynomial in terms of two other Legendre polynomials. $\forall\, a, b \in \mathbb{R} : a, b \in [-1, 1]$ and $\forall\, \ell \in \mathbb{N}$,

$$\int_a^b P_\ell(x)\, dx \;=\; \frac{1}{2\ell+1}\big[P_{\ell+1}(x) - P_{\ell-1}(x)\big]_a^b. \tag{15.15}$$

Some authors relax $\ell \in \mathbb{N}_0$ to $\ell \in \mathbb{Z}$ for easier application of recursion relations. Some define $P_\ell(x) \equiv 0$ for $\ell < 0$, but it is also common to define $P_{-1}(x) = -1$ (e.g. Jackson, 1999, Problem 3.2 (b)), which extends the validity of Eq. (15.15) to $\ell \in \mathbb{N}_0$. We will not consider $\ell < 0$ in this project, but note that its handling may be important in numerical applications.

Corollary 15.3 (Integral over domain)**.**

$$\int_{-1}^{1} P_\ell(x)\, dx \;=\; \int_0^{\pi} P_\ell\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \;=\; 2\,\delta_{\ell,0}. \tag{15.16}$$

Proof. Using the orthogonality relation Eq. (15.13) and that $1 = P_0(x)$, we have

$$\int_{-1}^{1} P_\ell(x)\, dx \;=\; \int_{-1}^{1} P_\ell(x)\, P_0(x) \;=\; \frac{2}{2\ell+1}\,\delta_{\ell,0}. \tag{15.17}$$

If $\ell \neq 0$, this is zero. If $\ell = 0$, it evaluates to $\frac{2}{2(0)+1} = 2$.
Alternatively, we can use Eq. (15.15) with Eqs. (15.7) and (15.8). For $\ell > 0$,

$$\int_{-1}^{1} P_\ell(x)\, dx \;=\; \frac{1}{2\ell+1}\Big[1 - 1\Big] - \frac{1}{\ell+1}\Big[(-1)^{2\ell+1} - (-1)^{\ell-1}\Big] \;=\; 0 - \frac{(-1)^\ell}{2\ell+1}\Big[-1 - \frac{1}{-1}\Big] \;=\; 0. \tag{15.18}$$

For $\ell = 0$, use that $P_{-1}(x) = -1$ to give

$$\int_{-1}^{1} P_0(x)\, dx \;=\; \frac{1}{2\ell+1}\Big[1 - (-1)\Big] - \frac{1}{2\ell+1}\Big[(-1)^1 - (-1)\Big] \;=\; 2. \tag{15.19}$$

□

16 Associated Legendre functions

16.1 Definition and examples

Definition 16.1. *Let $\ell \in \mathbb{N}_0$ and $m \in \mathbb{Z} : |m| \leq \ell$. The* **associated Legendre functions (ALFs)** *are defined by*

$$P_\ell^m(x) \;=\; (-1)^m \,(1-x^2)^{m/2}\, \frac{d^m}{dx^m}\, P_\ell(x). \tag{16.1}$$

The associated Legendre functions are often referred to as the **associated Legendre polynomials**, but we avoid this name since they are not polynomials if m is odd. The factor $(-1)^m$ is known as the **Condon-Shortley phase** and is omitted in some definitions; we employ it throughout this text. Notice that $P_\ell^0(x) = P_\ell(x)$, that is, the Legendre polynomials are special cases of the ALFs with $m = 0$. In this text, we define $P_\ell^m \equiv 0$ if $m > \ell$, which is useful for considering derivatives or recursion relations.

The first few ALFs are as follows. For $\ell = 1$,

$$P_1^{-1}(x) \;=\; \frac{1}{2}\left(1-x^2\right)^{1/2}, \tag{16.2}$$

$$P_1^{1}(x) \;=\; -\left(1-x^2\right)^{1/2}. \tag{16.3}$$

For $\ell = 2$,

$$P_2^{-2}(x) \;=\; \frac{1}{8}\left(1-x^2\right), \tag{16.4}$$

$$P_2^{-1}(x) \;=\; \frac{1}{2}\,x\left(1-x^2\right)^{1/2}, \tag{16.5}$$

$$P_2^{1}(x) \;=\; -3x\left(1-x^2\right)^{1/2}, \tag{16.6}$$

$$P_2^{2}(x) \;=\; 3\left(1-x^2\right)^{1/2}. \tag{16.7}$$

For $\ell = 3$,

$$P_3^{-3}(x) \;=\; \frac{1}{48}\left(1-x^2\right)^{3/2}, \tag{16.8}$$

$$P_3^{-2}(x) \;=\; \frac{1}{8}\,x\left(1-x^2\right), \tag{16.9}$$

$$P_3^{-1}(x) \;=\; -\frac{1}{8}\left(1-5x^2\right)\left(1-x^2\right)^{1/2}, \tag{16.10}$$

$$P_3^{1}(x) \;=\; \frac{3}{2}\left(1-5x^2\right)\left(1-x^2\right)^{1/2}, \tag{16.11}$$

$$P_3^{2}(x) \;=\; 15\,x\left(1-x^2\right), \tag{16.12}$$

$$P_3^{3}(x) \;=\; -15\left(1-x^2\right)^{3/2}. \tag{16.13}$$

16.2 Properties and relations

The ALFs follow the **generalised Bonnet recursion relation**. $\forall\,\ell \in \mathbb{N} : \ell > 1$ and $\forall\, m \in \mathbb{Z} : |m| < \ell$,

$$P_\ell^m(x) \;=\; \frac{1}{\ell - m}\Big[(2\ell-1)\,x\,P_{\ell-1}^m(x) - (\ell+m-1)\,P_{\ell-2}^m(x)\Big]. \tag{16.14}$$

This gives a way to generate all the ALFs for fixed m evaluated at x, given two "seed" values $P_m^m(x)$ and $P_{m+1}^m(x)$. It is most useful when we will only ever need the ALFs with one value of m; in our code, we only require those for $m = 1$ because they represent the derivatives of the Legendre polynomials.

Lemma 16.2. $\forall\,\ell \in \mathbb{N}$,

$$\frac{d^{\ell-1}}{dx^{\ell-1}}\,P_\ell(x) \;=\; (2\ell-1)!!\,x, \tag{16.15}$$

$$\frac{d^{\ell}}{dx^{\ell}}\,P_\ell(x) \;=\; (2\ell-1)!!. \tag{16.16}$$

Proof. We prove the $(\ell-1)$th derivative by induction. The result includes a double factorial and we will increment by 2, not 1, in our proof, so let us consider the first two base cases. For $\ell = 1$,

$$\frac{d^{1-1}}{dx^{1-1}}\,P_1(x) \;=\; P_1(x) \;=\; x, \tag{16.17}$$

$$\big[2(1)-1\big]!!\,x \;=\; (2-1)!!\,x \;=\; 1!!\,x \;=\; x. \tag{16.18}$$

For $\ell = 2$,

$$\frac{d^{2-1}}{dx^{2-1}}\,P_2(x) \;=\; \frac{d}{dx}\,\frac{1}{2}\left(3x^2-1\right) \;=\; 3x, \tag{16.19}$$

$$\big[2(2)-1\big]!!\,x \;=\; (4-1)!!\,x \;=\; 3!!\,x \;=\; 3x. \tag{16.20}$$

Suppose that the result holds for ℓ. Then, for $\ell + 2$,

$$\frac{d^{\ell+1}}{dx^{\ell+1}}\,P_{\ell+1}(x) \;=\; \frac{d^{\ell+1}}{dx^{\ell+1}}\,\frac{1}{\ell+2}\Big[\big[2(\ell+2)-1\big]\,x\,P_{\ell+1}(x) - (\ell+1)\,P_\ell(x)\Big] \tag{16.21}$$

$$\;=\; \frac{2\ell+3}{\ell+2}\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,x\,P_{\ell+1}(x) - \frac{\ell+1}{\ell+2}\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,P_\ell(x), \tag{16.22}$$

where we used Bonnet's recursion relation Eq. (15.10). The second term is zero by the induction hypothesis:

$$\frac{d^{\ell+1}}{dx^{\ell+1}}\,P_\ell(x) \;=\; \frac{d^2}{dx^2}\,\frac{dx^{\ell-1}}{dx^{\ell-1}}\,P_\ell(x) \;=\; \frac{d^2}{dx^2}\,(2\ell-1)!!\,x \;=\; 0. \tag{16.23}$$

For the first term, the product rule for the nth derivative generalises to Leibniz' rule,

$$\frac{d^n}{dx^n}\,A(x)\,B(x) \;=\; \sum_{k=0}^{n}\binom{n}{k}\,\frac{d^k}{dx^k}\,A(x)\,\frac{d^{n-k}}{dx^{n-k}}\,B(x), \tag{16.24}$$

so in particular

$$\frac{d^n}{dx^n}\,x\,A(x) \;=\; \binom{n}{0}\,\frac{d^0}{dx^0}\,x\,\frac{d^{n-0}}{dx^{n-0}}\,A(x) + \binom{n}{1}\,\frac{d^1}{dx^1}\,x\,\frac{d^{n-1}}{dx^{n-1}}\,A(x) + \sum_{n=2}^{n}\binom{n}{k}\,\frac{d^k}{dx^k}\,x\,\frac{d^{n-k}}{dx^{n-k}}\,A(x) \tag{16.25}$$

$$\;=\; x\,\frac{d^n}{dx^n}\,A(x) + n\,\frac{d^{n-1}}{dx^{n-1}}\,A(x). \tag{16.26}$$

Then,

$$\frac{2\ell+3}{\ell+2}\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,x\,P_{\ell+1}(x) \;=\; \frac{2\ell+3}{\ell+2}\Big[\,x\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,P_{\ell+1}(x) + (\ell+1)\,\frac{d^{\ell}}{dx^{\ell}}\,P_{\ell+1}(x)\Big]. \tag{16.27}$$

We have

$$x\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,P_{\ell+1}(x) \;=\; x\,\frac{d}{dx}\,\frac{d^{\ell}}{dx^{\ell}}\,P_{\ell+1}(x) \;=\; x\,\frac{d}{dx}\,\big[2(\ell+1)-1\big]!!\,x \;=\; (2\ell+1)!!\,x, \tag{16.28}$$

$$(\ell+1)\,\frac{d^{\ell}}{dx^{\ell}}\,P_{\ell+1}(x) \;=\; (\ell+1)\,\big[2(\ell+1)-1\big]!!\,x \;=\; (\ell+1)\,(2\ell+1)!!\,x, \tag{16.29}$$

giving

$$\frac{2\ell+3}{\ell+2}\,\frac{d^{\ell+1}}{dx^{\ell+1}}\,x\,P_{\ell+1}(x) \;=\; \frac{2\ell+3}{\ell+2}\,(1+\ell+1)\,(2\ell+1)!!\,x \;=\; (2\ell+3)\,(2\ell+1)!!\,x \tag{16.30}$$

$$\;=\; (2\ell+3)!!\,x \;=\; \big[2(\ell+2)-1\big]!!\,x, \tag{16.31}$$

which proves the $(\ell-1)$th derivative. The ℓth derivative follows immediately by differentiation; it can also be proven on its own using the same method. □

Proposition 16.3. $\forall\,\ell\in\mathbb{N}$,

$$P_\ell^\ell(x) \;=\; (-1)^\ell\,(2\ell-1)!!\,(1-x^2)^{\ell/2}, \tag{16.32}$$

$$P_{\ell+1}^\ell(x) \;=\; (2\ell+1)\,x\,P_\ell^\ell(x). \tag{16.33}$$

Proof. The expression for $P_\ell^\ell(x)$ follows immediately from Lemma 16.2 and Definition 16.1. For $P_{\ell+1}^\ell(x)$, we have

$$P_{\ell+1}^\ell(x) \;=\; (-1)^\ell\,(1-x^2)^{\ell/2}\,\frac{d^\ell}{dx^\ell}\,P_{\ell+1} \;=\; (-1)^\ell\,(1-x^2)^{\ell/2}\,\big[2(\ell+1)-1\big]!!\,x \tag{16.34}$$

$$\;=\; (-1)^\ell\,(1-x^2)^{\ell/2}\,(2\ell+1)!!\,x \;=\; (-1)^\ell\,(1-x^2)^{\ell/2}\,(2\ell+1)\,(2\ell-1)!!\,x \;=\; (2\ell+1)\,x\,P_\ell^\ell(x). \tag{16.35}$$

□

The ALFs for negative m are simply a rescaling of those with positive m. In particular, $\forall\,\ell\in\mathbb{N}_0,\ m\in\mathbb{N}_0 : m\le\ell$,

$$P_\ell^{-m}(x) \;=\; (-1)^m\,\frac{(\ell-m)!}{(\ell+m)!}\,P_\ell^m(x). \tag{16.36}$$

The values at the endpoints are, $\forall\,\ell\in\mathbb{N}_0,\ m\in\mathbb{Z} : |m|\le\ell$,

$$P_\ell^m(-1) \;=\; P_\ell^m\big[\cos(\pi)\big] \;=\; (-1)^\ell\,\delta_{m,0}, \tag{16.37}$$

$$P_\ell^m(1) \;=\; P_\ell^m\big[\cos(0)\big] \;=\; \delta_{m,0}. \tag{16.38}$$

That is, the functions are nonzero at the endpoints iff $m=0$.

16.3 Derivatives

As we touched on in Chapter 15, there is no simple expression for the derivative of an ALF which remains in polynomials. Perhaps it is easier to see for ALFs than Legendre polynomials, since ALFs were not polynomials to begin with. We will make do with a θ-derivative which is expressable as a sum of two ALFs with the same ℓ as the original, but differing m.

Proposition 16.4 (Derivatives of the Legendre polynomials)**.** $\forall\, x \in (-1,1),\ \ell \in \mathbb{N}_0,\ n \in \mathbb{N}$,

$$\frac{d^n}{dx^n} P_\ell(x) \;=\; \begin{cases} \frac{(-1)^n}{(1-x^2)^{n/2}} P_\ell^n(x) & n \le \ell, \\ 0 & n > \ell. \end{cases} \tag{16.39}$$

In particular,

$$\frac{dP_\ell}{dx} \;=\; \begin{cases} \frac{-1}{(1-x^2)^{1/2}} P_\ell^1(x) & \ell \ge 1, \\ 0 & \ell = 0. \end{cases} \tag{16.40}$$

In terms of θ,

$$\frac{dP_\ell}{d\theta} \;=\; P_\ell^1\big[\cos(\theta)\big]. \tag{16.41}$$

Proof. The expression in terms of x follows immediately from Definition 16.1, that $P_\ell^n(x) \equiv 0$ if $n > \ell$, and that $(-1)^{-n} = (-1)^n$. In terms of θ, we have

$$P_\ell^1(x) \;=\; (-1)^1 (1-x^2)^{1/2} \frac{d^1}{dx^1} P_\ell^0(x) \;=\; -(1-x^2)^{1/2} \frac{dP_\ell}{dx}. \tag{16.42}$$

Let $x = \cos(\theta)$. Then, for a function of one variable f, the chain rule gives

$$\frac{df}{d\theta} \;=\; \frac{df}{d\cos(\theta)} \frac{d\cos(\theta)}{d\theta} = \frac{df}{d\cos(\theta)} \cdot -\sin(\theta) = -\sqrt{1-x^2}\, \frac{df}{dx}, \tag{16.43}$$

where in the last step we used that $x = \cos(\theta)$. Setting $f(x) = P_\ell^1(x) = P_\ell^1\big[\cos(\theta)\big]$, the RHS of both equations are equal, so we can equate the LHS $P_\ell^1(x)$ and $\frac{df}{d\theta} = \frac{dP_\ell}{d\theta}$. □

Combining Eq. (16.41) with the recursion relation given in Eq. (12.94) of Arfken & Weber (2005), the first derivative of a general ALF is then

$$\frac{d}{d\theta} P_\ell^m\big[\cos(\theta)\big] \;=\; \frac{1}{2} P_\ell^{m+1} - \frac{1}{2} (\ell+m)(\ell-m+1)\, P_\ell^{m-1}. \tag{16.44}$$

This holds for $|m| = \ell$ with the definition that $P_\ell^m = 0$ for $|m| > \ell$.

16.4 Integrals

There is no general orthogonality condition for the ALFs, but the following two conditions hold. For constant m,

$$\int_{-1}^{1} P_\ell^m(x)\, P_{\ell'}^m(x)\, dx \;=\; \int_0^\pi P_\ell^m\big[\cos(\theta)\big]\, P_{\ell'}^m\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \;=\; \frac{2}{2\ell+1}\,\frac{(\ell+m)!}{(\ell-m)!}\,\delta_{\ell,\ell'}. \tag{16.45}$$

For constant ℓ,

$$\int_{-1}^{1} P_\ell^m(x)\, P_\ell^{m'}(x)\,\frac{1}{1-x^2}\, dx \;=\; \int_0^\pi P_\ell^m\big[\cos(\theta)\big]\, P_\ell^{m'}\big[\cos(\theta)\big]\,\frac{1}{\sin(\theta)}\, d\theta \;=\; \begin{cases} 0 & m \neq m', \\ \frac{1}{m}\,\frac{(\ell+m)!}{(\ell-m)!} & m = m' \neq 0, \\ \infty & m = m' = 0. \end{cases} \tag{16.46}$$

In particular, if it is <u>not</u> true that $m = m' = 0$, then

$$\int_{-1}^{1} P_\ell^m(x)\, P_\ell^{m'}(x)\,\frac{1}{1-x^2}\, dx \;=\; \int_0^\pi P_\ell^m\big[\cos(\theta)\big]\, P_\ell^{m'}\big[\cos(\theta)\big]\,\frac{1}{\sin(\theta)}\, d\theta \;=\; \frac{1}{m}\,\frac{(\ell+m)!}{(\ell-m)!}\,\delta_{m,m'}, \quad \neg\,(m = m' = 0). \tag{16.47}$$

16.5 Overlap integrals

The reader may skip this section.

The numerical method in our simulation will involve the evaluation of a very large number of integrals of products of ALFs. If we can avoid computing these integrals numerically and simply quote an exact result, this will aid the accuracy and long-term stability of the evolution, as well as improve execution times. We might even find that some terms are identically zero, which would enable us to remove these from our calculations.

Unfortunately, we found that while exact expressions for the integrals exist, they are extremely complicated. For completeness, or even as a way of justifying the choice to calculate the integrals numerically, we quote the results below. It is clear that we won't be able to use these to simplify any of our expressions. We also argue that a code to evaluate the right-hand side of these expressions may be even more computationally expensive than a simple numerical integration. It would increase the risk of typos or coding logic errors leading to spurious inaccuracies, which would be extremely hard to find given the added complexity of the written code.

As such, we do not pursue further the idea of evaluating exactly the integrals of products of ALFs.

Definition 16.5. *Let $\ell_1, \ell_2, \ell_3 \in \mathbb{N}_0$ and $m_1, m_2, m_3 \in \mathbb{Z}$. The* **Wigner $3j$ symbols** *are the real numbers*

$$\begin{aligned}\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \;=\; \delta_{m_1+m_2+m_3,0}\,(-1)^{\ell_1-\ell_2-m_3}\sqrt{\frac{(\ell_1+\ell_2-\ell_3)!\,(\ell_1-\ell_2+\ell_3)!\,(-\ell_1+\ell_2+\ell_3)!}{(\ell_1+\ell_2+\ell_3+1)!}} \\ \cdot\sqrt{(\ell_1-m_1)!\,(\ell_1+m_1)!\,(\ell_2-m_2)!\,(\ell_2+m_2)!\,(\ell_3-m_3)!\,(\ell_3+m_3)!} \\ \cdot\sum_{k=k_{\min}}^{k_{\max}} \frac{(-1)^k}{k!\,(\ell_1+\ell_2-\ell_3-k)!\,(\ell_1-m_1-k)!\,(\ell_2+m_2-k)!\,(\ell_3-\ell_2+m_1+k)!\,(\ell_3-\ell_1-m_2+k)!},\end{aligned} \tag{16.48}$$

where $k_{\min} = \max\{\,0\,,\,\ell_2-\ell_3-m_1\,,\,\ell_1-\ell_3+m_2\,\}$ and $k_{\max} = \min\{\,\ell_1+\ell_2-\ell_3\,,\,\ell_1-m_1\,,\,\ell_2+m_2\,\}$.

One must be careful with the notation used: the Wigner $3j$ symbols are real numbers and not matrices. They are commonly defined with $j_1, j_2, j_3 \in \mathbb{N}_0$ occupying the first row, hence the name; however, our application concerns ℓ. Note the following special case, which applies to the Legendre polynomialss:

$$\begin{pmatrix} \ell_1 & \ell_2 & \ell_3 \\ 0 & 0 & 0 \end{pmatrix} = (-1)^{\ell_1-\ell_2} \sqrt{\frac{(\ell_1+\ell_2-\ell_3)!\,(\ell_1-\ell_2+\ell_3)!\,(-\ell_1+\ell_2+\ell_3)!}{(\ell_1+\ell_2+\ell_3+1)!}}\,\ell_1!\,\ell_2!\,\ell_3! \cdot \sum_{k=k_{\min}}^{k_{\max}} \frac{(-1)^k}{k!\,(\ell_1+\ell_2-\ell_3-k)!\,(\ell_1-k)!\,(\ell_2-k)!\,(\ell_3-\ell_2+k)!\,(\ell_3-\ell_1+k)!}, \tag{16.49}$$

where $k_{\min} = \max\{\,0\,,\,\ell_2-\ell_3\,,\,\ell_1-\ell_3\,\}$ and $k_{\max} = \min\{\,\ell_1+\ell_2-\ell_3\,,\,\ell_1\,,\,\ell_2\,\}$.

§2 of Dong & Lemus (2002) gives the integral of the product of arbitrarily many ALFs over all space; we quote the results for two and three functions below. The expressions are only valid for nonnegative orders m_i; expressions for cases with any number of negative orders are easily obtained by Eq. (16.36) but are not of use to us.

The overlap integral of two ALFs with $m_1, m_2 \geq 0$ is

$$\int_0^\pi P_{\ell_1}^{m_1} P_{\ell_2}^{m_2} \sin(\theta)\,d\theta = \sqrt{\frac{(\ell_1+m_1)!}{(\ell_1-m_1)!}\frac{(\ell_2+m_2)!}{(\ell_2-m_2)!}} \sum_{\ell_{12}=\max\{|\ell_1-\ell_2|,m_{12}\}}^{\ell_1+\ell_2} G_{12} \sqrt{\frac{(\ell_{12}-m_{12})!}{(\ell_{12}+m_{12})!}} \cdot \left[(-1)^{\ell_{12}} + (-1)^{m_{12}}\right] 2^{m_{12}-2}\, m_{12}\, \frac{\Gamma\!\left(\frac{\ell_{12}}{2}\right)\Gamma\!\left(\frac{\ell_{12}+m_{12}+1}{2}\right)}{\Gamma\!\left(\frac{\ell_{12}+3}{2}\right)\Gamma\!\left(\frac{\ell_{12}-m_{12}+2}{2}\right)}, \tag{16.50}$$

where

$$G_{12} = (-1)^{m_{12}}\,(2\ell_{12}+1) \begin{pmatrix} \ell_1 & \ell_2 & \ell_{12} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell_1 & \ell_2 & \ell_{12} \\ m_1 & m_2 & -m_{12} \end{pmatrix} \tag{16.51}$$

and where ℓ_{12} is a summation index and $m_{12} \equiv m_1 + m_2$. This expression is slightly modified from the one given by the authors, in order to automatically satisfy the following two requirements:

1. $\ell_{12} \geq m_{12}$; accounted for by modifying the lower bound on ℓ_{12} from the $|\ell_1 - \ell_2|$ written in the paper.
2. $\ell_1 + \ell_2 + \ell_{12}$ is even; automatically accounted for because $\begin{pmatrix} \ell_1 & \ell_2 & \ell_{12} \\ 0 & 0 & 0 \end{pmatrix} = 0$ otherwise.

The overlap integral of three ALFs with $m_1, m_2, m_3 \geq 0$ is

$$\int_0^\pi P_{\ell_1}^{m_1} P_{\ell_2}^{m_2} P_{\ell_3}^{m_3} \sin(\theta)\,d\theta = \sqrt{\frac{(\ell_1+m_1)!}{(\ell_1-m_1)!}\frac{(\ell_2+m_2)!}{(\ell_2-m_2)!}\frac{(\ell_3+m_3)!}{(\ell_3-m_3)!}} \sum_{\ell_{12}=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{\ell_{123}=|\ell_{12}-\ell_3|}^{\ell_{12}+\ell_3} G_{12}\,G_{123} \sqrt{\frac{(\ell_{123}-m_{123})!}{(\ell_{123}+m_{123})!}} \cdot \left[(-1)^{\ell_{123}} + (-1)^{m_{123}}\right] 2^{m_{123}-2}\, m_{123}\, \frac{\Gamma\!\left(\frac{\ell_{123}}{2}\right)\Gamma\!\left(\frac{\ell_{123}+m_{123}+1}{2}\right)}{\Gamma\!\left(\frac{\ell_{123}+3}{2}\right)\Gamma\!\left(\frac{\ell_{123}-m_{123}+2}{2}\right)}, \tag{16.52}$$

where

$$G_{123} = (-1)^{m_{123}}\,(2\ell_{123}+1) \begin{pmatrix} \ell_{12} & \ell_3 & \ell_{123} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell_{12} & \ell_3 & \ell_{123} \\ m_{12} & m_3 & -m_{123} \end{pmatrix} \tag{16.53}$$

and where ℓ_{123} is a summation index and $m_{123} \equiv m_1 + m_2 + m_3$.

17 Spherical harmonics

As mentioned in Chapter 15, the reader interested solely in the workings of the final code may skip this chapter and those on vector spherical harmonics and VSH series.

The statements in this chapter hold $\forall\,\theta \in [0,\pi]$, $\forall\,\phi \in [0,2\pi]$, $\forall\,\ell \in \mathbb{N}_0$ and $\forall\,m \in \mathbb{Z} : |m| \leq \ell$, unless otherwise stated.

17.1 Definition and examples

Definition 17.1. *Let $\ell \in \mathbb{N}_0$ and $m \in \mathbb{Z} : |m| \leq \ell$.* **Laplace's equation**, *when applied to functions independent of the radial coordinate r, states that*

$$
\begin{aligned}
-\ell(\ell+1)\,Y_\ell^m(\theta,\phi) &= r^2\,\nabla^2 Y_\ell^m && (17.1)\\
&= \frac{1}{\sin(\theta)}\,\frac{\partial}{\partial\theta}\left(\sin(\theta)\,\frac{\partial Y_\ell^m}{\partial\theta}\right) + \frac{1}{\sin^2(\theta)}\,\frac{\partial^2 Y_\ell^m}{\partial\phi^2} && (17.2)\\
&= \frac{\partial^2 Y_\ell^m}{\partial\theta^2} + \frac{\cos(\theta)}{\sin(\theta)}\,\frac{\partial Y_\ell^m}{\partial\theta} + \frac{1}{\sin^2(\theta)}\,\frac{\partial^2 Y_\ell^m}{\partial\phi^2}. && (17.3)
\end{aligned}
$$

The solutions to this equation are the **spherical harmonics** $Y_\ell^m : [0,\pi],[0,2\pi] \to \mathbb{C}$, *given by*

$$
Y_\ell^m(\theta,\phi) = \sqrt{\frac{2\ell+1}{4\pi}\,\frac{(\ell-m)!}{(\ell+m)!}}\,P_\ell^m\big[\cos(\theta)\big]\,e^{im\phi}, \tag{17.4}
$$

which are well known and hence given without proof.

The first few spherical harmonics are as follows. For $\ell = 0$,

$$
Y_0^0(\theta,\phi) = \frac{1}{2}\sqrt{\frac{1}{\pi}}. \tag{17.5}
$$

For $\ell = 1$,

$$
\begin{aligned}
Y_1^0(\theta,\phi) &= \frac{1}{2}\sqrt{\frac{3}{\pi}}\,\cos(\theta), && (17.6)\\
Y_1^{\pm1}(\theta,\phi) &= \mp\frac{1}{2}\sqrt{\frac{3}{2\pi}}\,\sin(\theta)\,e^{\pm i\phi}. && (17.7)
\end{aligned}
$$

For $\ell = 2$,

$$
\begin{aligned}
Y_2^0(\theta,\phi) &= \frac{1}{4}\sqrt{\frac{5}{\pi}}\,\big[3\cos^2(\theta)-1\big], && (17.8)\\
Y_2^{\pm1}(\theta,\phi) &= \mp\frac{1}{2}\sqrt{\frac{15}{2\pi}}\,\sin(\theta)\,\cos(\theta)\,e^{\pm i\phi}, && (17.9)\\
Y_2^{\pm2}(\theta,\phi) &= \frac{1}{4}\sqrt{\frac{15}{2\pi}}\,\sin^2(\theta)\,e^{\pm i2\phi}. && (17.10)
\end{aligned}
$$

For $\ell = 3$,

$$Y_3^0(\theta,\phi) = \frac{1}{4}\sqrt{\frac{7}{\pi}}\left[5\cos^3(\theta) - 3\cos(\theta)\right], \tag{17.11}$$

$$Y_3^{\pm1}(\theta,\phi) = \mp\frac{1}{8}\sqrt{\frac{21}{\pi}}\,\sin(\theta)\left[5\cos^2(\theta) - 1\right]e^{\pm i\phi}, \tag{17.12}$$

$$Y_3^{\pm2}(\theta\,\phi) = \frac{1}{4}\sqrt{\frac{105}{2\pi}}\,\sin^2(\theta)\,\cos(\theta)\,e^{\pm i2\phi}, \tag{17.13}$$

$$Y_3^{\pm3}(\theta,\phi) = \mp\frac{1}{8}\sqrt{\frac{35}{\pi}}\,\sin^3(\theta)\,e^{\pm i3\phi}. \tag{17.14}$$

17.2 Properties and relations

For the case $m = 0$, the spherical harmonics are real and simply become rescaled Legendre functions:

$$Y_\ell^0(\theta,\phi) = \sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell\big[\cos(\theta)\big] \in \mathbb{R}. \tag{17.15}$$

Corollary 17.2 (Spherical harmonics for negative m and complex conjugates)**.**

$$Y_\ell^{-m} = (-1)^m\,e^{-i2m\phi}\,Y_\ell^m = (-1)^m\left[Y_\ell^m\right]^*, \tag{17.16}$$

$$\left[Y_\ell^m\right]^* = e^{-i2m\phi}\,Y_\ell^m = (-1)^m\,Y_\ell^{-m}, \tag{17.17}$$

$$\left[Y_\ell^{-m}\right]^* = (-1)^m\,Y_\ell^m = (-1)^m\,e^{i2m\phi}\left[Y_\ell^m\right]^*. \tag{17.18}$$

Proof. Substitute $m \to -m$ and use Eq. (16.36). That $\left[Y_\ell^0\right]^* = Y_\ell^0$ follows either as a corollary of these arguments, or because $Y_\ell^0 \in \mathbb{R}$. □

There is a contraction rule which allows us to write the product of two spherical harmonics as an infinite sum of single spherical harmonics:

$$Y_{\ell_1}^{m_1}\,Y_{\ell_2}^{m_2} = \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)}{4\pi}}\sum_{\ell_3=0}^{\infty}\sum_{m_3=-\ell_3}^{\ell_3}(-1)^{m_3}\sqrt{2\ell_3+1}\begin{pmatrix}\ell_1 & \ell_2 & \ell_3\\ m_1 & m_2 & -m_3\end{pmatrix}\begin{pmatrix}\ell_1 & \ell_2 & \ell_3\\ 0 & 0 & 0\end{pmatrix}Y_{\ell_3}^{m_3}, \tag{17.19}$$

where the objects in parentheses are Wigner $3j$ symbols (Definition 16.5).

Theorem 17.3 (Differentiation and complex conjugation commute)**.** *Let f be a complex-valued function whose variables include $x \in \mathbb{R}$. Then, $\left(\frac{\partial f}{\partial x}\right)^* = \frac{\partial}{\partial x}(f^*)$.*

It follows from Theorem 17.3 that

$$\left(\frac{\partial Y_\ell^m}{\partial\theta}\right)^* = \frac{\partial}{\partial\theta}\left[Y_\ell^m\right]^*, \tag{17.20}$$

$$\left(\frac{\partial Y_\ell^m}{\partial\phi}\right)^* = \frac{\partial}{\partial\phi}\left[Y_\ell^m\right]^*, \tag{17.21}$$

so we do not need separate expressions for the derivatives of the complex conjugates of the spherical harmonics. We can simply differentiate the normal expression and take the conjugate of the result.

17.3 Derivatives with respect to θ

There are many ways to express the θ-derivatives of the spherical harmonics. Unfortunately, none yield a sum of only spherical harmonics. Instead, we will obtain an expression given by the sum of two spherical harmonics with the same ℓ as the original function, but with multiplying exponential factors. This is "as close as we can get" to maintaining a sum of spherical harmonics, but clearly it is not. However, our expression will indeed yield spherical harmonics for the case $m = 0$.

For $m = 0$, it follows immediately from Eq. (17.15) and Eq. (16.41) that

$$\frac{\partial Y_\ell^0}{\partial\theta} = \sqrt{\frac{2\ell+1}{4\pi}}\, P_\ell^1\big[\cos(\theta)\big]. \tag{17.22}$$

Proposition 17.4 (First θ-derivatives)**.**

$$\frac{\partial Y_\ell^m}{\partial\theta} = \frac{1}{2}\Big[\sqrt{(\ell-m)(\ell+m+1)}\, Y_\ell^{m+1}\, e^{-i\phi} - \sqrt{(\ell+m)(\ell-m+1)}\, Y_\ell^{m-1}\, e^{i\phi}\Big] \tag{17.23}$$

$$= \frac{1}{2}\sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}\, e^{im\phi}\Big[P_\ell^{m+1} - (\ell+m)(\ell-m+1)\, P_\ell^{m-1}\Big]. \tag{17.24}$$

This is valid for all ℓ, m as long as we define that $\forall\, m \in \mathbb{Z} : |m| > \ell,\ Y_\ell^m \equiv 0$. In particular, the "edge cases" are

$$\frac{\partial Y_\ell^\ell}{\partial\theta} = -\sqrt{\frac{\ell}{2}}\, Y_\ell^{\ell-1}\, e^{i\phi}, \tag{17.25}$$

$$\frac{\partial Y_\ell^{-\ell}}{\partial\theta} = \sqrt{\frac{\ell}{2}}\, Y_\ell^{-\ell+1}\, e^{-\phi}, \tag{17.26}$$

$$\frac{\partial Y_0^0}{\partial\theta} = 0. \tag{17.27}$$

Proof. Differentiate the definition of Y_ℓ^m in Eq. (17.4) and use the derivative of the ALFs in Eq. (16.44):

$$\frac{\partial Y_\ell^m}{\partial\theta} = \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-1)!}{(\ell+1)!}}\, e^{im\phi}\, \frac{dP_\ell^m}{d\theta} \tag{17.28}$$

$$= \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-1)!}{(\ell+1)!}}\, e^{im\phi}\left[\frac{1}{2}\, P_\ell^{m+1} - \frac{1}{2}\,(\ell+m)(\ell-m+1)\, P_\ell^{m+1}\right]. \tag{17.29}$$

We wish to express our result in terms of spherical harmonics. Note that

$$Y_\ell^{m\pm1} = \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m\mp1)!}{(\ell+m\pm1)!}}\, e^{i(m\pm1)\phi}\, P_\ell^{m\pm1}, \tag{17.30}$$

and that

$$\frac{(\ell-m)!}{(\ell+m)!} = (\ell-m)(\ell+m+1)\,\frac{(\ell-m-1)!}{(\ell+m+1)!} = \frac{1}{(\ell+m)(\ell-m+1)}\,\frac{(\ell-m+1)!}{(\ell+m-1)!}, \tag{17.31}$$

giving

$$\frac{\partial Y_\ell^m}{\partial\theta} = \sqrt{\frac{2\ell+1}{4\pi}(\ell-m)(\ell+m+1)\frac{(\ell-m-1)!}{(\ell+m+1)!}}\,\frac{1}{2}\, e^{-i\phi}\, e^{i(m+1)\phi}\, P_\ell^{m+1}$$

$$+\sqrt{\frac{2\ell+1}{4\pi}\frac{1}{(\ell+m)(\ell-m+1)!}\frac{(\ell-m+1)!}{(\ell+m-1)!}}\,(\ell+m)(\ell-m+1)\,\frac{1}{2}\, e^{i\phi}\, e^{i(m+1)\phi}\, P_\ell^{m-1}, \tag{17.32}$$

which simplifies to the given result. □

Corollary 17.5 (θ-derivatives for negative m and for complex conjugates)**.**

$$\frac{\partial}{\partial\theta}\,Y_\ell^{-m} \;=\; (-1)^m\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta}, \tag{17.33}$$

$$\frac{\partial}{\partial\theta}\left[Y_\ell^m\right]^* \;=\; e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta}, \tag{17.34}$$

$$\frac{\partial}{\partial\theta}\left[Y_\ell^{-m}\right]^* \;=\; (-1)^m\,\frac{\partial Y_\ell^m}{\partial\theta}. \tag{17.35}$$

Proof. Using Corollary 17.2, we have the following:

- $\frac{\partial Y_\ell^{-m}}{\partial\theta} = \frac{\partial}{\partial\theta}\left[(-1)^m\,e^{-i2m\phi}\,Y_\ell^m\right] = (-1)^m\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta}$.
- $\frac{\partial}{\partial\theta}\left[Y_\ell^m\right]^* = \frac{\partial}{\partial\theta}\left[e^{-i2m\phi}\,Y_\ell^m\right] = e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta}$.
- $\frac{\partial}{\partial\theta}\left[Y_\ell^{-m}\right]^* = \frac{\partial}{\partial\theta}(-1)^m\,Y_\ell^m\right] = (-1)^m\,\frac{\partial Y_\ell^m}{\partial\theta}$.

□

17.4 Derivatives with respect to ϕ

The ϕ-derivatives are less important since we will only be focusing on axisymmetric solutions. Nevertheless, we characterise them here to facilitate future generalisations of the code or applications of the method to non-axisymmetric systems. It is perhaps unfortunate that the ϕ-derivatives, not the θ-derivatives, are the ones we will not need, since they are far simpler to calculate. Since we are merely differentiating an exponential function, we see immediately that

$$\frac{\partial^n Y_\ell^m}{\partial\phi^n} \;=\; (im)^n\,Y_\ell^m, \qquad \forall\, n\in\mathbb{N}. \tag{17.36}$$

In particular,

$$\frac{\partial Y_\ell^m}{\partial\phi} \;=\; im\,Y_\ell^m, \tag{17.37}$$

$$\frac{\partial^2 Y_\ell^m}{\partial\phi} \;=\; -m^2\,Y_\ell^m, \tag{17.38}$$

$$\frac{\partial^n Y_\ell^0}{\partial\phi^n} \;=\; 0. \tag{17.39}$$

Corollary 17.6 (ϕ-derivatives for negative m and for complex conjugates)**.**

$$\frac{\partial}{\partial\phi}\,Y_\ell^{-m} \;=\; (-1)^{m+1}\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\phi} \;=\; (-1)^{m+1}\,e^{-i2m\phi}\,im\,Y_\ell^m, \tag{17.40}$$

$$\frac{\partial}{\partial\phi}\,\big[Y_\ell^m\big]^* \;=\; -e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\phi} \;=\; -e^{-i2m\phi}\,im\,Y_\ell^m, \tag{17.41}$$

$$\frac{\partial}{\partial\phi}\,\big[Y_\ell^{-m}\big]^* \;=\; (-1)^m\,\frac{\partial Y_\ell^m}{\partial\phi} \;=\; (-1)^m\,im\,Y_\ell^m. \tag{17.42}$$

Proof. Using Corollary 17.2, we have the following:

- $$\frac{\partial Y_\ell^{-m}}{\partial\phi} \;=\; \frac{\partial}{\partial\phi}\Big[(-1)^m\,e^{-i2m\phi}\,Y_\ell^m\Big] \tag{17.43}$$
$$\;=\; (-1)^m\,\frac{\partial}{\partial\phi}\Big[e^{-i2m\phi}\Big]\,Y_\ell^m + (-1)^m\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\phi} \tag{17.44}$$
$$\;=\; (-1)^m\Big[-i2m\,e^{-i2m\phi}\Big]\,Y_\ell^m + (-1)^m\,e^{-i2m\phi}\,im\,Y_\ell^m \tag{17.45}$$
$$\;=\; (-1)^m\,e^{-i2m\phi}\,im\,Y_\ell^m\big[-2+1\big] \tag{17.46}$$
$$\;=\; (-1)^{m+1}\,e^{-i2m\phi}\,im\,Y_\ell^m \;=\; (-1)^{m+1}\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\phi}. \tag{17.47}$$
- $\frac{\partial}{\partial\phi}\big[Y_\ell^m\big]^* = \frac{\partial}{\partial\phi}\big[e^{-i2m\phi}\,Y_\ell^m\big] = (-1)^m\,\frac{\partial}{\partial\phi}\,Y_\ell^{-m} = (-1)^{2m+1}\,e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta} = -e^{-i2m\phi}\,\frac{\partial Y_\ell^m}{\partial\theta}$.
- $\frac{\partial}{\partial\phi}\big[Y_\ell^{-m}\big]^* = \frac{\partial}{\partial\phi}\big[(-1)^m\,(Y_\ell^m)^*\big]^* = (-1)^m\,\frac{\partial Y_\ell^m}{\partial\phi}$.

The alternative expressions follow from $\frac{\partial Y_\ell^m}{\partial\phi} = im\,Y_\ell^m$. □

We see from Eq. (9.4) that, for some vector function $\mathbf{A}$, the expression for $\boldsymbol{\nabla}\times\mathbf{A}$ contains terms like $\frac{1}{\sin(\theta)}\,A_\phi$ and $\frac{1}{\sin(\theta)}\,\frac{\partial A_\phi}{\partial\theta}$, so our approximations of $\mathbf{A}$ will contain terms like $\frac{1}{\sin(\theta)}\,Y_\ell^m$ and $\frac{1}{\sin(\theta)}\,\frac{\partial Y_\ell^m}{\partial\theta}$. The division by $\sin(\theta)$ will cause issues at the poles, so let us find an alternative expression for these terms that avoid the division.

Proposition 17.7. *$\forall\,\ell\in\mathbb{N}_0$ and $\forall\,m\in\mathbb{Z}:(|m|\le\ell)\wedge(m\neq 0)$,*

$$\frac{1}{\sin(\theta)}\,Y_\ell^m = -\frac{1}{2m}\sqrt{\frac{2\ell+1}{2\ell+3}}\Big(\sqrt{(\ell+m+2)(\ell+m+1)}\,Y_{\ell+1}^{m+1}\,e^{-i\phi} + \sqrt{(\ell-m+2)(\ell-m+1)}\,Y_{\ell+1}^{m-1}\,e^{i\phi}\Big) \tag{17.48}$$

We do not consider the case $m=0$.

Proof. The ALFs obey the recursion relation

$$\frac{1}{\sin(\theta)}\,P_\ell^m\big[\cos(\theta)\big] = -\frac{1}{2m}\,P_{\ell+1}^{m+1}\big[\cos(\theta)\big] - \frac{1}{2m}\,(\ell-m+1)(\ell-m+2)\,P_{\ell+1}^{m-1}\big[\cos(\theta)\big], \tag{17.49}$$

so it follows that

$$\frac{1}{\sin(\theta)}\,Y_\ell^m = \sqrt{\frac{2\ell+1}{4\pi}\,\frac{(\ell-1)!}{(\ell+1)!}}\,e^{im\phi}\left(-\frac{1}{2m}\right)P_{\ell+1}^{m+1} + \sqrt{\frac{2\ell+1}{4\pi}\,\frac{(\ell-1)!}{(\ell+1)!}}\,e^{im\phi}\left(-\frac{1}{2m}\right)(\ell-m+1)(\ell-m+2)\,P_{\ell+1}^{m-1}. \tag{17.50}$$

Now,

$$Y_{\ell+1}^{m+1} = \sqrt{\frac{2(\ell+1)+1}{4\pi}\frac{\big[(\ell+1)-(m+1)\big]!}{\big[(\ell+1)+(m+1)\big]!}}\,P_{\ell+1}^{m+1}\,e^{i(m+1)\phi} \tag{17.51}$$

$$= \sqrt{\frac{2\ell+3}{4\pi}\frac{(\ell-m)!}{(\ell+m+2)!}}\,P_{\ell+1}^{m+1}\,e^{i\phi}\,e^{im\phi} \tag{17.52}$$

$$= \sqrt{\frac{2\ell+3}{(\ell+m+2)(\ell+m+1)}}\,e^{i\phi}\sqrt{\frac{1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}\,P_{\ell+1}^{m+1}\,e^{im\phi}, \tag{17.53}$$

$$\Rightarrow \sqrt{\frac{1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}\,P_{\ell+1}^{m+1}\,e^{im\phi} = \sqrt{\frac{(\ell+m+2)(\ell+m+1)}{2\ell+3}}\,e^{-i\phi}\,Y_{\ell+1}^{m+1}, \tag{17.54}$$

and

$$Y_{\ell+1}^{m-1} = \sqrt{\frac{2(\ell+1)+1}{4\pi}\frac{\big[(\ell+1)-(m-1)\big]!}{\big[(\ell+1)+(m-1)\big]!}}\,P_{\ell+1}^{m-1}\,e^{i(m-1)\phi} \tag{17.55}$$

$$= \sqrt{\frac{2\ell+3}{4\pi}\frac{(\ell-m+2)!}{(\ell+m)!}}\,P_{\ell+1}^{m+1}\,e^{i\phi}\,e^{im\phi} \tag{17.56}$$

$$= \sqrt{(2\ell+3)(\ell-m+2)(\ell-m+1)}\,e^{-i\phi}\sqrt{\frac{1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}\,P_{\ell+1}^{m-1}\,e^{im\phi}, \tag{17.57}$$

$$\Rightarrow \sqrt{\frac{1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}}\,P_{\ell+1}^{m-1}\,e^{im\phi} = \frac{1}{\sqrt{(2\ell+3)(\ell-m+2)(\ell-m+1)}}\,e^{i\phi}\,Y_{\ell+1}^{m+1}. \tag{17.58}$$

Substituting these, we obtain the given result. □

Verification. `Check mathematical expressions/20230726_Ylm_over_sintheta.cpp` □

It follows from Proposition 17.7 that It follows that $\forall\,\ell\in\mathbb{N}_0$ and $\forall\,m\in\mathbb{Z}:(|m|\leq\ell)\wedge(m\neq 0)$,

$$\frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi} = -\frac{i}{2}\sqrt{\frac{2\ell+1}{2\ell+3}}\Big(\sqrt{(\ell+m+2)(\ell+m+1)}\,Y_{\ell+1}^{m+1}\,e^{-i\phi} + \sqrt{(\ell-m+2)(\ell-m+1)}\,Y_{\ell+1}^{m-1}\,e^{i\phi}\Big). \tag{17.59}$$

For $m=0$, the result is zero due to Eq. (17.39).

17.5 Integrals

Lemma 17.8. $\forall\, a \in \mathbb{R},\ \int_0^{2\pi} e^{iax}\, dx = 2\pi\, \delta_{a,0}$.

Proof. If $a \neq 0$, the power rule gives $\frac{1}{ia}\big[e^{iax}\big]_0^{2\pi} = -\frac{i}{a}\left[e^{i2\pi} - e^0\right] = -\frac{i}{a}\left[1-1\right] = 0$. If $a = 0$, we simply have $\int_0^{2\pi} dx = 2\pi$. □

Proposition 17.9 (Orthonormality)**.** $\iint Y_\ell^m \left[Y_{\ell'}^{m'}\right]^* d\Omega \ = \ \delta_{\ell,\ell'}\, \delta^{m,m'}$.

Proof. We have

$$\iint Y_\ell^m \left[Y_{\ell'}^{m'}\right]^* d\Omega \ = \ \int_0^{2\pi}\int_0^{\pi} Y_\ell^m \left[Y_{\ell'}^{m'}\right]^* \sin(\theta)\, d\theta\, d\phi \tag{17.60}$$

$$= \int_0^{2\pi}\int_0^{\pi} \sqrt{\frac{2\ell+1}{4\pi}\frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m\big[\cos(\theta)\big]\, e^{im\phi} \cdot \sqrt{\frac{2\ell'+1}{4\pi}\frac{(\ell'-m')!}{(\ell'+m')!}} P_{\ell'}^{m'}\big[\cos(\theta)\big]\, e^{-im'\phi}\, \sin(\theta)\, d\theta\, d\phi \tag{17.61}$$

$$= \frac{1}{4\pi}\sqrt{(2\ell+1)\,(2\ell'+1)\,\frac{(\ell-m)!}{(\ell+m)!}\,\frac{(\ell'-m')!}{(\ell'+m')!}} \cdot \int_0^{2\pi} e^{i(m-m')\phi}\, d\phi \int_0^{\pi} P_\ell^m\big[\cos(\theta)\big]\, P_{\ell'}^{m'}\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \tag{17.62}$$

$$= \frac{1}{4\pi}\sqrt{(2\ell+1)\,(2\ell'+1)\,\frac{(\ell-m)!}{(\ell+m)!}\,\frac{(\ell'-m')!}{(\ell'+m')!}} \cdot 2\pi\, \delta^{m-m',0} \int_0^{\pi} P_\ell^m\big[\cos(\theta)\big]\, P_{\ell'}^{m'}\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \tag{17.63}$$

$$= \frac{1}{2}\sqrt{(2\ell+1)\,(2\ell'+1)\,\frac{(\ell-m)!}{(\ell+m)!}\,\frac{(\ell'-m)!}{(\ell'+m)!}}\, \delta^{m,m'} \int_0^{\pi} P_\ell^m\big[\cos(\theta)\big]\, P_{\ell'}^{m}\big[\cos(\theta)\big]\, \sin(\theta)\, d\theta \tag{17.64}$$

$$= \delta_{\ell,\ell'}\, \delta^{m,m'}, \tag{17.65}$$

where we used Lemma 17.8, we used that $\delta_{a,b} = \delta^{a,b}$ and $\delta_{a-b,0} = \delta_{a,b}$, we were able to evaluate the remaining terms at $m = m'$, even though there was no summation over m, m', because the expression is zero otherwise, and we used the orthogonality condition (16.45). □

18 Vector spherical harmonics

In addition to the relations proven in these chapters, detailed discussions on the vector spherical harmonics and their application for modelling vector functions can be found in Barrera et al. (1985) and Appendix A of Pétri (2012).

18.1 Definition and examples

Definition 18.1. *Let* $\theta \in [0,\pi]$, $\phi \in [0,2\pi]$, $\ell \in \mathbb{N}_0$, $m \in \mathbb{Z} : |m| \le \ell$. *The* **vector spherical harmonics** *(abbreviation* **VSHs***; singular* **VSH***) are the vector functions* $\mathbf{Y}_\ell^m \in \mathbb{C}^3$, $\mathbf{\Psi}_\ell^m \in \mathbb{C}^3$, $\mathbf{\Phi}_\ell^m \in \mathbb{C}^3$ *given by*

$$\mathbf{Y}_\ell^m(\theta,\phi) = Y_\ell^m(\theta,\phi)\,\mathbf{e}_r, \tag{18.1}$$

$$\mathbf{\Psi}_\ell^m(\theta,\phi) = r\,\boldsymbol{\nabla} Y_\ell^m(\theta,\phi), \tag{18.2}$$

$$\mathbf{\Phi}_\ell^m(\theta,\phi) = \mathbf{e}_r \times r\,\boldsymbol{\nabla} Y_\ell^m(\theta,\phi) = \mathbf{e}_r \times \mathbf{\Psi}_\ell^m(\theta,\phi). \tag{18.3}$$

Note that some authors define

$$\mathbf{\Psi}_\ell^m = \frac{1}{\sqrt{\ell(\ell+1)}}\, r\,\boldsymbol{\nabla} Y_\ell^m, \tag{18.4}$$

so the numerical factors multiplying $\mathbf{\Psi}_\ell^m$ and hence $\mathbf{\Phi}_\ell^m$ may vary compared to other sources. The first few VSHs are as follows. For $\ell = 0$,

$$\mathbf{\Psi}_0^0(\theta,\phi) = \mathbf{\Phi}(\theta,\phi) = \mathbf{0}. \tag{18.5}$$

For $\ell = 1$,

$$\mathbf{\Psi}_1^0(\theta,\phi) = -\frac{1}{2}\sqrt{\frac{3}{\pi}}\,\sin(\theta)\,\mathbf{e}_\theta, \tag{18.6}$$

$$\mathbf{\Psi}_1^{\pm1}(\theta,\phi) = \mp\frac{1}{2}\sqrt{\frac{3}{2\pi}}\,e^{\pm i\phi}\Big[\cos(\theta)\,\mathbf{e}_\theta + i\,\mathbf{e}_\phi\Big]. \tag{18.7}$$

For $\ell = 2$,

$$\mathbf{\Psi}_2^0(\theta,\phi) = -\frac{3}{2}\sqrt{\frac{5}{\pi}}\,\sin(\theta)\,\cos(\theta)\,\mathbf{e}_\theta, \tag{18.8}$$

$$\mathbf{\Psi}_2^{\pm1}(\theta,\phi) = \mp\frac{1}{2}\sqrt{\frac{15}{2\pi}}\,e^{\pm i\phi}\big[\cos(2\theta)\,\mathbf{e}_\theta + i\,\cos(\theta)\,\mathbf{e}_\phi\big], \tag{18.9}$$

$$\mathbf{\Psi}_2^{\pm2}(\theta,\phi) = \frac{1}{2}\sqrt{\frac{15}{2\pi}}\,e^{\pm i2\phi}\,\sin(\theta)\,\big[\cos(\theta)\,\mathbf{e}_\theta + i\,\mathbf{e}_\phi\big]. \tag{18.10}$$

For $\ell = 3$,

$$\mathbf{\Psi}_3^0(\theta,\phi) = -\frac{3}{4}\sqrt{\frac{7}{\pi}}\,\sin(\theta)\,\big[5\cos^2(\theta)-1\big]\,\mathbf{e}_\theta, \tag{18.11}$$

$$\mathbf{\Psi}_3^{\pm1}(\theta,\phi) = \pm\frac{1}{8}\sqrt{\frac{21}{\pi}}\,e^{\pm i\phi}\Big[\big[5\cos^3(\theta)-9\cos(\theta)\big]\,\mathbf{e}_\theta - i\big[5\cos^2(\theta)-1\big]\mathbf{e}_\phi\Big], \tag{18.12}$$

$$\mathbf{\Psi}_3^{\pm2}(\theta,\phi) = \frac{1}{4}\sqrt{\frac{105}{2\pi}}\,e^{\pm i2\phi}\,\sin(\theta)\Big[\big[3\cos^2(\theta)-1\big]\,\mathbf{e}_\theta - i\,2\cos(\theta)\,\mathbf{e}_\phi\Big], \tag{18.13}$$

$$\mathbf{\Psi}_3^{\pm3}(\theta,\phi) = \mp\frac{3}{8}\sqrt{\frac{35}{\pi}}\,e^{\pm i3\phi}\,\sin^2(\theta)\big[\cos(\theta)\,\mathbf{e}_\theta + i\,\mathbf{e}_\phi\big]. \tag{18.14}$$

18.2 Properties and relations

Proposition 18.2 (Obtaining $\mathbf{\Phi}_\ell^m$ from $\mathbf{\Psi}_\ell^m$)**.** *Given $\mathbf{\Psi}_\ell^m$, we immediately obtain $\mathbf{\Phi}_\ell^m$ by swapping the order of the vector components and adding a minus sign to the new θ-component.*

Proof. Let us write that $\mathbf{\Psi}_\ell^m = \Psi_\theta\,\mathbf{e}_\theta + \Psi_\phi\,\mathbf{e}_\phi$. Then,

$$\begin{aligned}\mathbf{\Phi}_\ell^m &= \mathbf{e}_r \times \big(\Psi_\theta\,\mathbf{e}_\theta + \Psi_\phi\,\mathbf{e}_\phi\big) && (18.15)\\ &= \Psi_\theta\,\mathbf{e}_r \times \mathbf{e}_\theta + \Psi_\phi\,\mathbf{e}_r \times \mathbf{e}_\phi && (18.16)\\ &= -\Psi_\phi\,\mathbf{e}_\theta + \Psi_\theta\,\mathbf{e}_\phi. && (18.17)\end{aligned}$$

□

For example, we can read off from Eq. (18.13) that

$$\mathbf{\Phi}_3^{\pm 2}(\theta,\phi) = \frac{1}{4}\sqrt{\frac{105}{2\pi}}\,e^{\pm i2\phi}\,\sin(\theta)\Big[i\,2\cos(\theta)\,\mathbf{e}_\theta + \big[3\cos^2(\theta)-1\big]\,\mathbf{e}_\phi\Big]. \tag{18.18}$$

Proposition 18.3 (Expressions in spherical polar coordinates)**.** *We have*

$$\mathbf{\Psi}_\ell^m(\theta,\phi) = \frac{\partial Y_\ell^m}{\partial\theta}\,\mathbf{e}_\theta + \frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\,\mathbf{e}_\phi, \tag{18.19}$$

$$\mathbf{\Phi}_\ell^m(\theta,\phi) = -\frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\,\mathbf{e}_\theta + \frac{\partial Y_\ell^m}{\partial\theta}\,\mathbf{e}_\phi. \tag{18.20}$$

When evaluating the VSHs numerically, we may use Eq. (17.59) to avoid numerical issues at the poles $\theta \in \{0,\pi\}$. In particular, for $m = 0$,

$$\mathbf{Y}_\ell^0 = \sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell\big[\cos(\theta)\big]\,\mathbf{e}_r, \tag{18.21}$$

$$\mathbf{\Psi}_\ell^0 = \sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^1\big[\cos(\theta)\big]\,\mathbf{e}_\theta, \tag{18.22}$$

$$\mathbf{\Phi}_\ell^0 = \sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^1\big[\cos(\theta)\big]\,\mathbf{e}_\phi, \tag{18.23}$$

and so $\mathbf{Y}_\ell^0, \mathbf{\Psi}_\ell^0, \mathbf{\Phi}_\ell^0 \in \mathbb{R}^3$.

Proof. Evaluate the expression for $\mathbf{\Psi}_\ell^m$ in Definition 18.1 using the expression for the gradient of a function in Eq. (9.2). We then obtain $\mathbf{\Psi}_\ell^m$ immediately by Proposition 18.2. For the $m = 0$ expressions, use Eqs. (17.15) and (17.22). □

Proposition 18.4 (Dot products)**.** *We have*

$$\mathbf{Y}_\ell^m \cdot \mathbf{Y}_{\ell'}^{m'} = Y_\ell^m \, Y_{\ell'}^{m'}, \tag{18.24}$$

$$\boldsymbol{\Psi}_\ell^m \cdot \boldsymbol{\Psi}_{\ell'}^{m'} = \boldsymbol{\Phi}_\ell^m \cdot \boldsymbol{\Phi}_{\ell'}^{m'} = \frac{\partial Y_\ell^m}{\partial \theta} \frac{\partial Y_{\ell'}^{m'}}{\partial \theta} + \frac{1}{\sin^2(\theta)} \frac{\partial Y_\ell^m}{\partial \phi} \frac{\partial Y_{\ell'}^{m'}}{\partial \phi} = \frac{\partial Y_\ell^m}{\partial \theta} \frac{\partial Y_{\ell'}^{m'}}{\partial \theta} - \frac{m m'}{\sin^2(\theta)} Y_\ell^m \, Y_{\ell'}^{m'}, \tag{18.25}$$

$$\mathbf{Y}_\ell^m \cdot \boldsymbol{\Psi}_{\ell'}^{m'} = \mathbf{Y}_\ell^m \cdot \boldsymbol{\Phi}_{\ell'}^{m'} = 0, \tag{18.26}$$

$$\boldsymbol{\Psi}_\ell^m \cdot \boldsymbol{\Phi}_{\ell'}^{m'} = \frac{1}{\sin(\theta)} \left[\frac{\partial Y_\ell^m}{\partial \phi} \frac{\partial Y_{\ell'}^{m'}}{\partial \theta} - \frac{\partial Y_\ell^m}{\partial \theta} \frac{\partial Y_{\ell'}^{m'}}{\partial \phi} \right] = \frac{i}{\sin(\theta)} \left[m \, Y_\ell^m \frac{\partial Y_{\ell'}^{m'}}{\partial \theta} - m' \, Y_{\ell'}^{m'} \frac{\partial Y_\ell^m}{\partial \theta} \right]. \tag{18.27}$$

Proof. Substitute the expressions in Proposition 18.3 and take the dot products. □

In particular, note that

$$\boldsymbol{\Psi}_{\ell'}^{m'} \cdot \boldsymbol{\Phi}_\ell^m = -\boldsymbol{\Psi}_\ell^m \cdot \boldsymbol{\Phi}_{\ell'}^{m'}, \tag{18.28}$$

i.e. we can swap the indices on $\boldsymbol{\Psi}_\ell^m \cdot \boldsymbol{\Phi}_{\ell'}^{m'}$ at the cost of a minus sign, and that

$$\mathbf{Y}_\ell^m \cdot \boldsymbol{\Psi}_\ell^m = \mathbf{Y}_\ell^m \cdot \boldsymbol{\Phi}_\ell^m = \boldsymbol{\Psi}_\ell^m \cdot \boldsymbol{\Phi}_\ell^m = 0. \tag{18.29}$$

Proposition 18.5 (Orthogonality over a surface integral)**.** *We have the following relations:*

$$\iint \mathbf{Y}_\ell^m \cdot \left[\mathbf{Y}_{\ell'}^{m'}\right]^* d\Omega = \delta_{\ell,\ell'} \, \delta^{m,m'}, \tag{18.30}$$

$$\iint \boldsymbol{\Psi}_\ell^m \cdot \left[\boldsymbol{\Psi}_{\ell'}^{m'}\right]^* d\Omega = \iint \boldsymbol{\Phi}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right]^* d\Omega = \ell(\ell+1) \, \delta_{\ell,\ell'} \, \delta^{m,m'}, \tag{18.31}$$

$$\iint \mathbf{Y}_\ell^m \cdot \left[\boldsymbol{\Psi}_{\ell'}^{m'}\right]^* d\Omega = \iint \mathbf{Y}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right]^* d\Omega = \iint \boldsymbol{\Psi}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right] d\Omega = 0. \tag{18.32}$$

These results are given without proof in Eq. (3.21) of Barrera et al. (1985).

Partial proof.

- $\iint \mathbf{Y}_\ell^m \cdot \left[\mathbf{Y}_{\ell'}^{m'}\right]^* d\Omega = \iint Y_\ell^m \left[Y_\ell^m\right]^* d\Omega = \delta_{\ell,\ell'} \, \delta^{m,m'}$, where we used Proposition 17.9.
- That $\iint \mathbf{Y}_\ell^m \cdot \left[\boldsymbol{\Psi}_{\ell'}^{m'}\right]^* d\Omega = \iint \mathbf{Y}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right]^* d\Omega = 0$ follows immediately from Eq. (18.26).
- We are unable to prove $\iint \boldsymbol{\Psi}_\ell^m \cdot \left[\boldsymbol{\Psi}_{\ell'}^{m'}\right]^* d\Omega$, $\iint \boldsymbol{\Phi}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right]^* d\Omega$ or $\iint \boldsymbol{\Psi}_\ell^m \cdot \left[\boldsymbol{\Phi}_{\ell'}^{m'}\right] d\Omega$.
- The remaining expressions, e.g. $\iint \boldsymbol{\Psi}_\ell^m \cdot \left[\mathbf{Y}_\ell^m\right]^* d\Omega$, follow by taking the complex conjugate of both sides of the expression.

□

Verification.

`20221108 Vector spherical harmonics/Dot products/Check_integrated_dot_product_Psi_Psi.cpp`
`20221108 Vector spherical harmonics/Dot products/Check_integrated_dot_product_Phi_Phi.cpp` □

Proposition 18.6 (Cross products)**.**

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{Y}_{\ell_2}^{m_2} = \mathbf{0}, \tag{18.33}$$

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{\Psi}_{\ell_2}^{m_2} = Y_{\ell_1}^{m_1}\, \mathbf{\Phi}_{\ell_2}^{m_2}, \tag{18.34}$$

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{\Phi}_{\ell_2}^{m_2} = -Y_{\ell_1}^{m_1}\, \mathbf{\Psi}_{\ell_2}^{m_2}, \tag{18.35}$$

$$\mathbf{\Psi}_{\ell_1}^{m_1} \times \mathbf{\Psi}_{\ell_2}^{m_2} = \mathbf{\Phi}_{\ell_1}^{m_1} \times \mathbf{\Phi}_{\ell_2}^{m_2} = \left[\frac{1}{\sin(\theta)}\, \frac{\partial Y_{\ell_1}^{m_1}}{\partial \theta}\, \frac{\partial Y_{\ell_2}^{m_2}}{\partial \phi} - \frac{1}{\sin(\theta)}\, \frac{\partial Y_{\ell_1}^{m_1}}{\partial \phi}\, \frac{\partial Y_{\ell_2}^{m_2}}{\partial \theta}\right] \mathbf{e}_r, \tag{18.36}$$

$$\mathbf{\Psi}_{\ell_1}^{m_1} \times \mathbf{\Phi}_{\ell_2}^{m_2} = \left[\frac{\partial Y_{\ell_1}^{m_1}}{\partial \theta}\, \frac{\partial Y_{\ell_2}^{m_2}}{\partial \theta} - \frac{1}{\sin^2(\theta)}\, \frac{\partial Y_{\ell_1}^{m_1}}{\partial \phi}\, \frac{\partial Y_{\ell_2}^{m_2}}{\partial \phi}\right] \mathbf{e}_r. \tag{18.37}$$

Proof. Firstly,

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{Y}_{\ell_2}^{m_2} = \left(Y_{\ell_1}^{m_1}\, \mathbf{e}_r\right) \times \left(Y_{\ell_2}^{m_2}\, \mathbf{e}_r\right) = Y_{\ell_1}^{m_1} Y_{\ell_2}^{m_2}\, \mathbf{e}_r \times \mathbf{e}_r = \mathbf{0}. \tag{18.38}$$

Secondly,

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{\Psi}_{\ell_2}^{m_2} = \left(Y_{\ell_1}^{m_1}\, \mathbf{e}_r\right) \times \mathbf{\Psi}_{\ell_2}^{m_2} = Y_{\ell_1}^{m_1}\, \mathbf{e}_r \times \mathbf{\Psi}_{\ell_2}^{m_2} = Y_{\ell_1}^{m_1}\, \mathbf{\Phi}_{\ell_2}^{m_2}. \tag{18.39}$$

Lagrange's formula states that, for three general vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, we have

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\, \mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\, \mathbf{c}. \tag{18.40}$$

Then,

$$\mathbf{Y}_{\ell_1}^{m_1} \times \mathbf{\Phi}_{\ell_2}^{m_2} = \mathbf{Y}_{\ell_1}^{m_1} \times (\mathbf{e}_r \times \mathbf{\Psi}_{\ell_2}^{m_2}) = (\mathbf{Y}_{\ell_1}^{m_1} \cdot \mathbf{\Psi}_{\ell_2}^{m_2})\, \mathbf{e}_r - (\mathbf{Y}_{\ell_1}^{m_1} \cdot \mathbf{e}_r)\, \mathbf{\Psi}_{\ell_2}^{m_2} = 0 - Y_{\ell_1}^{m_1}\, \mathbf{\Psi}_{\ell_2}^{m_2}. \tag{18.41}$$

Lagrange's formula also gives the remaining expressions. □

In particular, note that $\mathbf{\Psi}_{\ell_2}^{m_2} \times \mathbf{\Phi}_{\ell_1}^{m_1} = \mathbf{\Psi}_{\ell_1}^{m_1} \times \mathbf{\Phi}_{\ell_2}^{m_2}$. When encoding these on a computer, it is important to consider the $m_1, m_2 = 0$ cases separately in order to avoid $0/0$ errors at $\sin(\theta) = 0$. In axisymmetry, we have

$$\mathbf{Y}_{\ell_1} \times \mathbf{Y}_{\ell_2} = \mathbf{\Psi}_{\ell_1} \times \mathbf{\Psi}_{\ell_2} = \mathbf{\Phi}_{\ell_1} \times \mathbf{\Phi}_{\ell_2} = \mathbf{0}, \tag{18.42}$$

$$\mathbf{Y}_{\ell_1} \times \mathbf{\Psi}_{\ell_2} = Y_{\ell_1}\, \mathbf{\Phi}_{\ell_2} = \frac{1}{4\pi} \sqrt{(2\ell_1+1)(2\ell_2+1)}\, P_{\ell_1}^0\, P_{\ell_2}^1\, \mathbf{e}_\phi, \tag{18.43}$$

$$\mathbf{Y}_{\ell_1} \times \mathbf{\Phi}_{\ell_2} = -Y_{\ell_1}\, \mathbf{\Psi}_{\ell_2} = -\frac{1}{4\pi} \sqrt{(2\ell_1+1)(2\ell_2+1)}\, P_{\ell_1}^0\, P_{\ell_2}^1\, \mathbf{e}_\theta, \tag{18.44}$$

$$\mathbf{\Psi}_{\ell_1} \times \mathbf{\Phi}_{\ell_2} = \frac{1}{4\pi} \sqrt{(2\ell_1+1)(2\ell_2+1)}\, P_{\ell_1}^1\, P_{\ell_2}^1\, \mathbf{e}_r. \tag{18.45}$$

Proposition 18.7 (Divergence)**.**

$$\boldsymbol{\nabla}\cdot\mathbf{Y}_\ell^m = \frac{2}{r}Y_\ell^m, \tag{18.46}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\Psi}_\ell^m = -\frac{\ell(\ell+1)}{r}Y_\ell^m, \tag{18.47}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_\ell^m = 0, \tag{18.48}$$

and so $\boldsymbol{\Phi}_\ell^m$ is solenoidal.

Proof. Using the expression for the divergence of a vector in spherical polar coordinates, Eq. (9.5), we obtain the following:

- For $\mathbf{Y}_\ell^m$,

$$\boldsymbol{\nabla}\cdot\mathbf{Y}_\ell^m = \frac{\partial Y_\ell^m}{\partial r}+\frac{2}{r}Y_\ell^m+0 = \frac{2}{r}Y_\ell^m. \tag{18.49}$$

- For $\boldsymbol{\Phi}_\ell^m$,

$$\boldsymbol{\nabla}\cdot\boldsymbol{\Phi}_\ell^m = 0+\frac{1}{r}\frac{\partial^2 Y_\ell^m}{\partial\theta^2}+\frac{\cos(\theta)}{r\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\theta}+\frac{1}{r\sin^2(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi^2} \tag{18.50}$$

$$= \frac{1}{r}\left[\frac{\partial^2 Y_\ell^m}{\partial\theta^2}+\frac{\cos(\theta)}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\theta}+\frac{1}{\sin^2(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi^2}\right] = -\frac{\ell(\ell+1)}{r}Y_\ell^m, \tag{18.51}$$

  where we recognised the defining differential equation for Y_ℓ^m (Definition 17.1).

- For $\boldsymbol{\Psi}_\ell^m$,

$$\boldsymbol{\nabla}\cdot\boldsymbol{\Psi}_\ell^m = 0+\frac{1}{r}\frac{\partial}{\partial\theta}\left(-\frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\right)-\frac{\cos(\theta)}{r\sin^2(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}+\frac{1}{r\sin(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi\,\partial\theta} \tag{18.52}$$

$$= \frac{1}{r}\left(\frac{\cos(\theta)}{\sin^2(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}-\frac{1}{\sin(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi^2}-\frac{\cos(\theta)}{\sin^2(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}+\frac{1}{\sin(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi\,\partial\theta}\right) = 0. \tag{18.53}$$

□

Proposition 18.8 (Curl)**.**

$$\boldsymbol{\nabla}\times\mathbf{Y}_\ell^m = -\boldsymbol{\nabla}\times\boldsymbol{\Psi}_\ell^m = -\frac{1}{r}\boldsymbol{\Phi}_\ell^m, \tag{18.54}$$

$$\boldsymbol{\nabla}\times\boldsymbol{\Phi}_\ell^m = -\frac{1}{r}\Big[\ell(\ell+1)\,\mathbf{Y}_\ell^m+\boldsymbol{\Psi}_\ell^m\Big]. \tag{18.55}$$

Proof. Use the expression for the curl of a vector in spherical polar coordinates, Eq. (9.6). The expressions for $\mathbf{Y}_\ell^m$ and $\boldsymbol{\Psi}_\ell^m$ follow immediately. For $\boldsymbol{\Phi}_\ell^m$, we have

$$\boldsymbol{\nabla}\times\boldsymbol{\Phi}_\ell^m = \left[\frac{1}{r}\frac{\partial Y_\ell^m}{\partial\theta^2}+\frac{\cos(\theta)}{r\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\theta}+\frac{1}{r\sin(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi^2}\right]\mathbf{e}_r+\left[0-\frac{1}{r}\frac{\partial Y_\ell^m}{\partial\theta}\right]\mathbf{e}_\theta+\left[0-\frac{1}{r\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\right]\mathbf{e}_\phi \tag{18.56}$$

$$= \frac{1}{r}\left[\left(\frac{\partial Y_\ell^m}{\partial\theta^2}+\frac{\cos(\theta)}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\theta}+\frac{1}{\sin(\theta)}\frac{\partial^2 Y_\ell^m}{\partial\phi^2}\right)\mathbf{e}_r-\frac{\partial Y_\ell^m}{\partial\theta}\mathbf{e}_\theta-\frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\mathbf{e}_\phi\right] \tag{18.57}$$

$$= \frac{1}{r}\left[-\ell(\ell+1)\,Y_\ell^m\,\mathbf{e}_r-\frac{\partial Y_\ell^m}{\partial\theta}\mathbf{e}_\theta-\frac{1}{\sin(\theta)}\frac{\partial Y_\ell^m}{\partial\phi}\mathbf{e}_\phi\right] \tag{18.58}$$

$$= -\frac{1}{r}\Big[\ell(\ell+1)\,\mathbf{Y}_\ell^m+\boldsymbol{\Psi}_\ell^m\Big]. \tag{18.59}$$

□

19 VSH series

19.1 VSH series of arbitrary vector functions

Definition 19.1. *A general vector field* $\mathbf{A}(r,\theta,\phi)$ *can be expanded as an infinite sum over vector spherical harmonics as*

$$\mathbf{A}(r,\theta,\phi) \;=\; \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell}\Big[A_m^{r,\ell}(r)\,\mathbf{Y}_\ell^m(\theta,\phi) + A_m^{(1),\ell}(r)\,\mathbf{\Psi}_\ell^m(\theta,\phi) + A_m^{(2),\ell}(r)\,\mathbf{\Phi}_\ell^m(\theta,\phi)\Big]. \tag{19.1}$$

In this text, we will refer to an expansion of this form as the **VSH series of** $\mathbf{A}(r,\theta,\phi)$ *and to* $A_m^{r,\ell}$, $A_m^{(1),\ell}$, $A_m^{(2),\ell}$ *as the* **VSH series coefficients of** $\mathbf{A}(r,\theta,\phi)$.

The coefficients carry all of the radial dependence of the original vector components. For example, they may include an infinite sum over radial polynomials to be evaluated.

Proposition 19.2. *The VSH series coefficients are*

$$A_m^{r,\ell}(r) \;=\; \iint \mathbf{A}(r,\theta,\phi)\cdot\left[\mathbf{Y}_\ell^m(\theta,\phi)\right]^* d\Omega, \tag{19.2}$$

$$A_m^{(1),\ell}(r) \;=\; \frac{1}{\ell(\ell+1)}\iint \mathbf{A}(r,\theta,\phi)\cdot\left[\mathbf{\Psi}_\ell^m(\theta,\phi)\right]^* d\Omega, \tag{19.3}$$

$$A_m^{(2),\ell}(r) \;=\; \frac{1}{\ell(\ell+1)}\iint \mathbf{A}(r,\theta,\phi)\cdot\left[\mathbf{\Phi}_\ell^m(\theta,\phi)\right]^* d\Omega. \tag{19.4}$$

Proof. Let us take the dot product of $\mathbf{A}$ with $\left[\mathbf{Y}_{\ell_2}^{m_2}\right]^*$ and integrate over all space. Applying the definition of the VSH series, we obtain

$$\begin{aligned}\iint \mathbf{A}\cdot\left[\mathbf{Y}_{\ell_2}^{m_2}\right]^* d\Omega \;&=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1}\Bigg[A_{m_1}^{r,\ell_1}\iint \mathbf{Y}_{\ell_1}^{m_1}\cdot\left[\mathbf{Y}_{\ell_2}^{m_2}\right]^* d\Omega + A_{m_1}^{(1),\ell_1}\iint \mathbf{\Psi}_{\ell_1}^{m_1}\cdot\left[\mathbf{Y}_{\ell_2}^{m_2}\right]^* d\Omega \\ &\qquad + A_{m_1}^{(2),\ell_1}\iint \mathbf{\Phi}_{\ell_1}^{m_1}\cdot\left[\mathbf{Y}_{\ell_2}^{m_2}\right]^* d\Omega\Bigg] \qquad (19.5)\\ &=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1} A_{m_1}^{r,\ell_1}\,\delta_{\ell_1,\ell_2}^{m_1,m_2} \;=\; A_{m_2}^{r,\ell_2}, \qquad (19.6)\end{aligned}$$

where we used the orthogonality results in Proposition 18.5. This is the expression for $A_m^{r,\ell}$. The expression for $A_m^{(1),\ell}$ is obtained by instead taking the dot product with $\left[\mathbf{\Psi}_{\ell_2}^{m_2}\right]^*$, and similarly the expression for $A_m^{(2),\ell}$ is obtained by taking the dot product with $\left[\mathbf{\Phi}_{\ell_2}^{m_2}\right]^*$. □

Proposition 19.3 (Spherical polar components)**.** *The spherical polar vector components are obtained from the VSH series components by*

$$A_r = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} A_m^{r,\ell}\, Y_\ell^m, \tag{19.7}$$

$$A_\theta = \sum_{\ell=1}^{\infty}\sum_{m=-\ell}^{\ell}\left(A_m^{(1),\ell}\,\frac{\partial Y_\ell^m}{\partial\theta} - A_m^{(2),\ell}\,\frac{1}{\sin(\theta)}\,\frac{\partial Y_\ell^m}{\partial\phi}\right), \tag{19.8}$$

$$A_\phi = \sum_{\ell=1}^{\infty}\sum_{m=-\ell}^{\ell}\left(A_m^{(2),\ell}\,\frac{\partial Y_\ell^m}{\partial\theta} + A_m^{(1),\ell}\,\frac{1}{\sin(\theta)}\,\frac{\partial Y_\ell^m}{\partial\phi}\right). \tag{19.9}$$

Proof. Note that

$$A_\theta = \mathbf{A}\cdot\mathbf{e}_\theta = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell}\left[A_m^{r,\ell}\,\mathbf{Y}_\ell^m\cdot\mathbf{e}_\theta + A_m^{(1),\ell}\,\mathbf{\Psi}_\ell^m\cdot\mathbf{e}_\theta + A_m^{(2),\ell}\,\mathbf{\Phi}_\ell^m\cdot\mathbf{e}_\theta\right] \tag{19.10}$$

$$= \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell}\left[A_m^{(1),\ell}\,\frac{\partial Y_\ell^m}{\partial\theta} + A_m^{(2),\ell}\left(-\frac{1}{\sin(\theta)}\,\frac{\partial Y_\ell^m}{\partial\phi}\right)\right]. \tag{19.11}$$

The results for A_r and A_ϕ follow by evaluating $\mathbf{A}\cdot\mathbf{e}_r$ and $\mathbf{A}\cdot\mathbf{e}_\phi$ respectively. □

Proposition 19.4 (Divergence)**.** *We have*

$$\boldsymbol{\nabla}\cdot\mathbf{A} = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell}\left[\frac{dA_m^{r,\ell}}{dr} + \frac{2}{r}\,A_m^{r,\ell} - \frac{\ell(\ell+1)}{r}\,A_m^{(1),\ell}\right]Y_\ell^m. \tag{19.12}$$

Proof. For all radial functions $f(r)$, we obtain from Eq. (9.10) and Proposition 18.7 that

$$\boldsymbol{\nabla}\cdot\left[f(r)\,\mathbf{Y}_\ell^m\right] = \left(\frac{df}{dr} + \frac{2}{r}\,f\right)Y_\ell^m, \tag{19.13}$$

$$\boldsymbol{\nabla}\cdot\left[f(r)\,\mathbf{\Psi}_\ell^m\right] = -\frac{\ell(\ell+1)}{r}\,f\,Y_\ell^m, \tag{19.14}$$

$$\boldsymbol{\nabla}\cdot\left[f(r)\,\mathbf{\Phi}_\ell^m\right] = 0. \tag{19.15}$$

The given result follows from this and the linearity of the divergence. □

Corollary 19.5. *The VSH expansion of a divergence-free vector field satisfies*

$$A_m^{(1),\ell} = \frac{1}{\ell(\ell+1)}\left(r\,\frac{dA_m^{r,\ell}}{dr} + 2\,A_m^{r,\ell}\right). \tag{19.16}$$

Proof. Set the expression in Proposition 19.4 to zero. This is most easily achieved if

$$\left[\frac{dA_m^{r,\ell}}{dr} + \frac{2}{r}\,A_m^{r,\ell} - \frac{\ell(\ell+1)}{r}\,A_m^{(1),\ell}\right]Y_\ell^m = 0, \qquad \forall\,\ell, m. \tag{19.17}$$

Note that $\nexists\,\ell, m : \forall\,\theta,\phi,\ Y_\ell^m(\theta,\phi) = 0$, so it must be that

$$\frac{dA_m^{r,\ell}}{dr} + \frac{2}{r}\,A_m^{r,\ell} - \frac{\ell(\ell+1)}{r}\,A_m^{(1),\ell} = 0, \qquad \forall\,\ell, m. \tag{19.18}$$

Thus, $A_m^{(1),\ell}$ and $A_m^{r,\ell}$ are inextricably linked. Rearranging for $A_m^{(1),\ell}$, we obtain the given result. □

For a more rigorous proof of Corollary 19.5, see §B.3 of Pétri (2012). As a result, we only require two sets of coefficients $A_m^{r,\ell}$ and $A_m^{(2),\ell}$. This reflects the fact that divergence-free vector fields have only two degrees of freedom. A useful consequence is that

$$\frac{dA_m^{(1),\ell}}{dr} = \frac{1}{\ell(\ell+1)}\left(r\,\frac{d^2A_m^{r,\ell}}{dr^2} + 3\,\frac{dA^{r,\ell}}{dr}\right). \tag{19.19}$$

Proposition 19.6 (Curl)**.** *We have*

$$\boldsymbol{\nabla}\times\mathbf{A} = \sum_{\ell=1}^{\infty}\sum_{m=-\ell}^{\ell}\left[-\frac{\ell(\ell+1)}{r}\,A_m^{(2),\ell}\,\mathbf{Y}_\ell^m - \left(\frac{dA_m^{(2),\ell}}{dr} + \frac{1}{r}\,A_m^{(2),\ell}\right)\boldsymbol{\Psi}_\ell^m + \left(\frac{dA_m^{(1),\ell}}{dr} - \frac{1}{r}\,A_m^{r,\ell} + \frac{1}{r}\,A_m^{(1),\ell}\right)\boldsymbol{\Phi}_\ell^m\right]. \tag{19.20}$$

Proof. For all radial functions $f(r)$, we obtain from Eq. (9.11) and Proposition 18.8 that

$$\boldsymbol{\nabla}\times\left[f(r)\,\mathbf{Y}_\ell^m\right] = -\frac{1}{r}\,f\,\boldsymbol{\Phi}_\ell^m, \tag{19.21}$$

$$\boldsymbol{\nabla}\times\left[f(r)\,\boldsymbol{\Psi}_\ell^m\right] = \left(\frac{df}{dr} + \frac{1}{r}\,f\right)\boldsymbol{\Phi}_\ell^m, \tag{19.22}$$

$$\boldsymbol{\nabla}\times\left[f(r)\,\boldsymbol{\Phi}_\ell^m\right] = -\frac{\ell(\ell+1)}{r}\,f\,\mathbf{Y}_\ell^m - \left(\frac{df}{dr} + \frac{1}{r}\,f\right)\boldsymbol{\Psi}_\ell^m. \tag{19.23}$$

The given result follows from this, the linearity of the curl and that $\mathbf{Y}_0^0$, $\boldsymbol{\Psi}_0^0$, $\boldsymbol{\Phi}_0^0$ have zero curl. □

Proposition 19.7. *If we truncate the VSH series decomposition of a function at $\ell_{\max}$, its VSH series contains $3(\ell_{\max}+1)^2-2$ terms.*

Proof. First, consider only $\mathbf{Y}_\ell^m$. For each ℓ, there are $2\ell+1$ terms due to the possible values of m. (For example, for $\ell=2$ we have 5 terms $\mathbf{Y}_2^{-2}, \mathbf{Y}_2^{-1}, \mathbf{Y}_2^0, \mathbf{Y}_2^1, \mathbf{Y}_2^2$.) Then, the total number of terms up to $\ell_{\max}$ is

$$\sum_{\ell=0}^{\ell_{\max}}(2\ell+1). \tag{19.24}$$

Now, the sum of an arithmetic progression with N terms a_n from $n=1$ to $n=N$ is given by $\frac{1}{2}N$ times the sum of the first and last terms:

$$\sum_{n=1}^{N} a_n = \frac{1}{2}\,N(a_1+a_N). \tag{19.25}$$

We have $N=\ell_{\max}+1$ terms, with first term $2(0)+1=1$ and last term $2\ell_{\max}+1$, giving

$$\sum_{\ell=0}^{\ell_{\max}}(2\ell+1) = \frac{1}{2}\left(\ell_{\max}+1\right)\left(1+2\ell_{\max}+1\right) = \frac{1}{2}\left(\ell_{\max}+1\right)2\left(\ell_{\max}+1\right) = \left(\ell_{\max}+1\right)^2. \tag{19.26}$$

The same applies for $\boldsymbol{\Psi}_\ell^m$ and $\boldsymbol{\Phi}_\ell^m$, multiplying our total by 3, but since $\boldsymbol{\Psi}_0^0 = \boldsymbol{\Phi}_0^0 = \mathbf{0}$ we can subtract two terms from this total. □

19.2 VSH series for operations of vectors

We have seen that, given the VSH series decomposition of a vector function, its divergence and the VSH series coefficients of its curl follow immediately without the need for further integrals. This motivates us to search for other simple operations for which the VSH series of the result is immediately obtainable by summing the already-calculated constant coefficients of the original functions.

If we can find such expressions, it will greatly ease integration of the time-evolution equations in our project by converting them into a set of ODEs for the VSH coefficients themselves, effectively removing all spatial dependence from the problem (see Chapter 21). Our attempt will be unsuccessful; all results still feature the basis functions, so we cannot obtain the VSH series coefficients of the new function by summing previously calculated constant coefficients.

In practice, the best course of action will be to recompute the vector functions via Proposition 19.3, perform the operations in the normal way (except the divergence and curl) and recalculate the VSH decomposition of the result. We appreciate that this second calculation will yield an increase in numerical error.

The following results were not implemented within the project code, but their derivation represented a substantial part of the background work. They may be useful on their own as generalised results, perhaps find applications applications in other areas of numerical modelling, or even save effort for others in search of similar relations.

In the expressions for the dot and cross products, Propositions 19.10 and 19.11, we are able to remove or combine many of the cross-terms. This may result in algorithms faster than simply evaluating the vectors one-at-a-time and taking the dot or cross product of these series, making these expressions of interest for any application. We do not test this claim.

Proposition 19.8 (VSH series of scalar function multiplying vector)**.** *Let $f(r,\theta,\phi)$ be a scalar function and $\mathbf{A}(r,\theta,\phi)$ be a vector with known VSH series expansion. The quantity $f\,\mathbf{A}$ has the following VSH series coefficients in terms of the VSH series coefficients of $\mathbf{A}$. The r-coefficients are*

$$\left[f\,\mathbf{A}\right]^{r,\ell}_{m} = \sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1} A^{r,\ell_1}_{m_1} \iint f\, Y^{m_1}_{\ell_1}\,(Y^{m}_{\ell})^{*}\,d\Omega. \tag{19.27}$$

The (1)-coefficients are

$$\begin{aligned}\left[f\,\mathbf{A}\right]^{(1),\ell}_{m} = \frac{1}{\ell(\ell+1)}\sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1}\Bigg[\; & A^{(1),\ell_1}_{m_1}\iint f\,\frac{\partial Y^{m_1}_{\ell_1}}{\partial\theta}\left(\frac{\partial Y^{m}_{\ell}}{\partial\theta}\right)^{*}d\Omega \\ & + A^{(2),\ell_1}_{m_1}\iint f\,\frac{\partial Y^{m_1}_{\ell_1}}{\partial\theta}\left(\frac{\partial Y^{m}_{\ell}}{\partial\phi}\right)^{*}\frac{1}{\sin(\theta)}\,d\Omega \\ & - A^{(2),\ell_1}_{m_1}\iint f\,\frac{\partial Y^{m_1}_{\ell_1}}{\partial\phi}\left(\frac{\partial Y^{m}_{\ell}}{\partial\theta}\right)^{*}\frac{1}{\sin(\theta)}\,d\Omega \\ & + A^{(1),\ell_1}_{m_1}\iint f\,\frac{\partial Y^{m_1}_{\ell_1}}{\partial\phi}\left(\frac{\partial Y^{m}_{\ell}}{\partial\phi}\right)^{*}\frac{1}{\sin^2(\theta)}\,d\Omega\;\Bigg].\end{aligned} \tag{19.28}$$

The (2)-coefficients are

$$\begin{aligned}\left[f\,\mathbf{A}\right]_m^{(2),\ell} \;=\; \frac{1}{\ell(\ell+1)}\sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1}\Bigg[\;& A_{m_1}^{(2),\ell_1}\iint f\,\frac{\partial Y_{\ell_1}^{m_1}}{\partial\theta}\left(\frac{\partial Y_\ell^m}{\partial\theta}\right)^*\,d\Omega \\ &-A_{m_1}^{(1),\ell_1}\iint f\,\frac{\partial Y_{\ell_1}^{m_1}}{\partial\theta}\left(\frac{\partial Y_\ell^m}{\partial\phi}\right)^*\frac{1}{\sin(\theta)}\,d\Omega \\ &+A_{m_1}^{(1),\ell_1}\iint f\,\frac{\partial Y_{\ell_1}^{m_1}}{\partial\phi}\left(\frac{\partial Y_\ell^m}{\partial\theta}\right)^*\frac{1}{\sin(\theta)}\,d\Omega \\ &+A_{m_1}^{(2),\ell_1}\iint f\,\frac{\partial Y_{\ell_1}^{m_1}}{\partial\phi}\left(\frac{\partial Y_\ell^m}{\partial\phi}\right)^*\frac{1}{\sin^2(\theta)}\,d\Omega\;\Bigg].\end{aligned} \tag{19.29}$$

Proof. The (1)-coefficients are

$$\left[f\,\mathbf{A}\right]_m^{(1),\ell} \;=\; \frac{1}{\ell(\ell+1)}\iint f\,\mathbf{A}\cdot\left(\mathbf{\Psi}_\ell^m\right)^*\,d\Omega \tag{19.30}$$

$$\;=\; \frac{1}{\ell(\ell+1)}\iint f\,A_\theta\left(\frac{\partial Y_\ell^m}{\partial\theta}\right)^*\,d\Omega+\frac{1}{\ell(\ell+1)}\iint f\,A_\phi\,\frac{1}{\sin(\theta)}\left(\frac{\partial Y_\ell^m}{\partial\phi}\right)^*\,d\Omega. \tag{19.31}$$

Substitute the expressions for A_θ and A_ϕ from Proposition 19.3. Since the VSH series coefficients $A_m^{r,\ell}$, $A_m^{(1),\ell}$, $A_m^{(2),\ell}$ are independent of θ,ϕ, they can be taken outside the integrals and the given result follows. The expressions for the r-coefficients and (2)-coefficients are obtained in the same way. □

Verification. `VSH series/VSH_series_fA_01.cpp` □

Lemma 19.9. *The product of two sums is the double sum of the product of their elements. That is, given* $A=\sum_{i=1}^{N}a_i$ *and* $B=\sum_{i=1}^{M}b_i$, *then*

$$AB \;=\; \sum_{i=1}^{N}\sum_{j=1}^{M}a_ib_j. \tag{19.32}$$

Proof. We have

$$AB \;=\; \left(\sum_{i=1}^{N}a_i\right)\left(\sum_{j=1}^{M}b_j\right) \tag{19.33}$$

$$\;=\; \left(a_1+a_2+\cdots+a_N\right)\left(\sum_{j=1}^{M}b_j\right) \tag{19.34}$$

$$\;=\; a_1\sum_{j=1}^{M}b_j+a_2\sum_{j=1}^{M}b_j+\cdots+a_N\sum_{j=1}^{M}b_j \tag{19.35}$$

$$\;=\; \sum_{j=1}^{M}a_i\,b_j+\sum_{j=1}^{M}a_2\,b_j+\cdots+\sum_{j=1}^{M}a_N\,b_j \tag{19.36}$$

$$\;=\; \sum_{i=1}^{N}\left(\sum_{j=1}^{M}a_i\,b_j\right), \tag{19.37}$$

which completes the proof. □

Proposition 19.10 (Dot product of two vector functions)**.** *Let* $\mathbf{A}(r,\theta,\phi)$ *and* $\mathbf{B}(r,\theta,\phi)$ *be vector functions with known VSH series expansions. Then,* $\mathbf{A}\cdot\mathbf{B}$ *can be simplified as*

$$\begin{aligned}\mathbf{A}\cdot\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=-\ell_2}^{\ell_2}\Big[&A_{m_1}^{r,\ell_1}\,B_{m_2}^{r,\ell_2}\,\mathbf{Y}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}\\&+\big[A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(1),\ell_2}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(2),\ell_2}\big]\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}\\&+\big[A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(2),\ell_2}-A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(1),\ell_2}\big]\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}\Big].\end{aligned}\tag{19.38}$$

Proof. Substitute the VSH series expansions for $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{A}\cdot\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\Big[A_{m_1}^{r,\ell_1}\,\mathbf{Y}_{\ell_1}^{m_1}+A_{m_1}^{(1),\ell_1}\,\boldsymbol{\Psi}_{\ell_1}^{m_1}+A_{m_1}^{(2),\ell_1}\,\boldsymbol{\Phi}_{\ell_1}^{m_1}\Big]\cdot\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[B_{m_2}^{r,\ell_2}\,\mathbf{Y}_{\ell_2}^{m_2}+B_{m_2}^{(1),\ell_2}\,\boldsymbol{\Psi}_{\ell_2}^{m_2}+B_{m_2}^{(2),\ell_2}\,\boldsymbol{\Phi}_{\ell_2}^{m_2}\Big].\tag{19.39}$$

Use the distributivity of the dot product, combine the sums and expand the brackets term-by-term:

$$\begin{aligned}\mathbf{A}\cdot\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[&A_{m_1}^{r,\ell_1}\,B_{m_2}^{r,\ell_2}\,\mathbf{Y}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}+A_{m_1}^{r,\ell_1}\,B_{m_2}^{(1),\ell_2}\,\underline{\mathbf{Y}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}}+A_{m_1}^{r,\ell_1}\,B_{m_2}^{(2),\ell_2}\,\underline{\mathbf{Y}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}}\\&+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{r,\ell_2}\,\underline{\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}}+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(1),\ell_2}\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(2),\ell_2}\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}\\&+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{r,\ell_2}\,\underline{\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(1),\ell_2}\,\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(2),\ell_2}\,\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}\Big].\end{aligned}\tag{19.40}$$

We have from Proposition 18.4 that $\mathbf{Y}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}=\mathbf{Y}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}=0$, so the underlined terms are zero, giving

$$\begin{aligned}\mathbf{A}\cdot\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[&A_{m_1}^{r,\ell_1}\,B_{m_2}^{r,\ell_2}\,\mathbf{Y}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}\\&+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(1),\ell_2}\,\underline{\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}}+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(2),\ell_2}\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}\\&+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(1),\ell_2}\,\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(2),\ell_2}\,\underline{\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}}\Big].\end{aligned}\tag{19.41}$$

We also have that $\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{\ell_2}=\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{\ell_2}$, so the underlined terms can be combined, giving

$$\begin{aligned}\mathbf{A}\cdot\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[&A_{m_1}^{r,\ell_1}\,B_{m_2}^{r,\ell_2}\,\mathbf{Y}_{\ell_1}^{m_1}\cdot\mathbf{Y}_{\ell_2}^{m_2}\\&+\big[A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(1),\ell_2}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(2),\ell_2}\big]\,\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}\\&+A_{m_1}^{(1),\ell_1}\,B_{m_2}^{(2),\ell_2}\,\underline{\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{m_2}}+A_{m_1}^{(2),\ell_1}\,B_{m_2}^{(1),\ell_2}\,\underline{\underline{\boldsymbol{\Phi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Psi}_{\ell_2}^{m_2}}}\Big].\end{aligned}\tag{19.42}$$

Finally, we have from Eq. (18.28) that $\boldsymbol{\Psi}_{\ell_1}^{m_1}\cdot\boldsymbol{\Phi}_{\ell_2}^{\ell_2}=-\boldsymbol{\Psi}_{\ell_2}^{m_2}\cdot\boldsymbol{\Phi}_{\ell_1}^{\ell_1}$, so we can swap the indices on the double-underlined terms at the cost of a minus sign and combine them with the single-underlined terms. Doing this, we arrive at the given result. □

Proposition 19.11 (Cross product of two vector functions)**.** *Let* $\mathbf{A}(r,\theta,\phi)$ *and* $\mathbf{B}(r,\theta,\phi)$ *be vector functions with known VSH series expansions. Then,* $\mathbf{A}\times\mathbf{B}$ *can be simplified as*

$$\begin{aligned}\mathbf{A}\times\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=-\ell_1}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=-\ell_2}^{\ell_2}\Bigg[&\\ &\Big(A^{r,\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}-A^{(1),\ell_2}_{m_2}B^{r,\ell_1}_{m_1}\Big)\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}+\Big(A^{r,\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}-A^{(2),\ell_2}_{m_2}B^{r,\ell_1}_{m_1}\Big)\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}\\ &+\Big(A^{(1),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}-A^{(2),\ell_2}_{m_2}B^{(1),\ell_1}_{m_1}\Big)\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}+\Big(A^{(1),\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}+A^{(2),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\Big)\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}\Bigg].\end{aligned} \tag{19.43}$$

Proof. Substitute the VSH series expansions for $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{A}\times\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\Big[A^{r,\ell_1}_{m_1}\,\mathbf{Y}^{m_1}_{\ell_1}+A^{(1),\ell_1}_{m_1}\,\mathbf{\Psi}^{m_1}_{\ell_1}+A^{(2),\ell_1}_{m_1}\,\mathbf{\Phi}^{m_1}_{\ell_1}\Big]\times\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[B^{r,\ell_2}_{m_2}\,\mathbf{Y}^{m_2}_{\ell_2}+B^{(1),\ell_2}_{m_2}\,\mathbf{\Psi}^{m_2}_{\ell_2}+B^{(2),\ell_2}_{m_2}\,\mathbf{\Phi}^{m_2}_{\ell_2}\Big]. \tag{19.44}$$

Use the distributivity of the cross product, combine the sums and expand the brackets term-by-term:

$$\begin{aligned}\mathbf{A}\times\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[&A^{r,\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\underline{\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}}+A^{r,\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}+A^{r,\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}\\ &+A^{(1),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}+A^{(1),\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}\,\underline{\underline{\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}}}+A^{(1),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}\\ &+A^{(2),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}+A^{(2),\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}\,\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}+A^{(2),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\,\underline{\underline{\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}}}\Big].\end{aligned} \tag{19.45}$$

We have from Proposition 18.6 that $\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}=\mathbf{0}$ and $\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}=\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}$, so the single-underlined terms are zero and the double-underlined terms may be combined, giving

$$\begin{aligned}\mathbf{A}\times\mathbf{B} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[&A^{r,\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}+A^{r,\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}\\ &+\underline{A^{(1),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}}+A^{(1),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Phi}^{m_2}_{\ell_2}\\ &+\underline{A^{(2),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2}}+\underline{A^{(2),\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}\,\mathbf{\Phi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}}\\ &+\Big(A^{(1),\ell_1}_{m_1}B^{(1),\ell_2}_{m_2}+A^{(2),\ell_1}_{m_1}B^{(2),\ell_2}_{m_2}\Big)\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}\Big].\end{aligned} \tag{19.46}$$

For the underlined terms, use the anticommutativity of the cross product to switch the order of the vectors at the cost of a minus sign. Then, since the sums over $\{\ell_1,m_1\}$ and $\{\ell_2,m_2\}$ have the same ranges, we can freely relabel the indices $1\leftrightarrow 2$ in these terms. For example,

$$\sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}A^{(1),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{\Psi}^{m_1}_{\ell_1}\times\mathbf{Y}^{m_2}_{\ell_2} \;=\; \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[-A^{(1),\ell_1}_{m_1}B^{r,\ell_2}_{m_2}\,\mathbf{Y}^{m_2}_{\ell_2}\times\mathbf{\Psi}^{m_1}_{\ell_1}\Big] \tag{19.47}$$

$$= \sum_{\ell_1=0}^{\infty}\sum_{m_1=0}^{\ell_1}\sum_{\ell_2=0}^{\infty}\sum_{m_2=0}^{\ell_2}\Big[-A^{(1),\ell_2}_{m_2}B^{r,\ell_1}_{m_1}\,\mathbf{Y}^{m_1}_{\ell_1}\times\mathbf{\Psi}^{m_2}_{\ell_2}\Big]. \tag{19.48}$$

Doing this for the remaining underlined terms and collecting common cross products, we obtain the given result. □

19.3 VSH series of axisymmetric vector functions

Definition 19.12. *An* **axisymmetric function** $f(r,\theta)$ *is one that does not depend on the* ϕ*-coordinate. An* **axisymmetric vector field** $\mathbf{A}(r,\theta)$ *is one whose components* $A_r(r,\theta)$, $A_\theta(r,\theta)$, $A_\phi(r,\theta)$ *do not depend on the* ϕ*-coordinate. An* **axisymmetric vector spherical harmonic series (AVSH series)** *is the VSH series of an axisymmetric vector field, which consists solely of axisymmetric VSHs.*

Consider now the VSH series of an axisymmetric vector field. We obtain the following results as special cases of the expressions obtained in the previous section.

The VSH series for an axisymmetric vector field is

$$\mathbf{A}(r,\theta) \;=\; \frac{A^{r,0}}{\sqrt{4\pi}}\,\mathbf{e}_r + \sum_{\ell=1}^{\infty}\sqrt{\frac{2\ell+1}{4\pi}}\Big[A^{r,\ell}(r)\,P_\ell^0\,\mathbf{e}_r + A^{(1),\ell}(r)\,P_\ell^1\,\mathbf{e}_\theta + A^{(2),\ell}(r)\,P_\ell^1\,\mathbf{e}_\phi\Big], \tag{19.49}$$

with coefficients

$$A^{r,\ell}(r) \;=\; \sqrt{(2\ell+1)\pi}\int_0^\pi A_r\,P_\ell^0\,\sin(\theta)\,d\theta, \tag{19.50}$$

$$A^{(1),\ell}(r) \;=\; \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)}\int_0^\pi A_\theta\,P_\ell^1\,\sin(\theta)\,d\theta, \tag{19.51}$$

$$A^{(2),\ell}(r) \;=\; \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)}\int_0^\pi A_\phi\,P_\ell^1\,\sin(\theta)\,d\theta, \tag{19.52}$$

where we dropped the subscript labelling m because $m=0$ always, making it redundant, and where arguments (r,θ) for the components of $\mathbf{A}$ and $\cos(\theta)$ for the (associated) Legendre functions are implied.

The spherical polar vector components are recovered by

$$A_r \;=\; \sum_{\ell=0}^{\infty}A^{r,\ell}\sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^0, \tag{19.53}$$

$$A_\theta \;=\; \sum_{\ell=1}^{\infty}A^{(1),\ell}\sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^1, \tag{19.54}$$

$$A_\phi \;=\; \sum_{\ell=1}^{\infty}A^{(2),\ell}\sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^1. \tag{19.55}$$

The divergence is

$$\boldsymbol{\nabla}\cdot\mathbf{A} \;=\; \sum_{\ell=0}^{\infty}\sqrt{\frac{2\ell+1}{4\pi}}\,P_\ell^0\left[\frac{dA^{r,\ell}}{dr}+\frac{2}{r}\,A^{r,\ell}-\frac{\ell(\ell+1)}{r}\,A^{(1),\ell}\right]. \tag{19.56}$$

The curl is

$$\begin{aligned}\boldsymbol{\nabla}\times\mathbf{A} \;=\; \sum_{\ell=1}^{\infty}\sqrt{\frac{2\ell+1}{4\pi}}\Bigg[&-\frac{\ell(\ell+1)}{r}\,A^{(2),\ell}\,P_\ell^0\,\mathbf{e}_r-\left(\frac{dA^{(2),\ell}}{dr}+\frac{1}{r}\,A^{(2),\ell}\right)P_\ell^1\,\mathbf{e}_\theta\\&+\left(\frac{dA^{(1),\ell}}{dr}-\frac{1}{r}\,A^{(1),\ell}+\frac{1}{r}\,A^{(1),\ell}\right)P_\ell^1\,\mathbf{e}_\phi\Bigg].\end{aligned} \tag{19.57}$$

Let $f(r,\theta)$ be an axisymmetric scalar function. Then, the AVSH series coefficients of $f\,\mathbf{A}$ are

$$\left[f\,\mathbf{A}\right]^{r,\ell} = \frac{1}{2}\sqrt{2\ell+1}\sum_{\ell_1=0}^{\infty}\sqrt{2\ell_1+1}\,A^{r,\ell_1}\int_0^\pi f\,P^0_{\ell_1}\,P^0_\ell\,\sin(\theta)\,d\theta, \tag{19.58}$$

$$\left[f\,\mathbf{A}\right]^{(1),\ell} = \frac{\sqrt{2\ell+1}}{2\ell(\ell+1)}\sum_{\ell_1=1}^{\infty}\sqrt{2\ell_1+1}\,A^{(1),\ell_1}\int_0^\pi f\,P^1_{\ell_1}\,P^1_\ell\,\sin(\theta)\,d\theta, \tag{19.59}$$

$$\left[f\,\mathbf{A}\right]^{(2),\ell} = \frac{\sqrt{2\ell+1}}{2\ell(\ell+1)}\sum_{\ell_1=1}^{\infty}\sqrt{2\ell_1+1}\,A^{(2),\ell_1}\int_0^\pi f\,P^1_{\ell_1}\,P^1_\ell\,\sin(\theta)\,d\theta. \tag{19.60}$$

Lemma 19.13. *The nonzero cross products of the axisymmetric VSHs have AVSH series coefficients*

$$\left(\mathbf{Y}_{\ell_1}\times\mathbf{\Psi}_{\ell_2}\right)^{(2),\ell} = -\left(\mathbf{Y}_{\ell_1}\times\mathbf{\Phi}_{\ell_2}\right)^{(1),\ell} = \frac{1}{\ell(\ell+1)}\,I^{0,1,1}_{\ell_1,\ell_2,\ell}, \tag{19.61}$$

$$\left(\mathbf{\Psi}_{\ell_1}\times\mathbf{\Phi}_{\ell_2}\right)^{r,\ell} = I^{0,1,1}_{\ell,\ell_1,\ell_2}, \tag{19.62}$$

where we define for brevity that

$$I^{0,1,1}_{\ell_1,\ell_2,\ell_3} \equiv \frac{1}{4\sqrt{\pi}}\sqrt{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}\int_0^\pi P^0_{\ell_1}\,P^1_{\ell_2}\,P^1_{\ell_3}\,\sin(\theta)\,d\theta \tag{19.63}$$

$$= -\frac{1}{\sqrt{\pi}}\sqrt{2\ell_1+1}\sqrt{\ell_2(\ell_2+1)(2\ell_2+1)}\sqrt{\ell_3(\ell_3+1)\,(2\ell_3+1)}\sum_{\ell_{12}=\max\{|\ell_1-\ell_2|,1\}}^{\ell_1+\ell_2}\sum_{\substack{\ell_{123}=\max\{|\ell_{12}-\ell_3|,2\}\\ \ell_{123}\ \text{even}}}^{\ell_{12}+\ell_3}$$
$$\frac{(2\ell_{12}+1)(2\ell_{123}+1)}{\sqrt{(\ell_{123}+2)(\ell_{123}+1)\ell_{123}(\ell_{123}-1)}}\begin{pmatrix}\ell_1 & \ell_2 & \ell_{12}\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\ell_1 & \ell_2 & \ell_{12}\\ 0 & 1 & -1\end{pmatrix}\begin{pmatrix}\ell_{12} & \ell_3 & \ell_{123}\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\ell_{12} & \ell_3 & \ell_{123}\\ 1 & 1 & -2\end{pmatrix}. \tag{19.64}$$

We have that $I^{0,1,1}_{\ell_1,\ell_2,\ell_3}=0$ for $\ell_3>\ell_1+\ell_2$. With this expression, we can immediately obtain the AVSH series coefficients of $\mathbf{A}\times\mathbf{B}$ in terms of the AVSH series coefficients of $\mathbf{A}$ and $\mathbf{B}$ without needing to perform further integrals.

Proof. We have from Proposition 18.6 that $\mathbf{Y}_{\ell_1}\times\mathbf{\Psi}_{\ell_2} = Y_{\ell_1}\,\mathbf{\Phi}_{\ell_2} = \frac{1}{4\pi}\sqrt{(2\ell_1+1)(2\ell_2+1)}\,P^0_{\ell_1}\,P^1_{\ell_2}\,\mathbf{e}_\phi$. Since this is an axisymmetric vector, it must possess an AVSH series. It only has a ϕ-component, so immediately the r-coefficients and (1)-coefficients are zero. The (2)-coefficients are

$$\left(\mathbf{Y}_{\ell_1}\times\mathbf{\Psi}_{\ell_2}\right)^{(2),\ell} = \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)}\int_0^\pi\frac{1}{4\pi}\sqrt{(2\ell_1+1)(2\ell_2+1)}\,P^0_{\ell_1}\,P^1_{\ell_2}\,P^1_\ell\,\sin(\theta)\,d\theta \tag{19.65}$$

$$= \frac{1}{4\sqrt{\pi}}\frac{1}{\ell(\ell+1)}\sqrt{(2\ell+1)(2\ell_1+1)(2\ell_2+1)}\int_0^\pi P^0_{\ell_1}\,P^1_{\ell_2}\,P^1_\ell\,\sin(\theta)\,d\theta. \tag{19.66}$$

The other two nonzero cross products are obtained in the same way. The integral is evaluated as follows. We have $m_1=0$, $m_2=1$ and $m_3=1$, so that $m_{12}=1$ and $m_{123}=2$. Then, by Eqs. (16.51) and (16.53),

$$G_{12}\Big|_{m_1=0,m_2=1} = -(2\ell_{12}+1)\begin{pmatrix}\ell_1 & \ell_2 & \ell_{12}\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\ell_1 & \ell_2 & \ell_{12}\\ 0 & 1 & -1\end{pmatrix}, \tag{19.67}$$

$$G_{123}\Big|_{m_1=0,m_2=1,m_3=1} = +(2\ell_{123}+1)\begin{pmatrix}\ell_{12} & \ell_3 & \ell_{123}\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\ell_{12} & \ell_3 & \ell_{123}\\ 1 & 1 & -2\end{pmatrix}. \tag{19.68}$$

Consider Eq. (16.52). Note the following simplifications:

$$\sqrt{\frac{(\ell_1+m_1)!}{(\ell_1-m_1)!}\frac{(\ell_2+m_2)!}{(\ell_2-m_2)!}\frac{(\ell_3+m_3)!}{(\ell_3-m_3)!}} = \sqrt{\frac{(\ell_1+0)!}{(\ell_1-0)!}\frac{(\ell_2+1)!}{(\ell_2-1)!}\frac{(\ell_3+1)!}{(\ell_3-1)!}} = \sqrt{\ell_2(\ell_2+1)\,\ell_3(\ell_3+1)}, \tag{19.69}$$

$$\sqrt{\frac{(\ell_{123}-m_{123}!)}{(\ell_{123}+m_{123}!)}} = \sqrt{\frac{(\ell_{123}-2)!}{(\ell_{123}+2)!}} = \frac{1}{\sqrt{(\ell_{123}+2)(\ell_{123}+1)\ell_{123}(\ell_{123}-1)}}, \tag{19.70}$$

$$\left[(-1)^{\ell_{123}}+(-1)^{m_{123}}\right]2^{m_{123}-2}\,m_{123} = \left[(-1)^{\ell_{123}}+(-1)^2\right]2^{2-2}\,2 = 2\left[(-1)^{\ell_{123}}+1\right] = 4\,\delta_{\ell_{123}\,(\mathrm{mod}\,2),0}, \tag{19.71}$$

$$\frac{\Gamma\left(\frac{\ell_{123}}{2}\right)\Gamma\left(\frac{\ell_{123}+m_{123}+1}{2}\right)}{\Gamma\left(\frac{\ell_{123}+3}{2}\right)\Gamma\left(\frac{\ell_{123}-m_{123}+2}{2}\right)} = \frac{\Gamma\left(\frac{\ell_{123}}{2}\right)\Gamma\left(\frac{\ell_{123}+2+1}{2}\right)}{\Gamma\left(\frac{\ell_{123}+3}{2}\right)\Gamma\left(\frac{\ell_{123}-2+2}{2}\right)} = \frac{\Gamma\left(\frac{\ell_{123}}{2}\right)\Gamma\left(\frac{\ell_{123}+3}{2}\right)}{\Gamma\left(\frac{\ell_{123}+3}{2}\right)\Gamma\left(\frac{\ell_{123}}{2}\right)} = 1. \tag{19.72}$$

The Kronecker delta symbol can be absorbed into the summation over ℓ_{123} by requiring that ℓ_{123} is even. Substituting these and multiplying by $\frac{1}{4\sqrt{\pi}}\sqrt{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}$, we obtain the given result. □

Proposition 19.14 (AVSH series of cross product)**.** *Let $\mathbf{A}(r,\theta)$ and $\mathbf{B}(r,\theta)$ be two axisymmetric vector functions with known AVSH series expansions. Then, their cross product has AVSH series coefficients which are purely given in terms of the AVSH coefficients of $\mathbf{A}$ and $\mathbf{B}$:*

$$\left[\mathbf{A}\times\mathbf{B}\right]^{r,\ell} = \sum_{\ell_1=1}^{\infty}\sum_{\ell_2=1}^{\infty}\left(A^{(1),\ell_1}B^{(2),\ell_2}-A^{(2),\ell_2}B^{(1),\ell_1}\right)I^{0,1,1}_{\ell,\ell_1,\ell_2}, \tag{19.73}$$

$$\left[\mathbf{A}\times\mathbf{B}\right]^{(1),\ell} = \sum_{\ell_1=0}^{\infty}\sum_{\ell_2=1}^{\infty}\left(A^{(2),\ell_2}B^{r,\ell_1}-A^{r,\ell_1}B^{(2),\ell_2}\right)\frac{1}{\ell(\ell+1)}I^{0,1,1}_{\ell_1,\ell_2,\ell}, \tag{19.74}$$

$$\left[\mathbf{A}\times\mathbf{B}\right]^{(2),\ell} = \sum_{\ell_1=0}^{\infty}\sum_{\ell_2=1}^{\infty}\left(A^{r,\ell_1}B^{(1),\ell_2}-A^{(1),\ell_2}B^{r,\ell_1}\right)\frac{1}{\ell(\ell+1)}I^{0,1,1}_{\ell_1,\ell_2,\ell}, \tag{19.75}$$

with $I^{0,1,1}_{\ell_1,\ell_2,\ell_3}$ defined in Eq. (19.63).

Proof. Proposition 19.11 with $\mathbf{\Psi}_{\ell_1}\times\mathbf{\Psi}_{\ell_2}=\mathbf{0}$ by Proposition 18.6 gives that

$$\begin{aligned}\mathbf{A}\times\mathbf{B} = \sum_{\ell_1=0}^{\infty}\sum_{\ell_2=0}^{\infty}\Big[&\left(A^{r,\ell_1}B^{(1),\ell_2}-A^{(1),\ell_2}B^{r,\ell_1}\right)\mathbf{Y}_{\ell_1}\times\mathbf{\Psi}_{\ell_2}\\ &+\left(A^{r,\ell_1}B^{(2),\ell_2}-A^{(2),\ell_2}B^{r,\ell_1}\right)\mathbf{Y}_{\ell_1}\times\mathbf{\Phi}_{\ell_2}+\left(A^{(1),\ell_1}B^{(2),\ell_2}-A^{(2),\ell_2}B^{(1),\ell_1}\right)\mathbf{\Psi}_{\ell_1}\times\mathbf{\Phi}_{\ell_2}\Big].\end{aligned} \tag{19.76}$$

This is an axisymmetric vector, so must possess an AVSH series. Further, it is a sum of vectors (cross products) with known AVSH series coefficients by Lemma 19.13, so we can immediately read off the AVSH coefficients of $\mathbf{A}\times\mathbf{B}$. The lower bound on ℓ_2 in the sums can be increased to 1 because all pairs of VSH series coefficients that appear feature one of $\{A^{(1),\ell_2}, A^{(2),\ell_2}, B^{(1),\ell_2}, B^{(2),\ell_2}\}$, which are zero for $\ell_2=0$. The lower bound on ℓ_1 in the sum for $\left[\mathbf{A}\times\mathbf{B}\right]^{r,\ell}$ can be increased to 1 for a similar reason. □

Part III

Numerical method

20 Evolutionary equations

20.1 General electrodynamics

Electrodynamics is concerned with modelling the evolution in time t of an electric field $\mathbf{E}(t, \mathbf{r})$ and a magnetic field $\mathbf{B}(t, \mathbf{r})$. Generally, this is described by the **Maxwell equations**

$$\boldsymbol{\nabla} \cdot \mathbf{E} = \frac{\rho_e}{\epsilon_0} \qquad \text{(Gauss's law)}, \tag{20.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{B} = 0 \qquad \text{(Solenoidal condition)}, \tag{20.2}$$

$$\boldsymbol{\nabla} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \qquad \text{(Faraday's law)}, \tag{20.3}$$

$$\boldsymbol{\nabla} \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \qquad \text{(Ampére's law)}, \tag{20.4}$$

where ρ_e is the charge density, the **Lorentz force law**

$$\boldsymbol{\mathcal{L}} = \rho_e \mathbf{E} + \mathbf{J} \times \mathbf{B}, \tag{20.5}$$

and a set of initial conditions. We also require an expression for the current density $\mathbf{J}(t, \mathbf{r})$, which describes the response of the medium to an imposed electromagnetic field. For many systems, we can use **Ohm's law**

$$\mathbf{J} = \sigma \mathbf{E}. \tag{20.6}$$

20.2 Force-free electrodynamics (FFE)

If the Lorentz force vanishes and all non-magnetic forces are negligible, we enter a regime known as **force-free electrodynamics (FFE)**. This is characterised by two conditions, known as the **force-free conditions**, which are fulfilled at all times:

1. **Degeneracy condition:** The electric and magnetic fields are orthogonal, $\mathbf{E} \cdot \mathbf{B} = 0$. As a result, charged particles in the magnetosphere flow exactly along magnetic field lines.
2. The electric field is weaker than the magnetic field, $B^2 - \frac{1}{c^2} E^2 > 0$.

There are many cases in which the Lorentz force is negligible. For example, highly conducting plasmas might have such a high magnetic flux density that any hydrodynamic forces become negligible. This is true if the magnetic energy is greater than the average particle energy:

$$\frac{B^2}{2\mu_0} \gg \rho_m c^2, \tag{20.7}$$

where ρ_m is the mass density.

Let us define the **force-free region** as the region in which the force-free conditions are satisfied for a given magnetic and electric field. For a rotating dipole, this is all points inside the last closed magnetic field line (Proposition 22.4).

Time evolution does not maintain these conditions automatically, so numerical codes must enforce them at each timestep. In our code, we will use the same prescription as described in §3.2 of Pétri (2012) and §3.8 of Parfrey (2012), in which we iterate through all gridpoints within the force-free region and modify the electric field at all points where either condition is not satisfied. This has the added advantage of maintaining $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$. However, the new values of $\mathbf{E}$ will "only just" fulfil the force-free conditions in that the inequalities are saturated, giving little protection against the conditions being again violated in the next timestep. We expect this method to only delay a breakdown of force-free conditions, as opposed to preventing it altogether.

Proposition 20.1. *If $\rho_e \neq 0$, the degeneracy condition and that $\mathbf{E}\cdot\mathbf{J}=0$ follow immediately from setting $\boldsymbol{\mathcal{L}}=0$. If $\rho_e=0$, these become independent basic equations of FFE.*

Proof. Let $\rho_e \neq 0$. If $\boldsymbol{\mathcal{L}} = \rho_e\,\mathbf{E} + \mathbf{J}\times\mathbf{B} = \mathbf{0}$, then $\mathbf{E} = -\frac{1}{\rho_e}\,\mathbf{J}\times\mathbf{B}$. This implies that $\mathbf{E}\perp\mathbf{J}\perp\mathbf{B}$. If $\rho_e = 0$, we instead obtain $\mathbf{J}\times\mathbf{B}=0$; this does not imply $\mathbf{E}\cdot\mathbf{B}=\mathbf{E}\cdot\mathbf{J}=0$ and we must set these as separate conditions. □

Lemma 20.2. *Consider two vectors $\mathbf{A}$ and $\mathbf{B}$. We can decompose $\mathbf{A}$ into components parallel and perpendicular to $\mathbf{B}$, so that $\mathbf{A}=\mathbf{A}_\parallel+\mathbf{A}_\perp$, where*

$$\mathbf{A}_\parallel \;=\; \frac{\mathbf{A}\cdot\mathbf{B}}{B^2}\,\mathbf{B}, \tag{20.8}$$

$$\mathbf{A}_\perp \;=\; -\frac{(\mathbf{A}\times\mathbf{B})\times\mathbf{B}}{B^2}. \tag{20.9}$$

Notice that $\mathbf{A}_\parallel$ is just a scalar multiple of $\mathbf{B}$, which is to be expected if they both point in the same direction.

Proof. For $\mathbf{A}_\parallel$, we have

$$\mathbf{A}_\parallel\times\mathbf{B} \;=\; \frac{\mathbf{A}\cdot\mathbf{B}}{B^2}\,\mathbf{B}\times\mathbf{B} \;=\; \mathbf{0}, \tag{20.10}$$

that is, $\mathbf{A}_\parallel$ is either parallel or antiparallel to $\mathbf{B}$. But clearly

$$\mathbf{A}_\parallel\cdot\mathbf{B} \;=\; \frac{\mathbf{A}\cdot\mathbf{B}}{B^2}\,\mathbf{B}\cdot\mathbf{B} \;=\; \frac{\mathbf{A}\cdot\mathbf{B}}{B^2}\,B^2 \;=\; \mathbf{A}\cdot\mathbf{B} \;\geq\; 0, \tag{20.11}$$

so $\mathbf{A}_\parallel$ cannot be antiparallel to $\mathbf{B}$, so it must be parallel. Recalling the cyclic rule for the scalar triple product,

$$(\mathbf{a}\times\mathbf{b})\cdot\mathbf{c} \;=\; (\mathbf{b}\times\mathbf{c})\cdot\mathbf{a} \;=\; (\mathbf{c}\times\mathbf{a})\cdot\mathbf{b}, \tag{20.12}$$

we can write

$$\mathbf{A}_\perp\cdot\mathbf{B} \;=\; -\frac{1}{B^2}\big([\mathbf{A}\times\mathbf{B}]\times\mathbf{B}\big)\cdot\mathbf{B} \;=\; -\frac{1}{B^2}\big(\mathbf{B}\times\mathbf{B}\big)\cdot[\mathbf{A}\times\mathbf{B}] \;=\; -\frac{1}{B^2}\big(\mathbf{0}\times[\mathbf{A}\times\mathbf{B}]\big) \;=\; \mathbf{0}, \tag{20.13}$$

so $\mathbf{A}_\perp$ is perpendicular to $\mathbf{B}$. Finally, we require that $B^2(\mathbf{A}_\parallel+\mathbf{A}_\perp)=\mathbf{A}$. Recalling the vector triple product

$$(\mathbf{a}\times\mathbf{b})\times\mathbf{c} \;=\; (\mathbf{c}\cdot\mathbf{a})\,\mathbf{b} - (\mathbf{c}\cdot\mathbf{b})\,\mathbf{a}, \tag{20.14}$$

so that $(\mathbf{A}\times\mathbf{B})\times\mathbf{B} = (\mathbf{A}\cdot\mathbf{B})\,\mathbf{B} - B^2\,\mathbf{A}$, we can write

$$\mathbf{A}_\perp = \frac{\mathbf{A}\cdot\mathbf{B}}{B^2}\,\mathbf{B} - \mathbf{A} \;=\; \mathbf{A}_\parallel - \mathbf{A}, \tag{20.15}$$

completing the proof. □

Proposition 20.3. *In FFE, the current is uniquely determined to be*

$$\mathbf{J} = \left(\frac{1}{\mu_0}\,\mathbf{B}\cdot\boldsymbol{\nabla}\times\mathbf{B} - \epsilon_0\,\mathbf{E}\cdot\boldsymbol{\nabla}\times\mathbf{E}\right)\frac{\mathbf{B}}{B^2} + \frac{\boldsymbol{\nabla}\cdot\mathbf{E}}{B^2}\,\mathbf{E}\times\mathbf{B}. \tag{20.16}$$

Proof. Use Lemma 20.2. In this case, $\mathbf{J}_\parallel$ is determined by requiring the degeneracy condition to be maintained at all times:

$$\begin{aligned} 0 &= \frac{\partial}{\partial t}(\mathbf{E}\cdot\mathbf{B}) && (20.17)\\ &= \frac{\partial\mathbf{E}}{\partial t}\cdot\mathbf{B} + \mathbf{E}\cdot\frac{\partial\mathbf{B}}{\partial t} && (20.18)\\ &= \left(\frac{1}{\mu_0\epsilon_0}\,\boldsymbol{\nabla}\times\mathbf{B} - \frac{1}{\epsilon_0}\,\mathbf{J}\right)\cdot\mathbf{B} + \mathbf{E}\cdot(-\boldsymbol{\nabla}\times\mathbf{E}) && (20.19)\\ &= \frac{1}{\mu_0\epsilon_0}\,\mathbf{B}\cdot\boldsymbol{\nabla}\times\mathbf{B} - \frac{1}{\epsilon_0}\,\mathbf{J}\cdot\mathbf{B} - \mathbf{E}\cdot\boldsymbol{\nabla}\times\mathbf{E}, && (20.20) \end{aligned}$$

which rearranges to

$$\mathbf{J}\cdot\mathbf{B} = \frac{1}{\mu_0}\,\mathbf{B}\cdot\boldsymbol{\nabla}\times\mathbf{B} - \epsilon_0\,\mathbf{E}\cdot\boldsymbol{\nabla}\times\mathbf{E}, \tag{20.21}$$

giving an expression for $\mathbf{J}_\parallel$. We could find the perpendicular component $\mathbf{J}_\perp$ in the same way, but a simpler method is to take the cross product of the Lorentz force law (20.5), which has been set to zero, with $\mathbf{B}$:

$$\begin{aligned} \mathbf{J}\times\mathbf{B} &= -\rho_e\,\mathbf{E}, && (20.22)\\ \Rightarrow\ (\mathbf{J}\times\mathbf{B})\times\mathbf{B} &= -\rho_e\,\mathbf{E}\times\mathbf{B} = -(\boldsymbol{\nabla}\cdot\mathbf{E})\,\mathbf{E}\times\mathbf{B}, && (20.23)\\ \Rightarrow\ \mathbf{J}_\perp &= \frac{(\boldsymbol{\nabla}\cdot\mathbf{E})\,\mathbf{E}\times\mathbf{B}}{B^2}. && (20.24) \end{aligned}$$

Combining $\mathbf{J}_\parallel$ and $\mathbf{J}_\perp$, we obtain the given expression for the current vector $\mathbf{J}$. □

The above expression for $\mathbf{J}$ agrees with Eq. (4) of Gruzinov (1999) (in cgs units) and Eq. (7) of Pétri (2012). The component of $\mathbf{J}$ parallel to the magnetic field

$$\mathbf{J}_\parallel = \left(\frac{1}{\mu_0}\,\mathbf{B}\cdot\boldsymbol{\nabla}\times\mathbf{B} - \epsilon_0\,\mathbf{E}\cdot\boldsymbol{\nabla}\times\mathbf{E}\right)\frac{\mathbf{B}}{B^2} \tag{20.25}$$

maintains the degeneracy condition. The perpendicular component is also known as the **drift current** because it can be expressed in terms of the the velocity of the magnetic field lines $\mathbf{v}_{\text{drift}} = \frac{1}{B^2}\,\mathbf{E}\times\mathbf{B}$ (Gruzinov, 1999):

$$\mathbf{J}_\perp = \rho_e\,\mathbf{v}_{\text{drift}} = \frac{\boldsymbol{\nabla}\cdot\mathbf{E}}{B^2}\,\mathbf{E}\times\mathbf{B}. \tag{20.26}$$

Proposition 20.4. *The requirement* $B^2 - \frac{1}{c^2}E^2 > 0$ *is equivalent to the requirement that* $v_{\text{drift}} < c$.

Proof. The magnitude of a cross product is less than or equal to the product of the magnitudes of the two individual vectors, so we can write

$$v_{\text{drift}} \leq \frac{EB}{B^2} = \frac{E}{B}. \tag{20.27}$$

Then, the condition that the drift velocity is less than the speed of light can be written

$$c > v_{\text{drift}} > \frac{E}{B}, \tag{20.28}$$

$$\Rightarrow B > \frac{E}{c}, \tag{20.29}$$

$$\Rightarrow B^2 > \frac{E^2}{c^2}, \tag{20.30}$$

$$\Rightarrow B^2 - \frac{E^2}{c^2} > 0. \tag{20.31}$$

□

The fact that FFE makes the current become uniquely determined is one of its most attractive features because it allows us to directly state our time evolution equations. Ampére's law and that $\frac{1}{\mu_0\epsilon_0} = c^2$ give

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{c^2}\boldsymbol{\nabla} \times \mathbf{B} - \frac{1}{\epsilon_0}\mathbf{J}. \tag{20.32}$$

Using this, our expression for the current vector $\mathbf{J}$ and Faraday's law, the expressions that we wish to evolve over time are

$$\frac{\partial}{\partial t}\mathbf{B} = -\boldsymbol{\nabla} \times \mathbf{E}, \tag{20.33}$$

$$\frac{\partial}{\partial t}\mathbf{E} = c^2\,\boldsymbol{\nabla} \times \mathbf{B} + \frac{\mathbf{E}\cdot\boldsymbol{\nabla}\times\mathbf{E} - c^2\,\mathbf{B}\cdot\boldsymbol{\nabla}\times\mathbf{B}}{B^2}\mathbf{B} - \frac{\boldsymbol{\nabla}\cdot\mathbf{E}}{B^2}\mathbf{E}\times\mathbf{B}. \tag{20.34}$$

20.3 Non-dimensionalisation

Physical quantities can have vastly different orders of magnitude from each other, which can introduce floating point errors in numerical codes. To counteract this, a common technique is to introduce **dimensionless units** in which we scale the physical quantities relative to some convenient measure, so that they all become of order unity.

In this section, we shall represent non-dimensionalised parameters with a tilde, e.g. t is the physical time coordinate and $\tilde{t}$ is the non-dimensionalised time coordinate. We choose the following non-dimensional units:

1. Rescale lengths to the radius of the neutron star $R_\star$, so that $r \equiv R_\star\,\tilde{r}$.
2. Rescale times to the **radial light-crossing time** $\tau = \frac{R_\star}{c}$, so that $t \equiv \tau\,\tilde{t} \;=\; \frac{R_*}{c}\,\tilde{t}$.
3. $B \equiv B_0\,\tilde{B}$, where $B_0 \equiv \frac{\mu_0}{4\pi}\,\frac{m_*}{R_*^3}$ and where m_* is the magnetic dipole moment of the NS. Typical values are $10^{31} - 10^{33}$ A m^2 (Coelho & Malheiro, 2014).
4. $E \equiv cB_0\,\tilde{E}$.

Non-dimensionalised magnetic and electric field vectors can then be defined as $\mathbf{B} = B_0\,\tilde{\mathbf{B}}$ and $\mathbf{E} = cB_0\,\tilde{\mathbf{E}}$. For a scalar function $f(t,r)$, it follows from the chain rule that

$$\frac{\partial f}{\partial r} \;=\; \frac{\partial \tilde{r}}{\partial r}\,\frac{\partial f}{\partial \tilde{r}} \;=\; \frac{\partial}{\partial r}\left(\frac{r}{R_\star}\right)\frac{\partial f}{\partial \tilde{r}} \;=\; \frac{1}{R_\star}\,\frac{\partial f}{\partial \tilde{r}}, \tag{20.35}$$

$$\frac{\partial f}{\partial t} \;=\; \frac{\partial \tilde{t}}{\partial t}\,\frac{\partial f}{\partial \tilde{t}} \;=\; \frac{\partial}{\partial t}\left(\frac{1}{\tau}\,t\right)\frac{\partial f}{\partial \tilde{r}} \;=\; \frac{1}{\tau}\,\frac{\partial f}{\partial \tilde{t}}. \tag{20.36}$$

From these and the definitions in Eqs. (9.3) and (9.4), we can define the non-dimensionalised divergence and curl of a vector function $\mathbf{A}$:

$$\tilde{\boldsymbol{\nabla}}\cdot\mathbf{A} \;\equiv\; \frac{\partial A_r}{\partial \tilde{r}} + \frac{2}{\tilde{r}}\,A_r + \frac{1}{\tilde{r}}\,\frac{\partial A_\theta}{\partial \theta} + \frac{\cos(\theta)}{\tilde{r}\,\sin(\theta)}\,A_\theta + \frac{1}{\tilde{r}\,\sin(\theta)}\,\frac{\partial A_\phi}{\partial \phi}, \tag{20.37}$$

$$\begin{aligned}\tilde{\boldsymbol{\nabla}}\times\mathbf{A} \;\equiv\; &\left[\frac{1}{\tilde{r}}\,\frac{\partial A_\phi}{\partial \theta} + \frac{\cos(\theta)}{\tilde{r}\,\sin(\theta)}\,A_\phi - \frac{1}{\tilde{r}\,\sin(\theta)}\,\frac{\partial A_\theta}{\partial \phi}\right]\mathbf{e}_r + \left[\frac{1}{\tilde{r}\,\sin(\theta)}\,\frac{\partial A_r}{\partial \phi} - \frac{\partial A_\phi}{\partial \tilde{r}} - \frac{1}{\tilde{r}}\,A_\phi\right]\mathbf{e}_\theta \\ &+ \left[\frac{\partial A_\theta}{\partial \tilde{r}} + \frac{1}{\tilde{r}}\,A_\theta - \frac{1}{\tilde{r}}\,\frac{\partial A_r}{\partial \theta}\right]\mathbf{e}_\phi.\end{aligned} \tag{20.38}$$

Note the simple relation between these and the fully dimensional derivatives:

$$\boldsymbol{\nabla}\cdot\mathbf{A} \;=\; \frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\cdot\mathbf{A}, \tag{20.39}$$

$$\boldsymbol{\nabla}\times\mathbf{A} \;=\; \frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\mathbf{A}. \tag{20.40}$$

Proposition 20.5. *The non-dimensionalised system of equations is*

$$\frac{\partial}{\partial \tilde{t}}\,\tilde{\mathbf{B}} \;=\; -\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{E}}, \tag{20.41}$$

$$\frac{\partial}{\partial \tilde{t}}\,\tilde{\mathbf{E}} \;=\; \tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}} + \frac{\tilde{\mathbf{E}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{E}} - \tilde{\mathbf{B}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}}}{\tilde{B}^2}\,\tilde{\mathbf{B}} - \frac{\tilde{\boldsymbol{\nabla}}\cdot\tilde{\mathbf{E}}}{\tilde{B}^2}\,\tilde{\mathbf{E}}\cdot\tilde{\mathbf{B}} \tag{20.42}$$

Proof. Consider Faraday's law first. The left-hand side becomes

$$\text{LHS} \;=\; \frac{c}{R_\star}\,\frac{\partial}{\partial \tilde{t}}\left(B_0\,\tilde{\mathbf{B}}\right) \;=\; \frac{cB_0}{R_\star}\,\frac{\partial}{\partial \tilde{t}}\,\tilde{\mathbf{B}}, \tag{20.43}$$

and the right-hand side becomes

$$\text{RHS} \;=\; -\frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\left(cB_0\,\tilde{\mathbf{E}}\right) \;=\; -\frac{cB_0}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{E}}. \tag{20.44}$$

Equating these and letting the common factor $\frac{cB_0}{R_\star}$ cancel, we obtain the non-dimensionalised version of the equation. Now consider Ampére's law. The left-hand side becomes

$$\text{LHS} \;=\; \frac{c}{R_\star}\,\frac{\partial}{\partial\tilde{t}}\left(cB_0\,\tilde{\mathbf{E}}\right) \;=\; \frac{c^2B_0}{R_\star}\,\frac{\partial}{\partial\tilde{t}}\,\tilde{\mathbf{E}}, \tag{20.45}$$

and the right-hand side becomes

$$\begin{aligned}\text{RHS} \;=\; & c^2\,\frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\left(B_0\,\tilde{\mathbf{B}}\right)+\frac{\left(cB_0\,\tilde{\mathbf{E}}\right)\cdot\frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\left(cB_0\,\tilde{\mathbf{E}}\right)-c^2\left(B_0\,\tilde{\mathbf{B}}\right)\cdot\frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\left(B_0\,\tilde{\mathbf{B}}\right)}{\left(B_0\,\tilde{B}\right)^2}\left(B_0\,\tilde{\mathbf{B}}\right)\\ & -\frac{\frac{1}{R_\star}\,\tilde{\boldsymbol{\nabla}}\cdot\left(cB_0\,\tilde{\mathbf{E}}\right)}{\left(B_0\,\tilde{B}\right)^2}\left(cB_0\,\tilde{\mathbf{E}}\right)\times\left(B_0\,\tilde{\mathbf{B}}\right) \qquad (20.46)\\ =\; & \frac{c^2B_0}{R_\star}\,\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}}+\frac{\frac{c^2B_0^2}{R_\star}\,\tilde{\mathbf{E}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{E}}-\frac{c^2B_0^2}{R_\star}\,\tilde{\mathbf{B}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}}}{B_0^2\,\tilde{B}^2}\,B_0\,\tilde{\mathbf{B}}-\frac{\frac{cB_0}{R_\star}\,\tilde{\boldsymbol{\nabla}}\cdot\tilde{\mathbf{E}}}{B_0^2\,\tilde{B}^2}\,cB_0^2\,\tilde{\mathbf{E}}\times\tilde{\mathbf{B}} \qquad (20.47)\\ =\; & \frac{c^2B_0}{R_\star}\left(\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}}+\frac{\tilde{\mathbf{E}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{E}}-\tilde{\mathbf{B}}\cdot\tilde{\boldsymbol{\nabla}}\times\tilde{\mathbf{B}}}{\tilde{B}^2}\,\tilde{\mathbf{B}}-\frac{\tilde{\boldsymbol{\nabla}}\cdot\tilde{\mathbf{E}}}{\tilde{B}^2}\,\tilde{\mathbf{E}}\times\tilde{\mathbf{B}}\right). \qquad (20.48)\end{aligned}$$

Equating these and letting the common factor $\frac{c^2B_0}{R_\star}$ cancel, we obtain the non-dimensionalised version of the equation. This completes the proof. □

An object moving a distance $r = R_\star\,\tilde{r}$ in time $t = \frac{R_\star}{c}\,\tilde{t}$ has speed $v = \frac{r}{t}$. Defining the dimensionless speed as $\tilde{v} = \frac{\tilde{r}}{\tilde{t}}$, we see that

$$v \;=\; c\,\tilde{v}. \tag{20.49}$$

For light, we have $v_{\text{light}} = c = c\,\tilde{v}_{\text{light}}$ and so the speed of light in our dimensionless units is $\tilde{v}_{\text{light}} = 1$.

If the NS has rotational period P in SI units and $\tilde{P}$ in code units such that $P = \tau\,\tilde{P}$, its angular velocity in SI units and code units is respectively

$$\Omega \;=\; \frac{2\pi}{P}, \tag{20.50}$$

$$\tilde{\Omega} \;=\; \frac{2\pi}{\tilde{P}}, \tag{20.51}$$

and so the values in SI and code units are related by

$$\Omega \;=\; \frac{2\pi}{\tau\tilde{P}} \;=\; \frac{1}{\tau}\,\tilde{\Omega}. \tag{20.52}$$

The **light cylinder** R_{LC} is the radial distance at which an object co-rotating with the star would be travelling at the speed of light. Linear velocity v and angular velocity Ω at radius r are related by $v = r\Omega$, so setting $v = c$ gives

$$R_{\text{LC}} \;=\; \frac{c}{\Omega}. \tag{20.53}$$

Then, in dimensionless units, $R_{\text{LC}} = R_\star\,\tilde{R}_{\text{LC}}$ and $\Omega = \frac{1}{\tau}\,\tilde{\Omega} = \frac{c}{R_\star}\,\tilde{\Omega}$, so

$$\tilde{R}_{\text{LC}} \;=\; \frac{1}{R_\star}\,\frac{c}{\frac{c}{R_\star}\,\tilde{\Omega}} \;=\; \frac{c}{R_\star}\,\frac{R_\star}{c}\,\frac{1}{\tilde{\Omega}} \;=\; \frac{1}{\tilde{\Omega}}, \tag{20.54}$$

and the values in SI and code units are related by

$$R_{\text{LC}} \;=\; \frac{c}{\frac{1}{\tau}\,\tilde{\Omega}} \;=\; c\,\tau\,\tilde{R}_{\text{LC}} \;=\; R_\star\,\tilde{\Omega}_{\text{LC}}, \tag{20.55}$$

as we may expect.

From now on, we will use non-dimensionalised quantities exclusively and refer to them without the tildes.

20.4 Inner boundary conditions

The inner boundary conditions are (Pétri, 2012, §3.3)

$$B_r(t, R_\star, \theta) = B_r^\star(t, \theta), \tag{20.56}$$
$$B_\theta(t, R_\star, \theta) = -R_\star\, \Omega(t)\, \sin(\theta)\, B_r^\star(t, \theta), \tag{20.57}$$
$$E_\phi(t, R_\star, \theta) = 0, \tag{20.58}$$

where $B_r^\star(t, \theta)$ is the radial magnetic field imposed by the star. For a dipole, Eq. (22.2) gives

$$B_r^\star(t, \theta) = \frac{2\cos(\theta)}{R_\star^3}, \tag{20.59}$$

which is constant in time. The remaining components $B_\phi(t, R_\star, \theta)$, $E_r(t, R_\star, \theta)$ and $E_\theta(t, R_\star, \theta)$ must be allowed to evolve freely (Parfrey, 2012, §3.9.1).

Whether the star is rotating or not, we allow the radial components to evolve freely at the outer boundary.

The boundary conditions are applied by the functions

```
void apply_inner_boundary_conditions()
void apply_outer_boundary_conditions()
```

which are enforced at every timestep. To check the effect of our implementation of boundary conditions on the evolution of the system, we save the field components at each coordinate immediately before and after their application and output them to the `_4_BCs` csv.

20.5 Outer boundary conditions

Our ideal simulation is that of a neutron star in an infinite domain, so we should expect electrodynamic waves to only leave the star and none to move toward it. Numerical domains are necessarily finite, necessitating an artificial outer boundary to be placed at some finite radial distance $r_{\max}$. Waves reaching the outer boundary may be erroneously reflected back into the system.

The outer boundary conditions are those of outgoing spherical waves. That is, we allow energy to be dissipated by the system but forbid waves to be reflected back toward the star. To forbid incoming (reflected) waves, we require (Pétri, 2012, §3.3)

$$E_\theta(t, r_{\max}, \theta) - c\, B_\phi(t, r_{\max}, \theta) = 0, \tag{20.60}$$
$$E_\phi(t, r_{\max}, \theta) + c\, B_\theta(t, r_{\max}, \theta) = 0, \tag{20.61}$$
$$E_\theta(t, r_{\max}, \theta) + c\, B_\phi(t, r_{\max}, \theta) = E_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta) + c\, B_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta), \tag{20.62}$$
$$E_\phi(t, r_{\max}, \theta) - c\, B_\theta(t, r_{\max}, \theta) = E_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta) - c\, B_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta), \tag{20.63}$$

where the superscript $^{\mathrm{PDE}}$ refers to the values of the fields that would be calculated by integrating our system of equations without applying outer boundary conditions. Adding and subtracting these four simultaneous equations, we obtain

$$B_\theta(t, r_{\max}, \theta) = \frac{1}{2}\left[B_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta) - \frac{1}{c}\, E_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta)\right], \tag{20.64}$$
$$B_\phi(t, r_{\max}, \theta) = \frac{1}{2}\left[B_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta) + \frac{1}{c}\, E_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta)\right], \tag{20.65}$$
$$E_\theta(t, r_{\max}, \theta) = c\, B_\phi(t, r_{\max}, \theta) = \frac{1}{2}\left[c\, B_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta) + E_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta)\right], \tag{20.66}$$
$$E_\phi(t, r_{\max}, \theta) = -c\, B_\theta(t, r_{\max}, \theta) = \frac{1}{2}\left[-c\, B_\theta^{\mathrm{PDE}}(t, r_{\max}, \theta) + E_\phi^{\mathrm{PDE}}(t, r_{\max}, \theta)\right]. \tag{20.67}$$

However, if the star is not rotating, $\mathbf{E}^{\rm PDE}(t,\mathbf{r}) = \mathbf{0}$ and these equations reduce to

$$B_\theta(t, r_{\rm max}, \theta) = \frac{1}{2} B_\theta^{\rm PDE}(t, r_{\rm max}, \theta), \tag{20.68}$$

$$B_\phi(t, r_{\rm max}, \theta) = \frac{1}{2} B_\phi^{\rm PDE}(t, r_{\rm max}, \theta), \tag{20.69}$$

$$E_\theta(t, r_{\rm max}, \theta) = c\, B_\phi(t, r_{\rm max}, \theta) = \frac{1}{2} c\, B_\phi^{\rm PDE}(t, r_{\rm max}, \theta), \tag{20.70}$$

$$E_\phi(t, r_{\rm max}, \theta) = -c\, B_\theta(t, r_{\rm max}, \theta) = -\frac{1}{2} c\, B_\theta^{\rm PDE}(t, r_{\rm max}, \theta), \tag{20.71}$$

implying that the magnetic field at the outer boundary should become half its previous value at each timestep, and that the electric field at the outer boundary should be equal to the magnetic field. This is true for B_ϕ and hence E_θ, but is only true for B_θ and hence E_ϕ as $r_{\rm max} \to \infty$ because $B_r \sim \frac{1}{r^3}$. We are then at risk of a diminishing B_θ and, if the evolution is not exact, nonzero B_ϕ, E_θ, E_ϕ at the outer boundary. To counter this, we introduce specific outer boundaries for all times at which the star is not rotating:

$$B_\theta(t, r_{\rm max}, \theta) = B_\theta^0(t, r_{\rm max}, \theta), \tag{20.72}$$

$$B_\phi(t, r_{\rm max}, \theta) = B_\phi^0(t, r_{\rm max}, \theta), \tag{20.73}$$

$$E_\theta(t, r_{\rm max}, \theta) = 0, \tag{20.74}$$

$$E_\phi(t, r_{\rm max}, \theta) = 0, \tag{20.75}$$

where $B_\theta^0(t, r_{\rm max}, \theta)$ and $B_\phi^0(t, r_{\rm max}, \theta)$ are the initial field components. For a dipole with $A(\Psi) = 0$, Eq. (22.2) gives that $B_\theta^0(t, r_{\rm max}, \theta) = \frac{\sin(\theta)}{r_{\rm max}^3}$ and $B_\phi^0(t, r_{\rm max}, \theta) = 0$.

20.6 Field lines

Field lines are lines in 3D space showing the direction in which a vector field is pointing as a function of position. They are only mathematical constructions, but can serve useful purposes. For example, if the field represents a force, the lines represent the direction in which a particle will move if it is subject only to that force. Divergenceless fields feature no sources or sinks, so they must have closed field lines. Open field lines imply that energy is lost to the surroundings, perhaps in the form of particles being ejected from the system.

When we model neutron star magnetospheres with force-free electrodynamics, we expect charged particles emitted from the neutron star surface to flow along the magnetic field lines; those that are emitted along open field lines will be ejected from the neutron star, forming the the beams of radiation near the poles that characterise pulsars but also the spontaneous bursts of radiation in giant flares.

It will be useful to develop a method to plot the field lines of an arbitrary magnetic field; however, we will argue in this section that such a goal is unrealistic in our model.

Field lines are created by drawing a curve that is tangent to the field at each point. The unit vector in the direction of the force is $\mathbf{e}_F = \frac{1}{F}\mathbf{F}$. If we start our field line at some arbitrary point $\mathbf{r}_i$, then we can draw a straight line of length δ, whose value can be freely chosen, to some new point

$$\mathbf{r}_{i+1} \;=\; \mathbf{r}_i + \delta\,\mathbf{e}_F \;=\; \mathbf{r}_i + \delta\,\frac{1}{F(\mathbf{r}_i)}\,\mathbf{F}(\mathbf{r}_i). \tag{20.76}$$

The process is repeated as many times as necessary until the field line has reached a satisfactory final coordinate $\mathbf{r}_{\text{final}}$. Because divergenceless fields feature closed field lines, we would then expect the coordinates to return to the original point $\mathbf{r}_{\text{final}} = \mathbf{r}_0$. The smaller the value of δ chosen, the more accurate the field lines will be, analogous to the step size used in numerical differentiation by finite differencing (Chapter 12).

Despite giving a clear description of how to generate field lines, our definition has highlighted three reasons preventing us from constructing field lines in our evolutionary model:

1. As we will see when constructing the magnetic field lines of a dipole in Figure 22.1, accurate field lines are only produced when δ is very small, necessitating the use of a very large number of gridpoints in the domain.

2. The choice of $\mathbf{r}_{\text{final}}$ is difficult to automate. We could simply end the field line after a certain number of points, but this will need to scale inversely to δ to prevent higher resolution (smaller δ) leading to lines which do not extend so far from their start point $\mathbf{r}_0$. For divergenceless fields, we could add a condition to end the field line once it returns within a certain proximity of $\mathbf{r}_0$, but this may never occur or only after a very large number of iterations.

3. Our evolution features discrete grids, so there is no guarantee that $\mathbf{r}_{i+1} = \mathbf{r}_i + \delta\,\mathbf{e}_F$ will coincide with a gridpoint. We will be forced to use interpolation to estimate $\mathbf{F}_{i+1}$ from the nearest surrounding gridpoints, introducing further numerical uncertainty.

These shortcomings prevent us from plotting magnetic field lines, which is unfortunate as it would have provided a useful tool to visualise the structure of the magnetic field surrounding the neutron star and how this evolves with time. One alternative would be to abandon the goal of plotting an accurate closed loop, and instead place a small sample of arrows depicting the direction of the field at relatively widely spaced intervals. We will not attempt to plot magnetic field lines in our simulations.

However, there are instances in which the magnetic field lines can be exactly computed; see §22.1 for those of a magnetic dipole.

In many simulations, the streamfunction (§9.4) is evolved instead of the fields themselves. Then, the field lines can be plotted immediately simply by producing a contour plot of the streamfunction.

21 Viability of pseudospectral evolution method

The original intention for this project was to model the magnetic and electric fields with a spectral method; that is, to expand into basis functions in the angular directions. In the non-axisymmetric case, these basis functions are the vector spherical harmonics (Chapters 18 and 19). Such methods are common in the literature; however, we will find that they are not well suited to our evolutionary equations.

We have already described how the basis functions can be reduced to Legendre polynomials and $m = 1$ associated Legendre functions if we enforce axisymmetry. Our final code uses only these functions.

Suppose that at time t, the magnetic field $\mathbf{B}(t, \mathbf{r})$ and electric field $\mathbf{E}(t, \mathbf{r})$ have been calculated and both have been expanded into VSH series, so that we have (Definition 19.1)

$$\mathbf{B}(t, \mathbf{r}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \Big[B_m^{r,\ell}(t, r)\, \mathbf{Y}_\ell^m(\theta, \phi) + B_m^{(1),\ell}(t, r)\, \mathbf{\Psi}_\ell^m(\theta, \phi) + B_m^{(2),\ell}(t, r)\, \mathbf{\Phi}_\ell^m(\theta, \phi) \Big], \tag{21.1}$$

$$\mathbf{E}(t, \mathbf{r}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \Big[E_m^{r,\ell}(t, r)\, \mathbf{Y}_\ell^m(\theta, \phi) + E_m^{(1),\ell}(t, r)\, \mathbf{\Psi}_\ell^m(\theta, \phi) + E_m^{(2),\ell}(t, r)\, \mathbf{\Phi}_\ell^m(\theta, \phi) \Big]. \tag{21.2}$$

The time dependence must then be relegated to the VSH series coefficients $B_m^{r,\ell}, B_m^{(1),\ell}, B_m^{(2),\ell}, E_m^{r,\ell}, E_m^{(1),\ell}, E_m^{(2),\ell}$, simply because the basis functions $\mathbf{Y}_\ell^m, \mathbf{\Psi}_\ell^m, \mathbf{\Phi}_\ell^m$ are constant in time. Now suppose that we have also expanded in the radial direction, for example by Chebyshev polynomials, so that[1] (Proposition 14.6)

$$B_m^{r,\ell}(t, r) = \sum_{k=0}^{\infty} B_{m,k}^{r,\ell}(t)\, T_k\big[\Lambda_{r_{\max}, r_{\min}}^{-1}(r)\big], \tag{21.3}$$

and similarly for $B_m^{(1),\ell}, B_m^{(2),\ell}, E_m^{r,\ell}, E_m^{(1),\ell}, E_m^{(2),\ell}$. Then, evolving the fields in time is equivalent to evolving only the VSH series coefficients in time: $B_{m,k}^{r,\ell}(t)$, $B_{m,k}^{(1),\ell}$, $B_{m,k}^{(2),\ell}$, $E_{m,k}^{r,\ell}$, $E_{m,k}^{(1),\ell}$, $E_{m_k}^{(2),\ell}$. Since the VSHs form an orthonormal basis, we can match the expressions for the coefficients term-by-term and index-by-index[2] We have simplified our 6 partial differential equations (Proposition 20.5; one for each vector component) to $(k_{\max}+1)\big[6(\ell_{\max}+1)^2-4\big]$ ordinary differential equations (Proposition 19.7), which are far simpler to compute.

These advantages still apply even if we do not expand in the radial direction, for example if we use a finite differencing scheme instead. Spectral methods do not force us to treat the radial direction in the same way. We will see in Chapter 24 that finite differencing is far more accurate for our model, so we do not lose the option to use it.

Further, we would only need to use the basis a handful of times: once at the start of the code to determine the initial coefficients, and then at timesteps when we wish to output the actual field values (Propositions 14.6 and 19.3). This simplifies even further when the VSH coefficients for the initial configuration can be calculated exactly, e.g. for a dipole (Proposition 22.2).

However, recasting the equations in this way presents some issues, which our evolutionary equations are susceptible to, losing much of the advantage of a full VSH decomposition:

1. Accuracy of the evolution is limited by the accuracy of the calculation of the VSH and Chebyshev coefficients of the initial configuration. Even though a dipole has exactly calculable VSH coefficients, its Chebyshev coefficients are both infinitely many and only calculable numerically (Example 14.7 and its discussion).

[1] We reserve the index n for the time coordinate, so let us use k here.

[2] For example, the evolutionary equations for $B_1^{r,2}$ are given by the $\ell = 2$ and $m = 1$ term on the RHS of the expanded form of Eq. (21.1).

2. We assume that the VSH coefficients of the RHS of our time-evolution equations can be simply read-off at each timestep based on the already-known coefficients of the field components. But even though Definition 19.1 and Proposition 14.6 guarantee that a Chebyshev-VSH expansion of the RHS *always* exists, we know from §19.2 that it may not be attainable without evaluating the basis functions.

Let us expand the time-evolution equation for $\mathbf{B}$, Eq. (20.41), as a VSH series. Applying Definition 19.1 to the LHS, we obtain

$$\frac{\partial \mathbf{B}}{\partial t} = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \left[\frac{\partial B_m^{r,\ell}}{\partial t} \mathbf{Y}_\ell^m + \frac{\partial B_m^{(1),\ell}}{\partial t} \mathbf{\Psi}_\ell^m + \frac{\partial B_m^{(2),\ell}}{\partial t} \mathbf{\Phi}_\ell^m \right], \tag{21.4}$$

Applying Proposition 19.6 to the RHS, we obtain

$$-\boldsymbol{\nabla} \times \mathbf{E} = \sum_{\ell=1}^{\infty} \sum_{m=-\ell}^{\ell} \left[\frac{\ell(\ell+1)}{r} E_m^{(2),\ell} \mathbf{Y}_\ell^m + \left(\frac{\partial E_m^{(2),\ell}}{\partial r} + \frac{1}{r} E_m^{(2),\ell} \right) \mathbf{\Psi}_\ell^m - \left(\frac{\partial E_m^{(1),\ell}}{\partial r} - \frac{1}{r} E_m^{r,\ell} + \frac{1}{r} E_m^{(1),\ell} \right) \mathbf{\Phi}_\ell^m \right]. \tag{21.5}$$

Assuming that we can match the coefficients term-by-term, we would obtain the far simpler set of evolutionary equations

$$\frac{\partial B_m^{r,\ell}}{\partial t} = \frac{\ell(\ell+1)}{r} E_m^{(2),\ell}, \tag{21.6}$$

$$\frac{\partial B_m^{(1),\ell}}{\partial t} = \frac{\partial E_m^{(2),\ell}}{\partial r} + \frac{1}{r} E_m^{(2),\ell}, \tag{21.7}$$

$$\frac{\partial B_m^{(2),\ell}}{\partial t} = -\frac{\partial E_m^{(1),\ell}}{\partial r} + \frac{1}{r} E_m^{r,\ell} - \frac{1}{r} E_m^{(1),\ell}. \tag{21.8}$$

If we expand in the radial direction, these reduce to ordinary differential equations. This appears to match up smoothly, but the issue lies in the equation for $\mathbf{E}$, Eq. (20.42). For example, consider the term

$$\frac{\mathbf{E} \cdot \boldsymbol{\nabla} \times \mathbf{E} - \mathbf{B} \cdot \boldsymbol{\nabla} \times \mathbf{B}}{B^2} \mathbf{B}. \tag{21.9}$$

We obtain $\boldsymbol{\nabla} \times \mathbf{E}$ and $\boldsymbol{\nabla} \times \mathbf{B}$ immediately from Proposition 19.6, and we can even use Proposition 19.10 to efficiently evaluate $\mathbf{E} \cdot \boldsymbol{\nabla} \times \mathbf{E}$ and $\mathbf{B} \cdot \boldsymbol{\nabla} \times \mathbf{B}$. Then, the fraction multiplying $\mathbf{B}$ is straightforward to evaluate. However, Proposition 19.8 tells us that we cannot easily obtain the VSH series coefficients of the vector function formed by multiplying this fraction by $\mathbf{B}$, without performing further integrals. The same will be true for calculating the $\boldsymbol{\nabla} \cdot \mathbf{E}$ term.

As a result, we cannot cast our evolutionary equations (20.41) and (20.42) into simple ODEs for the VSH series coefficients of $\mathbf{B}$ and $\mathbf{E}$. We must evolve the equations in full.

22 Application to magnetic dipole

22.1 Static magnetic dipole

Definition 22.1. *A general axisymmetric divergence-free vector field* $\mathbf{B}(\mathbf{r})$ *can be written as a sum of poloidal and toroidal components as*

$$\mathbf{B} \;=\; \mathbf{B}_\mathrm{p} + \mathbf{B}_\mathrm{t} \;=\; \frac{1}{r\sin(\theta)}\Big(\boldsymbol{\nabla}\Psi \times \mathbf{e}_\phi + A(\Psi)\,\mathbf{e}_\phi\Big), \tag{22.1}$$

where $\Psi(\mathbf{r})$ *is a* **streamfunction** *(§9.4) and* $A(\Psi)$ *is to be determined.*

The streamfunction of a magnetic dipole in spherical coordinates is $\Psi(r,\theta) = \frac{1}{r}\,\sin^2(\theta)$ (Timokhin, 2006), and so the vector field can be written as[1]

$$\mathbf{B} \;=\; \frac{2\cos(\theta)}{r^3}\,\mathbf{e}_r + \frac{\sin(\theta)}{r^3}\,\mathbf{e}_\theta + \frac{A(\Psi)}{r\sin(\theta)}\,\mathbf{e}_\phi. \tag{22.2}$$

It is useful to note that

$$B^2 \;=\; \frac{3\,\cos^2(\theta)+1}{r^6} + \frac{A^2(\Psi)}{r^2\,\sin^2(\theta)}, \tag{22.3}$$

$$\boldsymbol{\nabla}\times\mathbf{B} \;=\; \frac{1}{r\sin(\theta)}\left(\frac{1}{r}\,\frac{\partial A}{\partial\theta}\,\mathbf{e}_r - \frac{\partial A}{\partial r}\,\mathbf{e}_\theta\right). \tag{22.4}$$

Proposition 22.2. *The VSH series of a magnetic dipole is*

$$B^{r,1}(r) \;=\; 4\sqrt{\frac{\pi}{3}}\,\frac{1}{r^3}, \tag{22.5}$$

$$B^{(1),1}(r) \;=\; -\frac{1}{2}\,B^{r,1}(r) \;=\; -2\sqrt{\frac{\pi}{3}}\,\frac{1}{r^3}, \tag{22.6}$$

with $B^{(2),\ell}(r)$ *only calculable once* $A(\Psi)$ *is specified. If* $A(\Psi)=0$*, then* $B^{(2),\ell}(r)=0$*. If* $A(\Psi)=-2\Psi$ *such that* $B_\phi = -2\,\frac{\sin(\theta)}{r^2}$*, then* $B^{(2),\ell}(r) = B^{r,\ell}(r) = 4\sqrt{\frac{\pi}{3}}\,\frac{1}{r^3}\,\delta_{\ell,1}$*.*

Proof. The VSH series for this axisymmetric function is given by the expression in Definition definition: Maths: VSH series of arbitrary vector functions: VSH series expression, with coefficients $B^{r,\ell}(r), B^{(1),\ell}(r), B^{(2),\ell}(r)$ given in Proposition 19.2. For the r-coefficients, we have

$$B^{r,\ell}(r) \;=\; \sqrt{(2\ell+1)\pi}\int_0^\pi \left(\frac{2\cos(\theta)}{r^3}\right) P_\ell^0\,\sin(\theta)\,d\theta \tag{22.7}$$

$$\;=\; \sqrt{(2\ell+1)\,\pi}\,\frac{2}{r^3}\int_0^\pi P_1^0\,P_\ell^0\,\sin(\theta)\,d\theta \tag{22.8}$$

$$\;=\; \sqrt{(2\ell+1)\,\pi}\,\frac{2}{r^3}\,\frac{2}{2\ell+1}\,\delta_{\ell,1} \tag{22.9}$$

$$\;=\; 4\sqrt{\frac{\pi}{2\ell+1}}\,\frac{1}{r^3}\,\delta_{\ell,1} \tag{22.10}$$

$$\;=\; 4\sqrt{\frac{\pi}{3}}\,\frac{1}{r^3}\,\delta_{\ell,1}, \tag{22.11}$$

[1] For reference, in SI units we have $\Psi(r,\theta) = \frac{\mu_0\,m}{4\pi}\,\frac{1}{r}\,\sin^2(\theta)$ and so $\mathbf{B} = \frac{\mu_0\,m}{4\pi}\big(\frac{2\cos(\theta)}{r^3}\,\mathbf{e}_r + \frac{\sin(\theta)}{r^3}\,\mathbf{e}_\theta + \frac{A(\Psi)}{r\sin(\theta)}\,\mathbf{e}_\phi\big)$, where m is the magnetic dipole moment (e.g. Griffiths, 2017, Eq. (5.88)).

where we use that $P_1^0\big[\cos(\theta)\big] = \cos(\theta)$, the orthogonality relation (15.13), and finally evaluate the prefactor of the Kronecker delta symbol for $\ell = 1$ because the expression would clearly by zero for any other ℓ.
For the (1)-coefficients, we have

$$B^{(1),\ell}(r) = \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)} \int_0^\pi \left(\frac{\sin(\theta)}{r^3}\right) P_\ell^1 \, \sin(\theta)\, d\theta \tag{22.12}$$

$$= -\frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)} \frac{1}{r^3} \int_0^\pi P_1^1 \, P_\ell^1 \, \sin(\theta)\, d\theta \tag{22.13}$$

$$= -\frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)} \frac{1}{r^3} \frac{2}{2\ell+1} \frac{(\ell+1)!}{(\ell-1)!} \delta_{\ell,1} \tag{22.14}$$

$$= -2\sqrt{\frac{\pi}{2\ell+1}} \frac{1}{r^3} \delta_{\ell,1} \tag{22.15}$$

$$= -2\sqrt{\frac{\pi}{3}} \frac{1}{r^3} \delta_{\ell,1}, \tag{22.16}$$

where we use that $P_1^1\big[\cos(\theta)\big] = -\sin(\theta)$, the orthogonality relation (16.45), that

$$\frac{1}{\ell(\ell+1)} \frac{(\ell+1)!}{(\ell-1)!} = \frac{1}{\ell(\ell+1)} \frac{(\ell+1)\ell(\ell-1)!}{(\ell-1)!} = 1, \tag{22.17}$$

and evaluate the prefactor of the Kronecker delta symbol for $\ell = 1$.
For the (2)-coefficients, we have

$$B^{(2),\ell}(r) = \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)} \int_0^\pi \left(\frac{A(\Psi)}{r\sin(\theta)}\right) P_\ell^1 \, \sin(\theta)\, d\theta. \tag{22.18}$$

If $A(\Psi) = 0$ then clearly $B^{(2),\ell}(r) = 0$. If $A(\Psi) = -2\Psi$, then

$$B^{(2),\ell}(r)\Big|_{A=-2\Psi} = \frac{\sqrt{(2\ell+1)\pi}}{\ell(\ell+1)} \int_0^\pi \left(-\frac{2}{r^2}\sin(\theta)\right) P_\ell^1 \, \sin(\theta)\, d\theta. \tag{22.19}$$

□

The magnetic field lines of a non-rotating dipole satisfy (e.g. Willis & Young, 1987, Eq. (20))[2]

$$\frac{r}{\sin^2(\theta)} = \text{constant}. \tag{22.20}$$

This is illustrated in Figure 22.1.

[2]We can also see this immediately by recalling from §20.6 that field lines are lines of constant streamfunction, and taking the reciprocal of the streamfunction given by Timokhin (2006), which is a pure number so taking a reciprocal causes no dimensional issues.

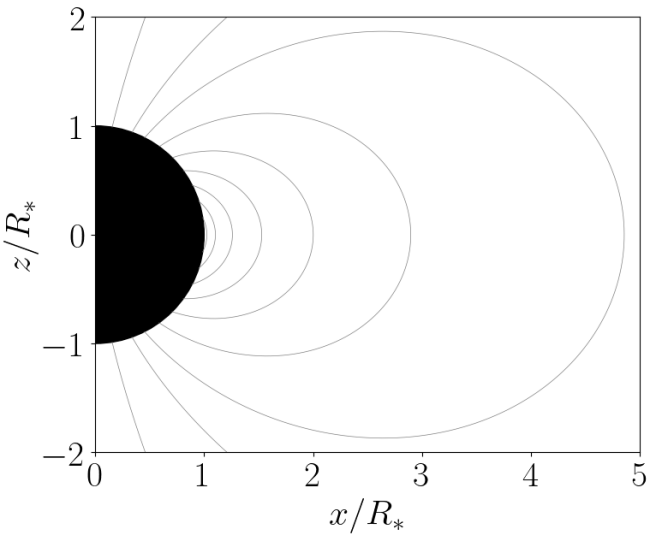

Figure 22.1: Magnetic field lines of a non-rotating magnetic dipole of radius R_*. Produced by codes: `Generate_magnetic_field_values_CSV_dipole.py` and `Magnetic_field_lines_plot_combined_02.py`

22.2 Modelling rotation via an induced electric field

The electric field induced by a magnetic field $\mathbf{B}$ rotating with angular velocity $\boldsymbol{\Omega} = \Omega_r\,\mathbf{e}_r + \Omega_\theta\,\mathbf{e}_\theta + \Omega_\phi\,\mathbf{e}_\phi$ is[3]

$$\mathbf{E}_{\rm rot} \;=\; -\big(\boldsymbol{\Omega}\times\mathbf{r}\big)\times\mathbf{B} \tag{22.21}$$

$$\;=\; r\,\Big(\big[-\Omega_\theta\,B_\theta - \Omega_\phi\,B_\phi\big]\,\mathbf{e}_r + \Omega_\theta\,B_r\,\mathbf{e}_\theta + \Omega_\phi\,B_r\,\mathbf{e}_\phi\Big), \tag{22.22}$$

where $\mathbf{r} = r\,\mathbf{e}_r$ is a position vector.

Proposition 22.3. *The electric field in Eq. (22.21) automatically satisfies the degeneracy condition for FFE,* $\mathbf{E}\cdot\mathbf{B} = 0$.

Proof. This is readily seen by the cyclic permutations of the scalar triple product $(\mathbf{a}\times\mathbf{b})\cdot\mathbf{c} = (\mathbf{b}\times\mathbf{c})\cdot\mathbf{a} = (\mathbf{c}\cdot\mathbf{a})\cdot\mathbf{b}$ and that $\mathbf{a}\times\mathbf{a} = \mathbf{0}$ for any vector $\mathbf{a}$. □

For constant angular velocity Ω about the z-axis, we have

$$\boldsymbol{\Omega} \;=\; \Omega\,\mathbf{e}_z \;=\; \Omega\,\cos(\theta)\,\mathbf{e}_r - \Omega\,\sin(\theta)\,\mathbf{e}_\theta, \tag{22.23}$$

and so

$$\mathbf{E}_{\rm rot} \;=\; \Omega\,r\,\sin(\theta)\,\big[B_\theta\,\mathbf{e}_r - B_r\,\mathbf{e}_\theta\big]. \tag{22.24}$$

For a magnetic dipole, this becomes

$$\mathbf{E}_{\rm rot} \;=\; \frac{\Omega\,\sin(\theta)}{r^2}\,\Big(\sin(\theta)\,\mathbf{e}_r - 2\cos(\theta)\,\mathbf{e}_\theta\Big), \tag{22.25}$$

which is independent of the choice of $A(\Psi)$. We have $\boldsymbol{\nabla}\cdot\mathbf{E}_{\rm rot} = \frac{2\Omega}{r^3}\big[1-3\cos^2(\theta)\big]$ and $\boldsymbol{\nabla}\times\mathbf{E}_{\rm rot} = \mathbf{0}$.

[3] For reference, in SI units we have $\mathbf{E}_{\rm rot} = -\frac{1}{c}\,(\boldsymbol{\Omega}\times\mathbf{r})\times\mathbf{B}$.

Proposition 22.4. *Our magnetic dipole with $A = 0$ with $\Omega < 1$ satisfies the second force-free condition for all points inside the light cylinder and violates it for all points outside the light cylinder. That is, $B^2 - E^2 > 0$ iff $r\,\sin(\theta) < R_{\rm LC}$.*

Proof. We see by Eq. (22.24) that

$$B^2 - E^2 \;=\; \left[1 - \Omega^2\, r^2\, \sin^2(\theta)\right]\left[B_r^2 + B_\theta^2\right] + B_\phi^2. \tag{22.26}$$

For a dipole with $A = 0$, we see from Eq. (22.3) that

$$B_r^2 + B_\theta^2 \;=\; \frac{3\cos^2(\theta) + 1}{r^6} \tag{22.27}$$

and $B_\phi = 0$. Now, if $r \geq 1$ then $\frac{1}{r^6} > 0$, and note also that $\forall\,\theta \in [0, 2\pi]$, then $3\cos^2(\theta) + 1 \in [1, 4]$, and so $B^2 - E^2 < 0$ iff $1 - \Omega^2\, r^2\, \sin^2(\theta) < 0$. Then, using that $\Omega \geq 0$,

$$\left|r\,\sin(\theta)\right| \;<\; \frac{1}{\Omega}. \tag{22.28}$$

But we saw in eq. (20.54) that $\frac{1}{\Omega} = R_{\rm LC}$, completing the proof. □

As a result, we define the **force-free region** to be all points (r, θ) such that $r\,\sin(\theta) < R_{\rm LC}$. Recall from Eq. (8.35) that $r\,\sin(\theta)$ is merely the x-coordinate in our axisymmetric coordinate system, so this is indeed a cylinder with maximum extent in the x-direction but no restriction in the z-direction.

22.3 Energy stored in an electromagnetic field

The **electric energy density** of an electromagnetic field is[4]

$$u_{\rm elec}(t, \mathbf{r}) \;=\; \frac{1}{2}\, E^2(t, \mathbf{r}), \tag{22.29}$$

and so the total electric energy is given by the volume integral over the region $\mathcal{V}$ occupied by the field:

$$U_{\rm elec}(t) \;=\; \int_{\mathcal{V}} u_{\rm elec}(t, \mathbf{r})\, dV. \tag{22.30}$$

In axisymmetric spherical polar coordinates, this is

$$U_{\rm elec}(t) \;=\; 2\pi \int_0^\pi \int_0^\infty u_{\rm elec}(t, \mathbf{r})\, r^2\, \sin(\theta)\, dr\, d\theta \;=\; \pi \int_0^\pi \int_0^\infty E^2(t, \mathbf{r})\, r^2\, \sin(\theta)\, dr\, d\theta. \tag{22.31}$$

Similarly, the **magnetic energy density** of an electric field is[5]

$$u_{\rm mag}(t, \mathbf{r}) \;=\; \frac{1}{2}\, B^2(t, \mathbf{r}), \tag{22.32}$$

and the total magnetic energy is

$$U_{\rm mag}(t) \;=\; \int_{\mathcal{V}} u_{\rm mag}(t, \mathbf{r})\, dV, \tag{22.33}$$

which in axisymmetric spherical polar coordinates is

$$U_{\rm mag}(t) \;=\; \pi \int_0^\pi \int_0^\infty B^2(t, \mathbf{r})\, r^2\, \sin(\theta)\, dr\, d\theta. \tag{22.34}$$

Then, the **total energy density** of the field is given by the sum of its electric and magnetic components, and likewise for the total energy (e.g. Griffiths, 2017, Eq. (9.53)):

$$u(t, \mathbf{r}) \;=\; u_{\rm elec}(t, \mathbf{r}) + u_{\rm mag}(t, \mathbf{r}), \tag{22.35}$$

$$U(t) \;=\; U_{\rm elec}(t) + U_{\rm mag}(t). \tag{22.36}$$

[4] For reference, in SI units this is $u_{\rm elec}(t, \mathbf{r}) \;=\; \frac{1}{2}\,\epsilon_0\, E^2(t, \mathbf{r})$.
[5] For reference, in SI units this is $u_{\rm mag}(t, \mathbf{r}) \;=\; \frac{1}{2}\,\frac{1}{\mu_0}\, B^2(t, \mathbf{r})$.

Proposition 22.5. *The total magnetic energy of a magnetic dipole with $A(\Psi) = 0$ in a finite domain $r \in [r_{\min}, r_{\max}]$ is constant in time, and has the value*

$$U_{\rm mag} \;=\; \frac{4\pi}{3}\bigg(\frac{1}{r_{\rm min}^3} - \frac{1}{r_{\rm max}^3}\bigg), \tag{22.37}$$

Proof. Eq. (22.32) with Eq. (22.2) with $A(\Psi) = 0$ gives

$$u_{\rm mag}(t,\mathbf{r}) \;=\; \frac{1}{2}\bigg[\frac{4\,\cos^2(\theta)}{r^6} + \frac{\sin^2(\theta)}{r^6}\bigg] \;=\; \frac{1}{2}\bigg[\frac{4\,\cos^2(\theta)}{r^6} + \frac{1-\cos^2(\theta)}{r^6}\bigg] \;=\; \frac{1}{2}\,\frac{3\,\cos^2(\theta)+1}{r^6} \tag{22.38}$$

$$=\; \frac{1}{r^6}\Big[P_2\big[\cos(\theta)\big] + P_0\big[\cos(\theta)\big]\Big], \tag{22.39}$$

where we used the Legendre polynomials $P_2\big[\cos(\theta)\big] = \frac{1}{2}\big[3\,\cos^2(\theta) - 1\big]$ and $P_0\big[\cos(\theta)\big] \;=\; 1$. Note that there is no time-dependence. Then, the total magnetic energy over the finite radial domain is

$$U_{\rm mag} \;=\; 2\pi\int_0^\pi\int_{r_{\rm min}}^{r_{\rm max}} \frac{1}{r^6}\Big[P_2\big[\cos(\theta)\big] + P_0\big[\cos(\theta)\big]\Big]\,\sin(\theta)\,r^2\,\sin(\theta)\,dr\,d\theta \tag{22.40}$$

$$=\; 2\pi\bigg(\int_0^\pi P_2\big[\cos(\theta)\big]\,\sin(\theta)\,d\theta + \int_0^\pi P_0\big[\cos(\theta)\big]\,\sin(\theta)\,d\theta\bigg)\bigg(\int_{r_{\rm min}}^{r_{\rm max}}\frac{1}{r^4}\,dr\bigg) \tag{22.41}$$

$$=\; 2\pi\Big(2\,\delta_{2,0} + 2\,\delta_{0,0}\Big)\bigg[-\frac{1}{3}\,\frac{1}{r^3}\bigg]_{r_{\rm min}}^{r_{\rm max}} \tag{22.42}$$

$$=\; 2\pi\Big(0+2\Big)\cdot -\frac{1}{3}\bigg(\frac{1}{r_{\rm max}^3} - \frac{1}{r_{\rm min}^3}\bigg), \tag{22.43}$$

where we used Corollary 15.3. This rearranges to the given result. □

Proposition 22.6. *The total electric energy of a magnetic dipole in a finite domain $r \in [r_{\min}, r_{\max}]$, rotating with constant angular velocity Ω, is constant in time, and has the value*

$$U_{\rm elec} \;=\; \frac{32\pi}{15}\,\Omega^2\,\big(r_{\rm max} - r_{\rm min}\big). \tag{22.44}$$

Proof. Eq. (22.29) with Eq. (22.25) gives

$$u_{\rm elec}(t,\mathbf{r}) \;=\; \frac{1}{2}\,\frac{\Omega^2\sin^2(\theta)}{r^2}\Big[\sin^2(\theta) + 4\cos^2(\theta)\Big] \tag{22.45}$$

$$=\; \frac{\Omega^2\sin^2(\theta)}{2r^2}\Big[1 + 3\cos^2(\theta)\Big] \tag{22.46}$$

$$=\; \frac{\Omega^2}{2r^2}\Big[-3\,\cos^4(\theta) + 2\,\cos^2(\theta) + 1\Big]. \tag{22.47}$$

Use Eq. (15.6) to give

$$-3\,\cos^4(\theta) \;=\; -\frac{24}{35}\,P_4\big[\cos(\theta)\big] - \frac{18}{7}\,\cos^2(\theta) + \frac{9}{35}, \tag{22.48}$$

so that

$$u_{\rm elec}(t,\mathbf{r}) \;=\; \frac{\Omega^2}{2r^2}\Big[-\frac{24}{35}\,P_4\big[\cos(\theta)\big] - \frac{4}{7}\,\cos^2(\theta) + \frac{44}{35}\Big]. \tag{22.49}$$

Use Eq. (15.6) to give

$$-\frac{4}{7}\,\cos^2(\theta) \;=\; -\frac{8}{21}\,P_2\big[\cos(\theta)\big] - \frac{4}{21}\big], \tag{22.50}$$

so that

$$u_{\text{elec}}(t,\mathbf{r}) \;=\; \frac{\Omega^2}{2r^2}\Big[-\frac{24}{35}\,P_4\big[\cos(\theta)\big]-\frac{8}{21}\,P_2\big[\cos(\theta)\big]+\frac{16}{15}\,P_0\big[\cos(\theta)\big]\Big]. \tag{22.51}$$

Now the volume integral is straightforward:

$$\begin{aligned} U_{\text{elec}}(t,\mathbf{r}) \;&=\; 2\pi\int_0^{\pi}\int_{r_{\min}}^{r_{\max}}\frac{\Omega^2}{2r^2}\Big[-\frac{24}{35}\,P_4\big[\cos(\theta)\big]-\frac{8}{21}\,P_2\big[\cos(\theta)\big]+\frac{16}{15}\,P_0\big[\cos(\theta)\big]\Big]\,r^2\,\sin(\theta)\,dr\,d\theta && (22.52)\\ &=\; \pi\Omega^2\bigg(-\frac{24}{35}\int_0^{\pi}P_4\big[\cos(\theta)\big]\,\sin(\theta)\,d\theta-\frac{8}{21}\int_0^{\pi}P_2\big[\cos(\theta)\big]\,\sin(\theta)\,d\theta \\ &\qquad +\frac{16}{15}\int_0^{\pi}P_0\big[\cos(\theta)\,\sin(\theta)\,d\theta\bigg)\int_{r_{\min}}^{r_{\max}}dr && (22.53)\\ &=\; \pi\Omega^2\bigg(0+0+\frac{16}{15}\cdot 2\bigg)\Big[r\Big]_{r_{\min}}^{r_{\max}}, && (22.54)\end{aligned}$$

which simplifies to the given result. □

The total energy of the rotating dipole is then the sum of Eqs. (22.37) and (22.44).

If the dipole is either stationary or rotating at a constant rate, U is constant in time. This provides a useful tool for testing numerical evolutions. Its value may be calculated at each timestep, and an erroneous gain or dissipation of energy can be attributed to inaccuracies in the calculation methods; we touched on this in a footnote in §10.1.

If the dipole is spinning up, U will increase with time, reflecting the fact that the dipole gains energy as its rotation rate increases. Rotational energy goes like $E_{\text{rot}} = \frac{1}{2}\,I\,\Omega^2$, so have the correct Ω dependence in Eq. (22.44).

For $r_{\min} = 1$ and $r_{\max} \to \infty$, we have $U_{\text{mag}} \to \frac{4\pi}{3}$ but $U_{\text{elec}} \to \infty$, which reflects that more energy is required to rotate magnetic field lines that extend further out into space. This corotation breaks at the light cylinder, which should prevent our expression from predicting infinite energies as the domain increases.

23 Coordinate system and function generation

23.1 Time and rotation

Evolution is performed linearly in time, with constant timestep Δt and total number of time values N_t set by the user (see §31.7), running from $t = 0$ to $t = (N_t - 1)\,\Delta t$. These values are enumerated by `T_index`.

We saw in §22.2 that rotation enters the evolution equations via angular velocity $\mathbf{\Omega}$ with magnitude Ω, but observational data usually refers to a rotation period P. Hence, the code allows the user to specify a desired final rotation period P in seconds, which is then converted to a final angular velocity $\tilde{\Omega}$ in code units by Eq. (20.51) with the relevant conversion factor.

Angular velocity is ramped up linearly from zero to this final value. There is an initial period with no rotation, to ensure that the simulation is stable. The user controls the timesteps at which rotation ramp-up begins and ends by the parameters `T_index_rotation_ramp_start` and `T_index_rotation_ramp_stop`. They also have control of the ramp-up rate by varying the difference between these two values.

As we will see in §23.2, it may be useful to set the maximum radial coordinate $r_{\max}$ to twice the final value of the light cylinder radius. As such, it is important to calculate the values of $t, P, \Omega, R_{\rm LC}$ before the radial coordinates are generated. We implement a failsafe to reset back to the chosen $r_{\max}$ in `Initial_Conditions_xx.h` if the evolution will finish before rotation starts, i.e. `n_T` is less than or equal to `T_index_rotation_ramp_start`. Otherwise, the code would return $r_{\max} = 0$ and the evolution would be invalid.

Finally, when using computers to repeatedly increment by the same value x, errors can be introduced due to floating point inaccuracies. Only numbers that can be expressed as a sum of reciprocal powers of 2, e.g. $2^{-9}+2^{-12} = 0.00244140625$, can be represented *exactly* by a floating point; other numbers are only approximated to the nearest sum of reciprocal powers of 2 to the accuracy of the floating point used. For constant numbers that are repeatedly added, which is Δt in this case, the resulting error can become significant. To get around this, we define the function

```
double next_sum_of_reciprocal_powers_of_2( double x )
```

to give the first value above some input `x` that *can* be represented exactly. In this function, we arbitrarily choose a smallest value of $2^{-14} = 6.103515625 \times 10^{-5}$. The function will always return a value between x and $x + 2^{-14}$ inclusive, i.e. greater than or equal to the input. Changing Δt like this and keeping the initial time (zero) and number of time values n_t constant means that the final time $t_{\max}$ will also be changed. The new and old values of Δt and $t_{\max}$ are output to the screen and the log file for reference.

The values of time, angular velocity, rotation period and total rotation angle are pre-calculated at the beginning of the code and stored in arrays by the function

```
void calculate_time_and_rotation_values()
```

Clearly just one value of each is required at a given timestep, but calculating everything in advance allows the values to be tested before committing to an evolution. This may also be useful when determining a ramp-up rate. It also maintains consistency with the treatment of the coordinates r, θ.

We calculate a ramp-up rate in code units per timestep,

```
double dOmega_by_dT_index =
Omega_final /( T_index_rotation_ramp_stop - T_index_rotation_ramp_start + 1.0 );
```

Between `T_index_rotation_ramp_start` and `T_index_rotation_ramp_stop`, inclusive, we calculate

```
Omega[T_index] = Omega[ T_index - 1 ] + dOmega_by_dT_index;
```

with `Omega[T_index] = 0` before `T_index_rotation_ramp_start` and `Omega[T_index] = Omega_final` at and after `T_index_rotation_ramp_stop`. This ramp-up rate is output to the console and the log files for reference, along with its conversion to SI units (rad s^{-2}). Rotation period is recalculated from this, as it may be easier to interpret, and we set `P[T_index] = 0` before rotation begins to avoid `inf` values.

23.1.1 Testing implementation

We write the codes

`Test_time_and_rotation.cpp`
`Test_time_and_rotation.py`

which read the evolution code, run the function `calculate_time_and_rotation_values()`, output the calculated values to a `CSV` file, and plot them in a graph. The results in SI units are shown in Figure 23.1. The chosen parameters are as follows:

```
int    n_T                          = 150;
double delta_T                      = 0.002;
double P_SI_final                   = 2.0;
int    T_index_rotation_ramp_start  = 50;
int    T_index_rotation_ramp_stop   = 100;
```

The value of Δt was updated to 0.00201416015625.

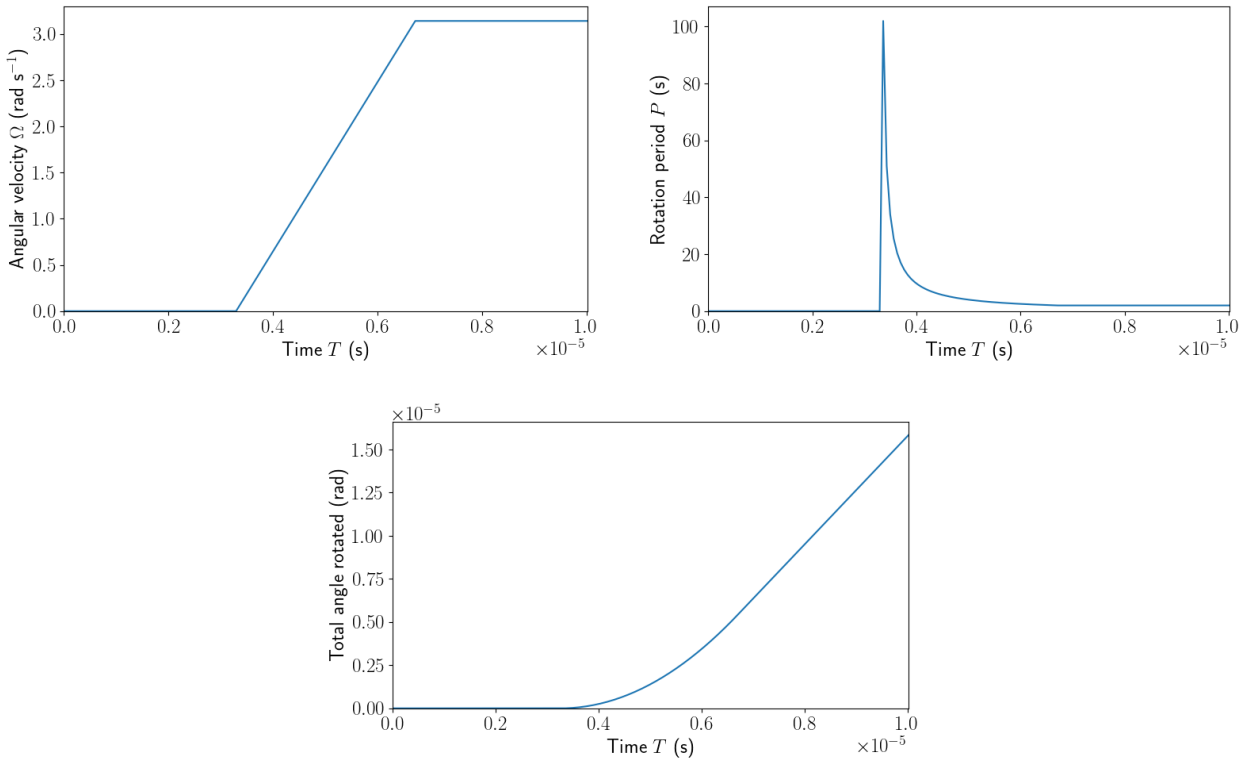

Figure 23.1: Angular velocity, rotation period and total angle rotated by the neutron star as a function of time, given in SI units.

23.2 Coordinate system

The code uses a 2D axisymmetric coordinate system in spherical polar coordinates (r, θ). The radial coordinate $r \in [r_{\rm min}, r_{\rm max}]$ is scaled to the $R_\star$, such that a common choice would be to start at the NS surface $r_{\rm min} = 1$ and end at some cutoff radius $r_{\rm max}$. When rotation is considered, grids should extend beyond the light cylinder; for this reason, we add `bool set_r_max_to_2_R_LC` to `Header_Initial_Conditions_xx.h` to automatically update $r_{\rm max} = 2\,R_{\rm LC}$ as described in §23.1.

The polar coordinate $\theta \in [\frac{\Delta\theta}{2}, \frac{\Delta\theta}{2} + \pi]$ is shifted by half a gridpoint to avoid the north pole $\theta = 0$ and south pole $\theta = \pi$, since there we may encounter numerical errors due to the $\frac{1}{\sin(\theta)}$ term in the divergence and curl, Eqs. (9.3) and (9.4).

The grid is separated into discrete cells with spacing $\Delta r(r)$ and $\Delta\theta(\theta)$. In future code versions, it may prove useful to explore alternative coordinate mappings focussing the gridpoints toward potentially problematic regions, increasing the resolution there at the cost of other regions where high resolution is less critical. However, for simplicity, we use a **linear spacing** such that Δr and $\Delta\theta$ are both constant. The user inputs their choice of $r_{\rm min}$ and $r_{\rm max}$, and decides on the number of gridpoints to include in both dimensions (including the two endpoints), N_r and N_θ. Then, the grid spacing is calculated by

$$\Delta r = \frac{r_{\rm max} - r_{\rm min}}{N_r - 1}, \tag{23.1}$$

$$\Delta\theta = \frac{\pi}{N_\theta - 1}. \tag{23.2}$$

As with the timestep, we then modify Δr to the next decimal that can be represented as a sum of reciprocal powers of 2, which also modifies $r_{\rm max}$. We cannot do this for $\Delta\theta$ because $\theta_{\rm min} = \frac{\Delta\theta}{2}$, $\theta_{\rm max} = \frac{\Delta\theta}{2} + \pi$ and N_θ are fixed.

Enumerating the radial gridpoints by $i \in \mathbb{Z} : 0 \leq i \leq N_r - 1$ and the polar gridpoints by $j \in \mathbb{Z} : 0 \leq j \leq N_\theta - 1$, the coordinates of these gridpoints are then

$$r_i = r_{\rm min} + i\,\Delta r, \tag{23.3}$$

$$\theta_j = \frac{\Delta\theta}{2} + j\,\Delta\theta. \tag{23.4}$$

The polar coordinates r and θ are each stored in 1D lists. Then, to avoid repeat calculation throughout the simulation, the code immediately calculates the values of $\sin(\theta_j)$ and $\cos(\theta_j)$, the values of the Cartesian coordinates

$$x_{i,j} = r_i\,\sin(\theta_j), \tag{23.5}$$

$$z_{i,j} = r_i\,\cos(\theta_j), \tag{23.6}$$

by Eqs. (8.17) and (8.19), and the finite line elements and area elements for 1D and 2D numerical integration in axisymmetric spherical polar coordinates using Eqs. (10.62) and (10.60) respectively, storing these in 1D or 2D arrays as appropriate. Note that the expressions for $x_{i,j}, z_{i,j}, (d\theta)_j, (dA)_{i,j}$ are independent of the coordinate spacing used; they simplify for linear spacing, but there is no performance gain to hard-coding those simplified expressions because the arrays are only populated once.

23.2.1 Testing implementation

We wish to ensure that the lists of coordinate values are produced correctly, including both endpoints, featuring the same number of gridpoints that we specify (i.e. without any "counting the fenceposts" errors) and that the spacing is indeed linear.

We write a code `Test_Coordinates.cpp` which opens the `Header_Time_Evolution.cpp` file and calls the function `void calculate_gridpoints()`. It is important to make sure we test the actual file storing the function used in the evolution code, as opposed to copying that function into a separate testing file. This makes the test easily repeatable following updates to the evolutionary code. Once the gridpoints are calculated, the test code outputs them to a `CSV` file so that they can easily be read by a human or an external program.

We also calculate the first and second derivatives of each coordinate with respect to their indices, $\frac{dr}{di}, \frac{d^2r}{di^2}, \frac{d\theta}{dj}, \frac{d^2\theta}{dj^2}$, numerically at each index i, j. for a linear spacing, we expect that the first derivatives are constant and equal to Δr or $\Delta\theta$, and that the second derivatives are zero.

The chosen parameters are as follows:

```
int    n_r                          = 300;
int    n_t                          = 1000;
double r_min                        = 1.0;
double r_max                        = 10.0;
int    n_T                          = 150;
double delta_T                      = 0.002;
double P_SI_final                   = 2.0;
int    T_index_rotation_ramp_start = 50;
int    T_index_rotation_ramp_stop  = 100;
bool   set_r_max_to_2_R_LC         = true;
```

The value of Δr was updated from 63.827359989792 to 63.827392578125, and correspondingly $r_{\max}$ was updated from 19085.3806369478 to 19085.3903808594. Note that `bool set_r_max_to_2_R_LC` works as expected. The derivatives of the coordinates with respect to the indices are as follows, clearly indicating that the spacing is indeed linear. We also verify from the `CSV` file that the coordinate endpoints and the number of gridpoints are indeed as expected.

```
Mean value of |dr_by_di|  : 63.827392578125
Exp. value of |dr_by_di|  : 63.827392578125
Max  value of |d2r_by_di2|: 0

Mean value of |dt_by_dj|  : 0.00314473739098077
Exp. value of |dt_by_dj|  : 0.00314473739098077
Max  value of |d2t_by_dj2|: 4.44089209850063e-16
```

Were we not to modify Δt or Δr to the nearest power of 2, we would obtain a nonzero second radial derivative:

```
Mean value of |dr_by_di|  : 63.827359989792
Exp. value of |dr_by_di|  : 63.827359989792
Max  value of |d2r_by_di2|: 3.63797880709171e-12
```

Although marginal, we see that this modification is indeed necessary to produce truly linearly spaced coordinates.

23.3 Associated Legendre functions

In order to calculate and evaluate VSH series of axisymmetric vector functions and their spatial derivatives, we require constant evaluation of Legendre polynomials and $m = 1$ associated Legendre functions. To increase code performance, we pre-calculate the values of $P_\ell^0[\cos(\theta)]$ and $P_\ell^1[\cos(\theta)]$ at all gridpoints from $\ell = 0$ up to the cutoff value $\ell_{\max}$ specified by the user. The code populates the 2D arrays `P0[j][ell]`, representing $P_\ell^0[\cos(\theta_j)]$, and `P1[j][ell]`, representing $P_\ell^1[\cos(\theta_j)]$. For each gridpoint θ_j, we perform the following calculations.

1. For $m = 0$ (the Legendre polynomials), hard-code the values for $\ell = 0$ and $\ell = 1$:

$$P_0^0[\cos(\theta_j)] = 1, \tag{23.7}$$
$$P_1^0[\cos(\theta_j)] = \cos(\theta_j). \tag{23.8}$$

   Then, use Bonnet's recursion relation Eq. (15.10) to calculate $P_\ell^0[\cos(\theta_j)]$ for $\ell \in \mathbb{Z} : 2 \leq \ell \leq \ell_{\max}$.

2. For $m = 1$, hard-code the values for $\ell = 1$ and $\ell = 2$:

$$P_1^1[\cos(\theta_j)] = -\sin(\theta_j), \tag{23.9}$$
$$P_2^1[\cos(\theta_j)] = -3\cos(\theta_j)\sin(\theta_j). \tag{23.10}$$

   Note that the values for $\ell = 0$ are already zero upon initialisation of the array `P1[j][ell]`. Then, use the generalised Bonnet recursion relation Eq. (16.14) to calculate $P_\ell^1[\cos(\theta_j)]$ for $\ell \in \mathbb{Z} : 3 \leq \ell \leq \ell_{\max}$.

23.3.1 Testing implementation

To ensure that all the values of $P_\ell^0[\cos(\theta_j)]$ and $P_\ell^1[\cos(\theta_j)]$ are accurate, we write two codes in the subfolder `Test codes`:

1. `Test_Associated_Legendre_Functions_Part1.cpp` exports the calculated values at all gridpoints to a `CSV` file. Similarly to before, we call the same header file and function that produces the function values in the full evolution code, instead of copying the function into the test code. In this case, the function is `void calculate_associated_legendre_functions()`.

2. `Test_Associated_Legendre_Functions_Part2.py` reads this `CSV` file, and then calculates the function values at the same gridpoints via two existing `Python` functions: `scipy.special.eval_legendre()` and `scipy.special.lpmv()`. For each function, e.g. $P_2^1[\cos(\theta)]$, we calculate the standard deviation of the `C++` values with respect to the `Python` values across all gridpoints.

The results are displayed in Table 23.1; we see agreement to 14-15 significant figures across all values of ℓ tested. Our method assumes that the `Python` values are exactly correct, which of course is not true, but in any case such strong agreement with values produced by alternative, publicly available methods, provides reassurance in our calculated values.

Table 23.1: Standard deviation of $m = 0$ and $m = 1$ associated Legendre functions, up to $\ell_{\max} = 10$, calculated by the evolutionary code and compared to values in standard `Python SciPy` modules. Values are copied straight from the `Python` console for easy reproducibility of this table.

ℓ	$m = 0$	$m = 1$
0	0.0	0.0
1	1.8781543466942318e-15	1.734034442963705e-15
2	2.4343783810638837e-15	6.18597381411709e-15
3	3.0426152223290122e-15	1.0965910304560862e-14
4	3.7315273993892195e-15	1.656094595064458e-14
5	4.190064288180544e-15	2.402037537648517e-14
6	4.627552123259113e-15	3.240527627733262e-14
7	5.156903857233102e-15	4.11638385749612e-14
8	5.57471582400508e-15	5.1636150665116126e-14
9	5.938297600437188e-15	6.32402803465098e-14
10	6.394651772972282e-15	7.474016340308689e-14

23.4 Chebyshev polynomials

In order to calculate radial derivatives, we will need to expand functions as Chebyshev series on the interval $[r_{\min}, r_{\max}]$, and so we will need to evaluate the Chebyshev polynomials of the first and second kinds at each gridpoint.

There is a slight complication: we will not be working in terms of the radial coordinate r itself, but the shifted integration variable R (renamed from θ in Eq. (14.45) to avoid confusion with the polar coordinate). Looking at the argument of the function in Eq. (14.45), the function is evaluated at the gridpoints $r = \Lambda_{A,B}\big[\cos(R)\big]$, and so the values of the integration variable should be $R = \cos^{-1}\big[\Lambda^{-1}_{A,B}(r)\big]$. Generating a list of values of R allows us to compute numerical integrals with the trapezium rule as normal.

As with the associated Legendre functions (§23.3), we pre-calculate the values of $T_n(r) = T_n\big[\Lambda_{A,B}\big[\cos(R)\big]\big]$ at all gridpoints from $n = 0$ up to $n_{\max}$, populating the 2D arrays `Tn[i][n]` and `Un[i][n]`. For each gridpoint r_i, we perform the following calculations.

1. To generate the polynomials of the first kind, hard-code the values for $n = 0$ and $n = 1$:

$$T_0\big[\Lambda^{-1}_{A,B}(r)\big] = 1, \tag{23.11}$$
$$T_1\big[\Lambda^{-1}_{A,B}(r_i)\big] = \Lambda^{-1}_{A,B}(r_i). \tag{23.12}$$

   Then, use the recursion relation Eq. (13.15) to calculate $T_n\big[\Lambda^{-1}_{A,B}\big]$ for $n \in \mathbb{Z} : 2 \leq n \leq n_{\max}$.

2. To generate the polynomials of the second kind, hard-code the values for $n = 0$ and $n = 1$:

$$U_0\big[\Lambda^{-1}_{A,B}(r)\big] = 1, \tag{23.13}$$
$$U_1\big[\Lambda^{-1}_{A,B}(r_i)\big] = 2\,\Lambda^{-1}_{A,B}(r_i). \tag{23.14}$$

   Then, use the recursion relation Eq. (13.16) to calculate $U_n\big[\Lambda^{-1}_{A,B}\big]$ for $n \in \mathbb{Z} : 2 \leq n \leq n_{\max}$.

23.5 Determining suitable coordinate spacing and timesteps

There is of course a trade-off between choosing large timesteps and few coordinates for computational speed, and small timesteps and many coordinates for accuracy of calculations. The number of coordinates should be decided first, by testing the accuracy of functions like the VSH decomposition of the vectors. Then, the maximum timestep can be determined as a function of these, by the **Courant-Friedrichs-Lewy (CFL) condition** (Courant et al., 1967).

The CFL condition argues that a signal should not be able to propagate between any two adjacent gridpoints in a time shorter than the timestep; otherwise, energy transfer would be undetectable by the numerical model, putting it at risk of energy dissipation. The timestep might be further restricted by the choice of integration method, but we do not explore this; to be safe, we avoid getting too close to the upper limit on the timestep.

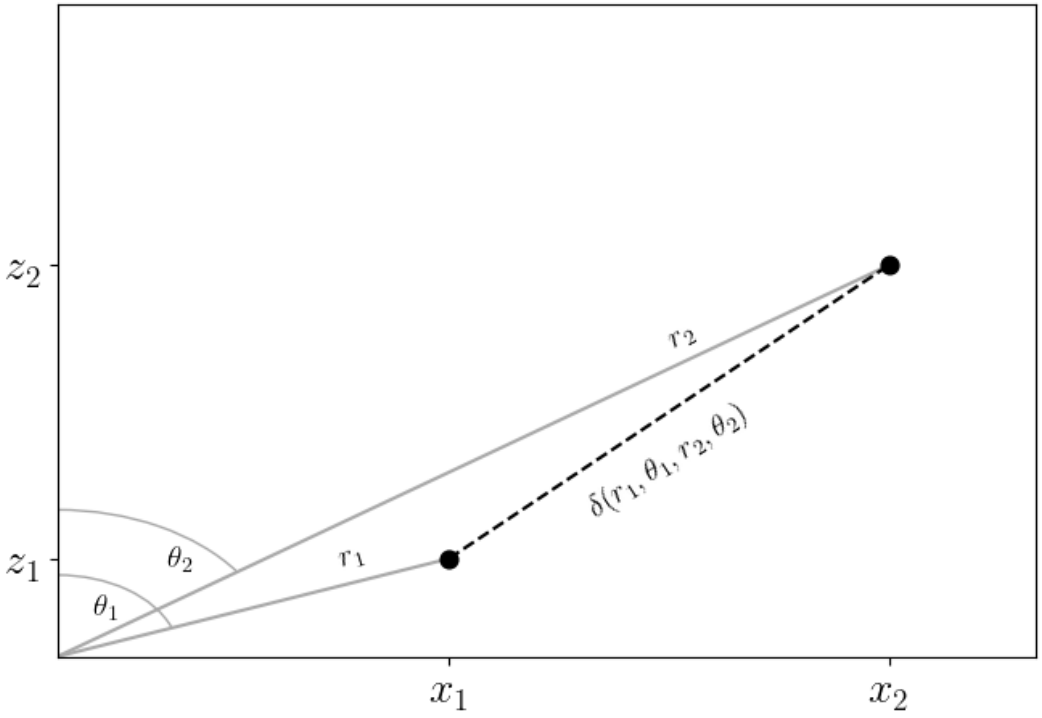

Figure 23.2: Visualisation of the separation $\delta(r_1, \theta_1, r_2, \theta_2)$ between two points $\mathbf{r}_1 = (r_1, \theta_1)$ and $\mathbf{r}_2 = (r_2, \theta_2)$ in 2D polar coordinates.
Produced by code: `Plot_Gridpoint_Separation_Visualisation.py`

Lemma 23.1. *The separation between two gridpoints* $\mathbf{r}_1 = (r_1, \theta_1)$ *and* $\mathbf{r}_2 = (r_2, \theta_2)$ *is*

$$\delta(r_1, \theta_1, r_2, \theta_2) \equiv |\mathbf{r}_2 - \mathbf{r}_1| = \sqrt{r_2^2 + r_1^2 - 2\, r_2\, r_1 \left[\sin(\theta_2)\sin(\theta_1) + \cos(\theta_2)\cos(\theta_1)\right]}. \tag{23.15}$$

This is visualised in Figure 23.2.

Proof. It is easier to calculate the separation between vectors in Cartesian coordinates. By Eqs. (8.35 - 8.37), a gridpoint is

$$\mathbf{r} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} = r\,\sin(\theta)\,\mathbf{i} + r\,\cos(\theta)\,\mathbf{k}, \tag{23.16}$$

and so

$$\begin{aligned}
\delta(r_1,\theta_1,r_2,\theta_2)^2 &= |\mathbf{r}_2-\mathbf{r}_1|^2 && (23.17)\\
&= \Big|\big[r_2\,\sin(\theta_2)-r_1\,\sin(\theta_1)\big]\,\mathbf{i}+\big[r_2\,\cos(\theta_2)-r_1\,\cos(\theta_1)\big]\,\mathbf{k}\Big|^2 && (23.18)\\
&= \big[r_2\,\sin(\theta_2)-r_1\,\sin(\theta_1)\big]^2+\big[r_2\,\cos(\theta_2)-r_1\,\cos(\theta_1)\big] && (23.19)\\
&= \big[r_2^2\,\sin^2(\theta_2)-2\,r_2\,r_1\,\sin(\theta_2)\,\sin(\theta_1)+r_1^2\,\sin^2(\theta_1)\big]\\
&\quad+\big[r_2^2\,\cos^2(\theta_2)-2\,r_2\,r_1\,\cos(\theta_2)\,\cos(\theta_1)+r_1^2\,\cos^2(\theta_1)\big] && (23.20)\\
&= r_2^2\big[\sin^2(\theta_2)+\cos^2(\theta_2)\big]+r_1^2\big[\sin^2(\theta_1)+\cos^2(\theta_1)\big]\\
&\quad-2\,r_2\,r_1\big[\sin(\theta_2)\,\sin(\theta_1)+\cos(\theta_2)\,\cos(\theta_1)\big]. && (23.21)
\end{aligned}$$

Using that $\sin^2(\theta)+\cos^2(\theta)=1$ and taking the square root, we obtain the given result. □

Proposition 23.2 (Separation between neighbouring gridpoints)**.** *For neighbouring gridpoints on the same arc of radius r, with polar angles θ and $\theta+\Delta\theta$, where $\Delta\theta(r,\theta)$ is the spacing in the polar direction, we have*

$$\delta(r,\theta,r,\theta+\Delta\theta) = r\,\sqrt{2}\,\sqrt{1-\cos(\Delta\theta)}. \quad (23.22)$$

For neighbouring gridpoints on the same radial line of polar angle θ, with radial coordinates r and $r+\Delta r$, where $\Delta r(r,\theta)$ is the spacing in the radial direction, we have

$$\delta(r,\theta,r+\Delta r,\theta) = \Delta r(r,\theta). \quad (23.23)$$

Proof. There are three situations in which we can find two neighbouring gridpoints:

1. Constant r, with θ varying by one value of the grid spacing $\Delta\theta(r,\theta)$, which in general is not constant. Then, $r_1=r_2=r$, $\theta_1=\theta$ and $\theta_2=\theta+\Delta\theta(r,\theta)$ and we have

$$\begin{aligned}
\delta(r,\theta,r,\theta+\Delta\theta)^2 &= r^2+r^2-2\,r\,r\big[\sin(\theta+\Delta\theta)\,\sin(\theta)+\cos(\theta+\Delta\theta)\,\cos(\theta)\big] && (23.24)\\
&= 2r^2\big[1-\sin(\theta+\Delta\theta)\,\sin(\theta)-\cos(\theta+\Delta\theta)\,\cos(\theta)\big]. && (23.25)
\end{aligned}$$

   The trig identities

$$\begin{aligned}
\sin(a+b) &= \sin(a)\,\cos(b)+\cos(a)\,\sin(b), && (23.26)\\
\cos(a+b) &= \cos(a)\,\cos(b)-\sin(a)\,\sin(b), && (23.27)
\end{aligned}$$

   yield

$$\begin{aligned}
\delta(r,\theta,r,\theta+\Delta\theta)^2 &= 2r^2\Big(1-\big[\sin(\theta)\,\cos(\Delta\theta)+\cos(\theta)\,\sin(\Delta\theta)\big]\,\sin(\theta)\\
&\qquad-\big[\cos(\theta)\,\cos(\Delta\theta)-\sin(\theta)\,\sin(\Delta\theta)\big]\,\cos(\theta)\Big) && (23.28)\\
&= 2r^2\Big(1-\sin^2(\theta)\,\cos(\Delta\theta)-\cos(\theta)\,\sin(\theta)\,\sin(\Delta\theta)\\
&\qquad-\cos^2(\theta)\,\cos(\Delta\theta)+\sin(\theta)\,\cos(\theta)\,\sin(\Delta\theta)\Big) && (23.29)\\
&= 2r^2\Big(1-\big[\sin^2(\theta)+\cos^2(\theta)\big]\,\cos(\Delta\theta)\Big). && (23.30)
\end{aligned}$$

   Using that $\sin^2(\theta)+\cos^2(\theta)=1$ and taking the square root, we obtain the given result.

2. Constant θ, with r varying by one value of the grid spacing $\Delta r(r,\theta)$, which in general is not constant. Then, $r_1=r$, $r_2=r+\Delta r(r,\theta)$ and $\theta_1=\theta_2=\theta$ and we have

$$\begin{aligned}
\delta(r,\theta,r+\Delta r,\theta)^2 &= (r+\Delta r)^2+r^2-2\,(r+\Delta r)\,r\,\big[\sin(\theta)\,\sin(\theta)+\cos(\theta)\,\cos(\theta)\big] && (23.31)\\
&= (r^2+2\,r\,\Delta r+\Delta r^2)+r^2-2r^2-2\,r\,\Delta r\,\big[1\big] && (23.32)\\
&= \Delta r^2. && (23.33)
\end{aligned}$$

   Taking the square root, we obtain the given result.

3. Both r and θ varying by one gridpoint; that is, gridpoints touching diagonally. We do not consider this situation because our objective is to find the smallest spacing, and diagonally neighbouring gridpoints cannot be closer than those differing by just one coordinate.

□

Corollary 23.3. *If the grid spacing is constant in the radial and polar directions, such that* $\Delta r(r,\theta) = \text{const}$ *and* $\Delta\theta(r,\theta) = \text{const}$*, with* Δr *and* $\Delta\theta$ *not necessarily equal to each other, the smallest possible gridpoint separation is*

$$\delta_{\min} \;=\; \min\left\{\, r_{\min}\sqrt{2}\,\sqrt{1-\cos(\Delta\theta)}\,,\,\Delta r\,\right\}, \tag{23.34}$$

where $r_{\min}$ *is the smallest value of* r *included in the domain. For* $\Delta\theta \ll 1$*, this is well approximated by*

$$\delta_{\min} \;=\; \min\left\{\, r_{\min}\,\Delta\theta\,,\,\Delta r\,\right\}. \tag{23.35}$$

Proof. The expression for $\delta(r,\theta,r,\theta+\Delta\theta)$ is linear in r, so it is minimised when r takes its smallest value. The expression for $\delta(r,\theta,r+\Delta r,\theta)$ is constant. The smallest gridpoint separation is then the smallest of these two minimum values. Now, the Taylor series

$$\cos(\Delta\theta) \;=\; 1 - \frac{1}{2}\,\Delta\theta^2 + \frac{1}{24}\,\Delta\theta^4 + \mathcal{O}(\Delta\theta^6) \tag{23.36}$$

yields

$$1-\cos(\Delta\theta) \;=\; \frac{1}{2}\,\Delta\theta^2 - \frac{1}{24}\,\Delta\theta^4 + \mathcal{O}(\Delta\theta^6) \;=\; \frac{1}{2}\,\Delta\theta^2\left[1 - \frac{1}{12}\,\Delta\theta^2 + \mathcal{O}(\Delta\theta^4)\right], \tag{23.37}$$

and so

$$\sqrt{2}\,\sqrt{1-\cos(\Delta\theta)} \;=\; \Delta\theta\,\sqrt{1 - \frac{1}{12}\,\Delta\theta^2 + \mathcal{O}(\Delta\theta^4)}. \tag{23.38}$$

The binomial approximation $(1+x)^n = 1 + nx + \mathcal{O}(x^2)$ then gives

$$\sqrt{2}\,\sqrt{1-\cos(\Delta\theta)} \;=\; \Delta\theta\left[1 - \frac{1}{24}\,\Delta\theta^2 + \mathcal{O}(\Delta\theta)^4\right] + \mathcal{O}\big[(\Delta\theta^2+\Delta\theta^4)^2\big] \;=\; \Delta\theta + \mathcal{O}(\Delta\theta^3). \tag{23.39}$$

If $\Delta\theta \ll 1$ then $\Delta\theta^3$ is negligibly small compared to $\Delta\theta$. □

To find the smallest neighbouring gridpoint separation in the general case with non-uniform grid spacing, we must evaluate both expressions in Proposition 23.2 at all gridpoints and then determine the smallest of the calculated values. For linear spacing in both directions, an exact expression is given by Corollary 23.3 without the need to calculate any separations. Recall that we neglect other effects on the maximum timestep. If we account for this by avoiding timesteps close to the upper bound, then the approximation Eq. (23.35) for the upper bound will suffice.

Energy transfer in electrodynamics occurs by radiation of light, so signals in our system propagate at the speed of light. We saw in §20.3 that the speed of light in our dimensionless units is unity, so the time to propagate across the smallest grid spacing is simply the value of that separation. We conclude that our maximum timestep in dimensionless units with linear grid spacing in both directions is

$$(\Delta t)_{\max} \;=\; \min\left\{\, r_{\min}\,\Delta\theta\,,\,\Delta r\,\right\}, \tag{23.40}$$

and repeat that values "not too close" to this upper bound should be chosen to account for effects we did not consider here.

23.5.1 Testing implementation

We write the code

`Test codes/Test_CFL_condition.cpp`

to read the evolution code, run the function `calculate_gridpoints()` and determine the spacing between all adjacent pairs of gridpoints. The calculation may be performed by either the Cartesian expression (the square root of the RHS of Eq. (23.19)), the spherical polar expression (Lemma 23.1), the specific spherical polar expressions for adjacent gridpoints (Proposition 23.2) or the approximation of the former (Proposition 23.2), allowing us to verify the accuracy of all four sets of expressions.

The results are shown in Table 23.2. We see agreement between the three exact expressions to 11 significant figures for constant r, and 10 significant figures for constant θ. The approximation for constant r agrees with the exact expressions to 7 significant figures. For linear spacing in both directions, we can use the approximate expressions to good accuracy.

In the evolution code, the CFL condition is implemented by the function

`void calculate_CFL_max_timestep()`

which calculates the value of `double delta_T_CFL` by the approximation Eq. (23.40). One can easily adapt this function for arbitrary step sizes by copying the relevant functions from `Test_CFL_condition.cpp` into `void calculate_CFL_max_timestep()` so that all possible neighbouring gripoint separations are calculated and the function returns the smallest value.

Table 23.2: Smallest separation between adjacent gridpoints, calculated by four different expressions, with $r_{\min} = 1$, $r_{\max} = 10$, $N_r = 300$, $N_\theta = 1000$ and constant grid spacing $\Delta r = 0.0301003344481605$ and $\Delta\theta = 0.00314473739098077$.

Expression	Constant r	Constant θ
Cartesian	0.00314473609516886	0.0301003344481572
Spherical	0.00314473609511657	0.0301003344469711
Spherical (constant)	0.00314473609515188	0.0301003344481605
Spherical (constant) (approximation)	0.00314473739098077	0.0301003344481605

23.6 Ramping up electric field

To ensure a stable evolution, we begin the simulation with a static dipole and let it evolve freely until

```
T_index = T_index_rotation_ramp_start
```

Then, we ramp-up the angular velocity Ω linearly between

```
T_index_rotation_ramp_start < T_index < T_index_rotation_ramp_stop
```

such that Ω reaches its final value Ω_{final} at `T_index_rotation_ramp_stop`, and is maintained until the evolution finishes. As discussed in §22.2, rotation is implemented by adding a term to the electric field, given by Eq. (22.24). To a good approximation, we can split this linearly into additions

$$\mathbf{E}_{\text{rot,step}} = \frac{1}{n_{\text{ramp}}} \Omega_{\text{final}} \sin(\theta) \left[B^\star_\theta \, \mathbf{e}_r - B^\star_r \, \mathbf{e}_\theta \right] \tag{23.41}$$

where $B^\star_r$ and $B^\star_\theta$ are the components of the magnetic field imposed by the star, i.e. a dipole, (borrowing notation from §20.4) and n_{ramp} is the number of timesteps over which the ramping-up occurs, given by

```
T_index_rotation_ramp_stop - T_index_rotation_ramp_start + 1
```

This expression does not account for the fact that $\mathbf{B}$ is time-dependent, but recall the purpose of this ramping-up phase. We know that the final configuration should indeed resemble Eq. (22.24), and we only ramp-up the electric field over a finite time scale to ensure that the code remains stable. Hence, we can neglect any changes this will cause to $\mathbf{B}$ during the ramp-up phase. We have tested using (a) the exact dipole values for B^*_θ and B^*_r, and (b) the values of the magnetic field according to the code at the current timestep. There is no discernible difference between evolutions using either option; we choose the exact dipole value for all simulations. The ramping-up is performed by the function

```
void ramp_up_electric_fields_for_rotation()
```

which is called at every timestep but only updates the electric field if `T_index` is between

```
T_index_rotation_ramp_start
T_index_rotation_ramp_stop
```

We find that re-applying the inner and outer boundary conditions when this code executes, and avoiding gridpoints with `j = 0` and `j = n_r - 1` ($\theta = 0$ and $\theta = \pi$), yield better results.

To check whether this ramping-up gives the final electric field we expect, under the approximation that $\mathbf{B}$ is unaffected, we write the code

```
Test codes/Test_Electric_field_rampup.cpp
```

This simply calls `ramp_up_electric_fields_for_rotation()` for every timestep and outputs the electric field at a specific gridpoint to a `CSV` for quick manual verification. Once the final electric field has been reached, the values are compared to those we would expect if the ramp-up was instantaneous, given by Eq. (22.24). When running this test, we disable the application of boundary conditions and include `j = 0` and `j = n_r - 1`. Using the same parameters as in previous tests, the standard deviation between the resulting and expected final electric field across all gridpoints is as follows:

```
Stdev of final electric field in radial direction: 1.82566e-17
Stdev of final electric field in theta  direction: 2.1078e-17
Stdev of final electric field in phi    direction: 0
```

Note that this test does not model $\mathbf{B}$, so we cannot test our assumption that $\mathbf{B}$ is unaffected during the ramp-up. The only real way to do this is to perform a normal evolution which ends at

```
T_index = T_index_rotation_ramp_stop
```

24 Estimating spatial derivatives of fields

We see from our expression in Proposition 20.5 that, in order to evolve the magnetic and electric fields, we must estimate $\boldsymbol{\nabla} \cdot \mathbf{E}$, $\boldsymbol{\nabla} \times \mathbf{E}$ and $\boldsymbol{\nabla} \times \mathbf{B}$ at each timestep. There are two ways to achieve this: a 2D finite difference method, or expansion of the fields into VSH series to handle the angular component. Here, we shall employ the latter method. From Eqs. (19.56) and (19.57), using VSH series will still require estimation of radial derivatives. This can be achieved in two ways:

1. Calculate numerical derivatives by the finite difference expressions in Chapter 12.
2. Expand each VSH coefficient as a Chebyshev series over the entire radial domain by Proposition 14.6 and then calculate its radial derivative by the relation found in Proposition 14.13.

If the function may be evaluated at any radial coordinate, or if we are free to locate the radial gridpoints at the Chebyshev nodes, then a Chebyshev decomposition is among the most accurate ways to approximate a function. However, in our code, we only have a discrete set of radial gridpoints and they are linearly spaced. Further, the act of decomposing a function into a Chebyshev series is itself an approximation, so we introduce numerical error before even taking a derivative.

Chebyshev decompositions are not always the best choice. Example 14.7 and its discussion showed us that the Chebyshev series coefficients for the radial dependence of a dipole, $\frac{1}{r^3}$, are both infinitely many and only calculable numerically. Further, they decay relatively slowly with n. Then, the accuracy of the decomposition then depends strongly on our choice of truncation index $n_{\max}$, but including too many unnecessary terms gives more opportunities for numerical error to disturb the fit.

Recall also from Proposition 14.13 that the first derivative of the nth term in a Chebyshev depends on an infinite number of Chebyshev polynomials. Although a function with a finite Chebyshev series avoids this issue, that does not match our situation. Thus, our choice of $n_{\max}$ affects the radial derivatives over and above its effect on the decomposition itself.

Finite differencing is far less computationally intensive to implement. It could be argued that any deficiency in its accuracy compared to a Chebyshev series is negated by the ability to use finer grids for the same processing power. However, that assumes that the deficiency can be cured by scaling alone, which is not always true, and in our application the grid spacing is limited by the spacing of the simulation itself.

However, one drawback of finite differencing is that we cannot use the more accurate symmetric derivative at the endpoints. This potentially introduces a source of errors at $r_{\min}$ and $r_{\max}$, whereas there are no obviously troublesome gridpoints for a Chebyshev series.

24.1 Testing radial derivatives with finite differencing

We described in Chapter 12 how to calculate the first and second derivatives of a function $f(x)$ evaluated at a set of equally spaced gridpoints. We produced expressions of varying accuracy, and these have been encoded in the header file `Finite_Difference_Expressions.h`. In this section, we will compare finite difference expressions between themselves in order to find the best candidate for this method. This will be added to the main evolution code within the functions

```
std::vector<double> radial_derivatives_1_FD( std::vector<double> &f )
std::vector<double> radial_derivatives_2_FD( std::vector<double> &f )
```

which take a vector `v` of function values at the gridpoints and return a vector of derivative values at the same gridpoints. Note that arrays are passed by reference (hence the use of `&`); this is because passing large arrays by value requires many values to be read by the function, which can easily take more time than the execution of the function itself.

Testing of radial finite difference methods is performed by the code

`Test codes/Test_radial_derivatives_FD.cpp`

The user defines a function `f` of the radius `r` along with exact expressions for its first two radial derivatives. The desired accuracy $\mathcal{O}(h^N)$ is specified and the code is ran, calculating the absolute relative error

$$\left| 1 - \frac{\left(\frac{df}{dr}\right)_{\text{FD}}}{\left(\frac{df}{dr}\right)_{\text{exact}}} \right| \tag{24.1}$$

at each gridpoint, along with the standard deviation between $\left(\frac{df}{dr}\right)_{\text{FD}}$ and $\left(\frac{df}{dr}\right)_{\text{exact}}$ across all gridpoints. This is performed for the first and second radial derivatives.

Since it is a realistic function matching the radial dependence of a magnetic dipole, we test $f(r) = \frac{1}{r^3}$ such that $\frac{df}{dr} = \frac{-3}{r^4}$ and $\frac{d^2f}{dr^2} = \frac{12}{r^5}$. The parameters are set as follows:

```
int    n_r                    = 300;
int    n_t                    = 1000;
int    n_T                    = 150;
double delta_T                = 0.002;
double P_SI_final             = 0.001;
double r_min                  = 1.0;
double r_max                  = 10.0;
bool   set_r_max_to_2_R_LC    = true;
```

The resulting standard deviation for various $\mathcal{O}(h^n)$ is given in Table 2, and the absolute relative error as a function of r is plotted in Figure 24.1. As expected, error decreases as the chosen order increases. For the second derivative at $\mathcal{O}(h^6)$, we approach floating point precision and low levels of noise enter the data; this still yields better accuracy than lower-order methods, so is not a concern.

Higher-order expressions require more datapoints and hence more function evaluations. In general, this comes at a cost of increasing the computation power required. However, since all of our function values are precalculated in this example and will be numerical data in the evolution code, there is no associated penalty. We can thus choose a method with high accuracy, without drawbacks.

Table 24.1: Standard deviation of first and second radial derivatives of $f(r) = \frac{1}{r^3}$ to various accuracy $\mathcal{O}(h^N)$, with relevant parameters defined in the text.

N	First derivative	Second derivative
2	0.0011419	0.012834
4	1.4424e-05	0.00045611
6	4.1378e-07	2.0894e-05

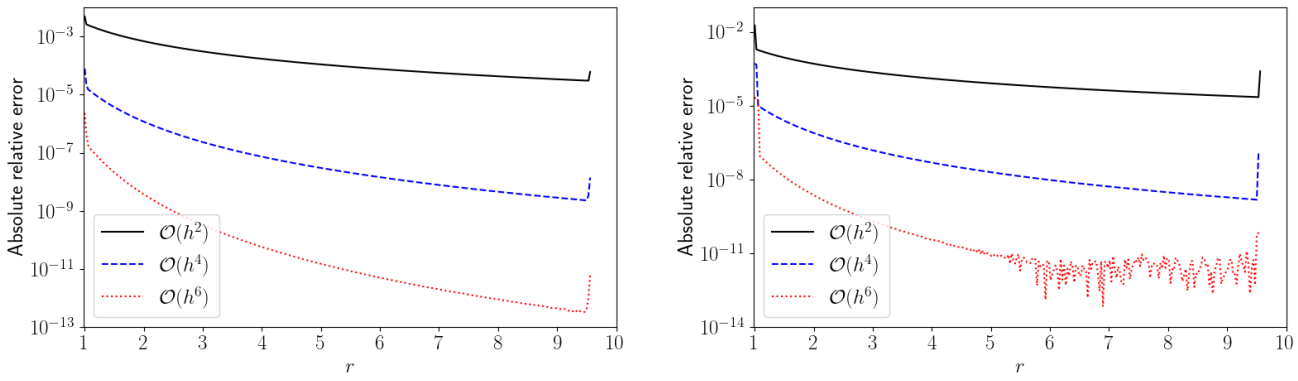

Figure 24.1: Absolute relative error of first derivative (*Left*) and second derivative (*Right*) of a function with respect to the radial coordinate using a three-point finite difference method.

Although we have expressions for the endpoints that in theory yield similar accuracy as for the intermediate points, there is still a jump in error at these values. To mitigate this, we can produce a "hybrid" system of finite-difference expressions that use a higher-order expression than required near the endpoints. We try a hybrid system with $\mathcal{O}(h^8)$ near the endpoints (the first and last four gridpoints) and $\mathcal{O}(h^6)$ between. The results near the endpoints are given in Figure 24.2. We see that a hybrid system improves both derivatives near both endpoints, by around an order of magnitude, and brings the error roughly in line with the last gridpoint at which a symmetric $\mathcal{O}(h^6)$ expression can be use, which is also plotted for comparison.

We choose a hybrid $\mathcal{O}(h^6 - h^8)$ system as our preferred finite difference method for the remainder of the project, and so we implement this function within

```
std::vector<double> radial_derivatives_1_FD( std::vector<double> &f )
std::vector<double> radial_derivatives_2_FD( std::vector<double> &f )
```

in the main evolution code.

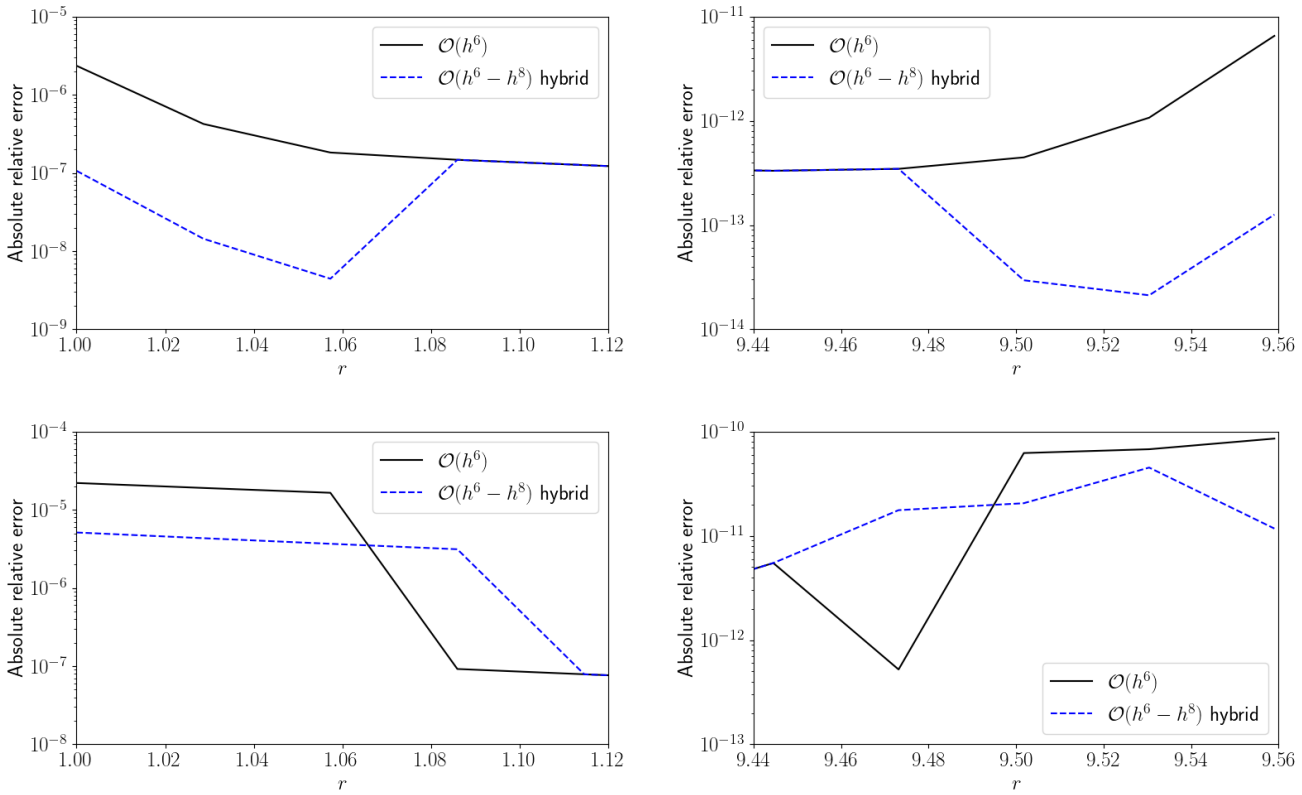

Figure 24.2: Absolute relative error of first derivative (*Top*) and second derivative (*Bottom*), for the innermost (*Left*) and outermost (*Right*) few gridpoints, for a finite-difference system using $\mathcal{O}(h^6)$ at all gridpoints and a system using $\mathcal{O}(h^8)$ near the endpoints.

24.2 Testing radial derivatives from Chebyshev series

One method of calculating the radial derivative of a function is to calculate its Chebyshev series by Eq. (14.45) and differentiate it term-by-term. In the evolution code, the Chebyshev series coefficients are calculated by the function

```
std::vector<double> chebyshev_series_coeffs( std::vector<double> &v )
```

which performs a 1D numerical integral by the trapezium rule in Cartesian coordinates, Eq. (10.73). Here, we suppose that f is only known at N uniformly spaced gridpoints r_i, since that is the situation we will encounter in the evolution code. This precludes integration methods allowing function evaluations off the gridpoints such as the RK4 method (§10.1) and the ability to set the gridpoints equal to the Chebyshev nodes; we expect accuracy to be suboptimal as a result. The function and its first derivative are then evaluated by the two functions

```
std::vector<double> evaluate_chebyshev_series( std::vector<double> &coeffs )

std::vector<double> evaluate_chebyshev_series_dr( std::vector<double> &coeffs )
```

To test the accuracy of these implementations, we write the code

```
Test codes/Test_Chebyshev_series_arbitrary_interval.cpp
```

A realistic test function might be $f(r) = \frac{1}{r^3}$, but its resulting Chebyshev series cannot be calculated analytically, as can be appreciated from Example 14.7. Instead, we choose the simpler function $f(r) = r^2$, whose Chebyshev series coefficients were calculated exactly in Example 14.12. Setting the parameters the same as for the finite-differencing test, we obtain the results shown in Table 24.2.

We see reasonable agreement for small n, but the relative error grows with n and the calculated values are substantially nonzero for $n > 2$. This may have to do with the wide range of radial gridpoints, or equivalently the relatively small gridspacing compared to $r_{\max}$, but since these radial coordinates represent typical values that we wish to use, we do not have the freedom to try smaller ranges. The issue of potentially overfitting, where we use more Chebyshev polynomials than needed and hence become susceptible to numerical error when the coefficients are not accurate, is made more substantial by the inaccuracy of the calculated coefficients and the fact that the relative error grows with n. In this simple example, it is obvious that we should truncate at $n = 2$, but in real situations that may not be the case. Indeed, functions like $1/r^3$ may require infinitely many coefficients.

The standard deviation of the recalculated function values from the Chebyshev series, over all gridpoints, is respectively

```
Stdev f(x) : 0.509373824373812
Stdev df/dx: 4.04846327106684
```

for $n_{\max} = 10$ and

```
Stdev f(x) : 0.0282734491338191
Stdev df/dx: 0.0224819520646707
```

for $n_{\max} = 2$.

Table 24.2: Calculated and exact Chebyshev coefficents of $f(r) = r^2$ on an arbitrary interval, with relevant parameters defined in the text.

n	Calculated	Exact	Relative error
0	37.0268617116986	37.0303361504339	-9.38356256690032e-05
1	45.170295936518	45.1874412018806	-0.000379569471642416
2	9.11624740009207	9.15710505144671	-0.0044818497745267
3	-0.0800977853421981	0	1
4	-0.132725455852873	0	1
5	-0.191619470608629	0	1
6	-0.265226716254139	0	1
7	-0.333894796127272	0	1
8	-0.419877997140564	0	1
9	-0.488688971820303	0	1
10	-0.577974768981311	0	1

Now that the accuracy of the Chebyshev series coefficients themselves has been established, we can perform the decomposition for $f(r) = \frac{1}{r^3}$. Using the same parameters, we repeat the test for various values of $n_{\max}$ and find a best result at $n_{\max} = 10$ with standard deviations

```
Stdev f(x) : 0.00481196799223615
Stdev df/dx: 0.0662394945416891
```

The first derivative is an order of magnitude less accurate than for finite differencing $\mathcal{O}(h^2)$. We attribute this to the Chebyshev decomposition itself already being relatively poor, for the reasons outlined above brought on by the restrictions of our coordinate system. The absolute relative error as a function of r is plotted for a few $n_{\max}$ in Figure 24.3. Note that the error grows with r. The approximations at the endpoints are not

particularly worse than the intermediate points, but the approximation at $r_{\max}$ is poor as a result of the error growing with r.

At this point, we can already eliminate Chebyshev series as a candidate for handling divergence-free functions, making finite differencing the method to use if maintaining zero divergence is less important than an accurate decomposition. This avoids the need to calculate expressions for the second radial derivative, which we already saw in Eq. (13.45) for $r \in [-1, 1]$ to be a complicated expression.

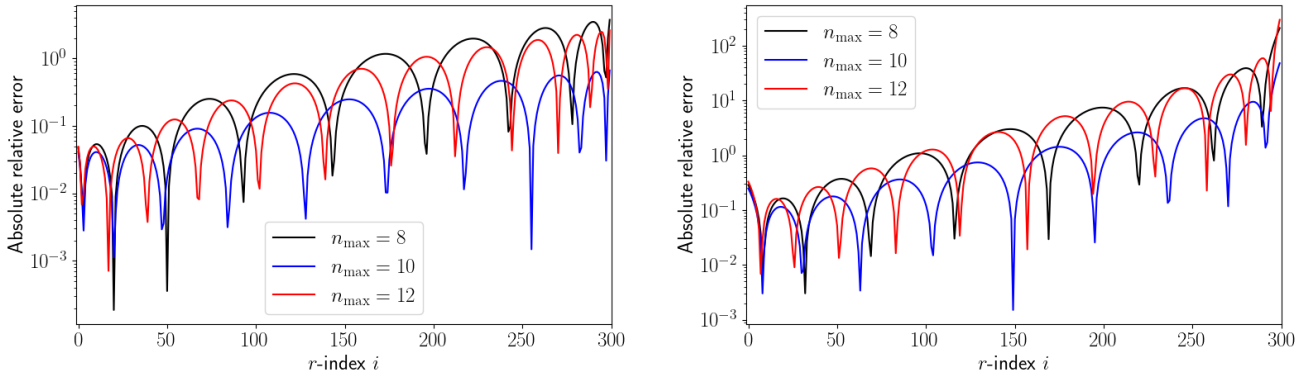

Figure 24.3: Absolute relative error of reconstructed function (*Left*) and its first derivative (*Right*) with respect to the radial coordinate using a Chebyshev series approximation of varying maximum order.

Note that the "sweet spot" for a choice of truncation index depends also on the number of gridpoints used. Table 24.3 shows the standard standard deviation of the VSH series of $f(r) = \frac{1}{r^3}$ for various $n_{\max}$, both for $N_r = 300$ and $N_r = 3000$ gridpoints. In both cases, a "sweet spot" appears in the balance between adding more Chebyshev polynomials and introducing more opportunities for numerical error; this occurs at $n_{\max} = 10$ and $n_{\max} = 14$, respectively.

Table 24.3: Standard deviation of the Chebyshev series coefficients of $f(r) = \frac{1}{r^3}$, first with 300 radial gridpoints and then with 3000.

$n_{\max}$	$N_r = 300$	$N_r = 3000$
2	0.112838781973848	0.112189670740875
4	0.0526601223826733	0.053110542010322
6	0.0208758873336772	0.0219069611833538
8	0.00736559271247053	0.00825758354638785
10	0.00481196799223615	0.00287120944065367
12	0.00646971912189649	0.000904685992760827
14	0.00808585023097846	0.000430350336366974
16	0.00931073258260299	0.000567577051818395
18	0.0101361607314389	0.000743958703430782
20	0.0105493573166358	0.000913512435034331

24.3 Testing VSH series

With a radial derivative method chosen, we next verify our calculation method for axisymmetric VSH series. The VSH series for axisymmetric fields are calculated with separate functions for each component:

```
std::vector< std::vector<double> > VSH_decomposition_r
( std::vector< std::vector< std::vector<double> > > &v )

std::vector< std::vector<double> > VSH_decomposition_1
( std::vector< std::vector< std::vector<double> > > &v ){

std::vector< std::vector<double> > VSH_decomposition_2
( std::vector< std::vector< std::vector<double> > > &v ){
```

If a vector $\mathbf{B}$ is divergenceless, we have two options to calculate its $B^{(1),\ell}$ coefficients:

1. Use the same expression as for the non-divergenceless case.
2. Use Corollary 19.5 and calculate $\frac{dB^{r,\ell}}{dr}$ by finite differencing.

In the latter case, we replace `VSH_decomposition_1` by

```
for( int ell=0; ell<=ell_max; ell++ ){
    B_VSH_coeffs_dr[0][ell] = radial_derivatives_1_FD( B_VSH_coeffs[0][ell] );
}
```

which calculates $\frac{dB^{r,\ell}}{dr}$ by finite differencing, followed by

```
void calculate_B_1_ell_and_derivative_divergenceless()
```

which calculates $B^{(1),\ell}$ and $\frac{dB^{(1),\ell}}{dr}$ based on these values.

The advantage of using separate functions for each component is that `VSH_decomposition_r` and `VSH_decomposition_2` can be called regardless of the method used; it is only `VSH_decomposition_1` that changes. This avoids copying code and introducing errors or code divergence as functions are updated.

To test both of the above methods, we use the code

```
Test codes/Test_VSH_series_axisymmetric.cpp
```

For our test vector, we use the static magnetic dipole Eq (22.2) with $A(\Psi) = -2\,\Psi$. It is of course divergenceless and its exact VSH series coefficients are known (Proposition 22.2). We use separate functions for the vector values to those in `Initial_Conditions_xx.h`,

```
double B_r_function_test( double r, double t )
double B_t_function_test( double r, double t )
double B_p_function_test( double r, double t )
```

alongside functions for the known exact values of the VSH coefficients,

```
double B_VSH_coefficient_exact_r_test( double r, int ell )
double B_VSH_coefficient_exact_1_test( double r, int ell )
double B_VSH_coefficient_exact_2_test( double r, int ell )
```

This makes it easier to change the vector being tested and keep expressions for its exact values and those of the known VSH coefficients in the same file, without fear of desynchronisation if `Initial_Conditions_xx.h` is updated for future code runs.

These functions are used to populate the pre-existing vector in the main evolution code,

```
std::vector< std::vector< std::vector<double> > > B
( 3, std::vector< std::vector<double> > ( n_points_r, std::vector<double> ( n_points_t ) ) )
```

which stores the magnetic field **B** evaluated at all the gridpoints. the first index of size 3 represents the three components B_r, B_θ, B_ϕ, so that, for example, `B[0][3][5]` represents $B_r(r_3, \theta_5)$.

We use the same parameters as given in §24.1, and maximum associated Legendre function order $\ell_{\max} = 10$. The standard deviations of the recalculated vector components $A_r(r, \theta)$ and $A_\phi(r, \theta)$ across all coordinates are:

```
stdev of r-component across all coordinates: 9.10713e-06
stdev of p-component across all coordinates: 2.32977e-06
```

Figure 24.4 plots the standard deviation of each vector component recalculated from its VSH series, as a function of each coordinate. For example, the value as a function of r is calculated by holding r constant and calculating the standard deviation as a function of θ across these gridpoints.

Both the recalculated B_r (hence $B^{r,\ell}$) and B_ϕ (hence $B^{(2),\ell}$) agree with the original vector within 10^{-4} to 10^{-8} at all gridpoints. The agreement for B_r is poorest at smaller radii and near the poles, exhibiting fairly strong radial and angular dependence. The agreement for B_ϕ is independent of radius, which is to be expected since $B_\phi = -2\sin(\theta)$ is independent of r. Although the exact behaviour is unique to our choice of function, we can be confident by the magnitude of the standard deviation that the VSH decomposition is accurate in the r- and ϕ-directions.

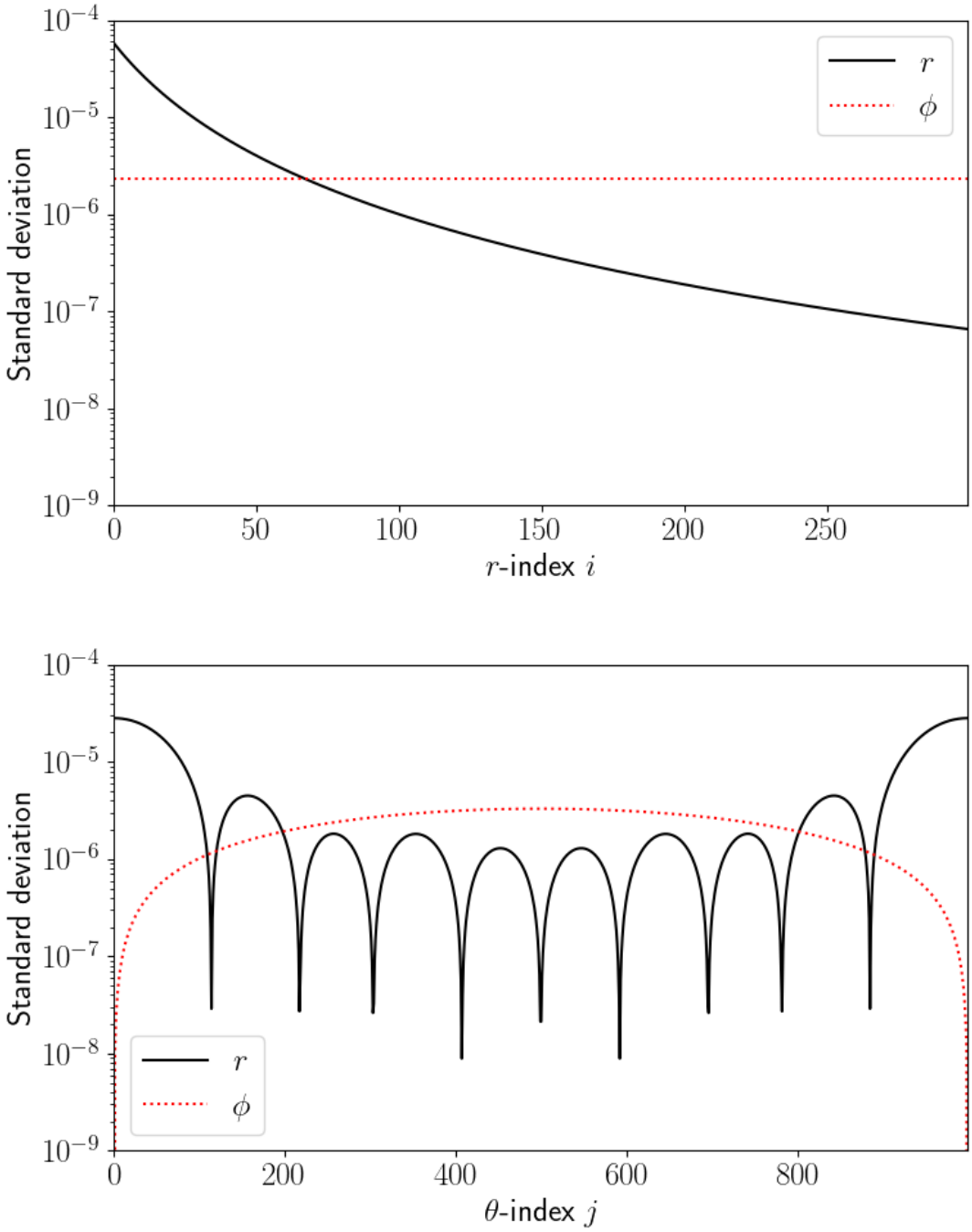

Figure 24.4: Standard deviation of spherical polar components of a vector with nonzero divergence, recalculated from its VSH series. *Top:* As a function of the radial coordinate (i.e. the standard deviation is taken across all θ coordinates for a given r). *Bottom:* As a function of the polar coordinate θ.

For the θ-component, we compare the results from the two methods described above. The standard deviation across all coordinates is:

```
Original            : 1.84197e-07
Finite differencing : 1.05419e-06
```

Figure 24.5 plots the standard deviation of the recalculated B_θ as a function of r and as a function of θ. The finite differencing method is far less accurate than the original method, which is to be expected since the calculation of numerical derivatives necessarily introduces an extra source of uncertainty. Curiously, the error is greatest at the two radial endpoints

When deciding which method to use for a divergenceless vector, we must consider whether guaranteeing that its VSH series remains divergenceless is more important than modelling its θ-component (in axisymmetry) to high precision. A nonzero divergence may grow over time and create significantly more issues than a method which is simply less accurate. The issue can be resolved by some divergence-cleaning algorithm, but our numerical model does not yet feature one.

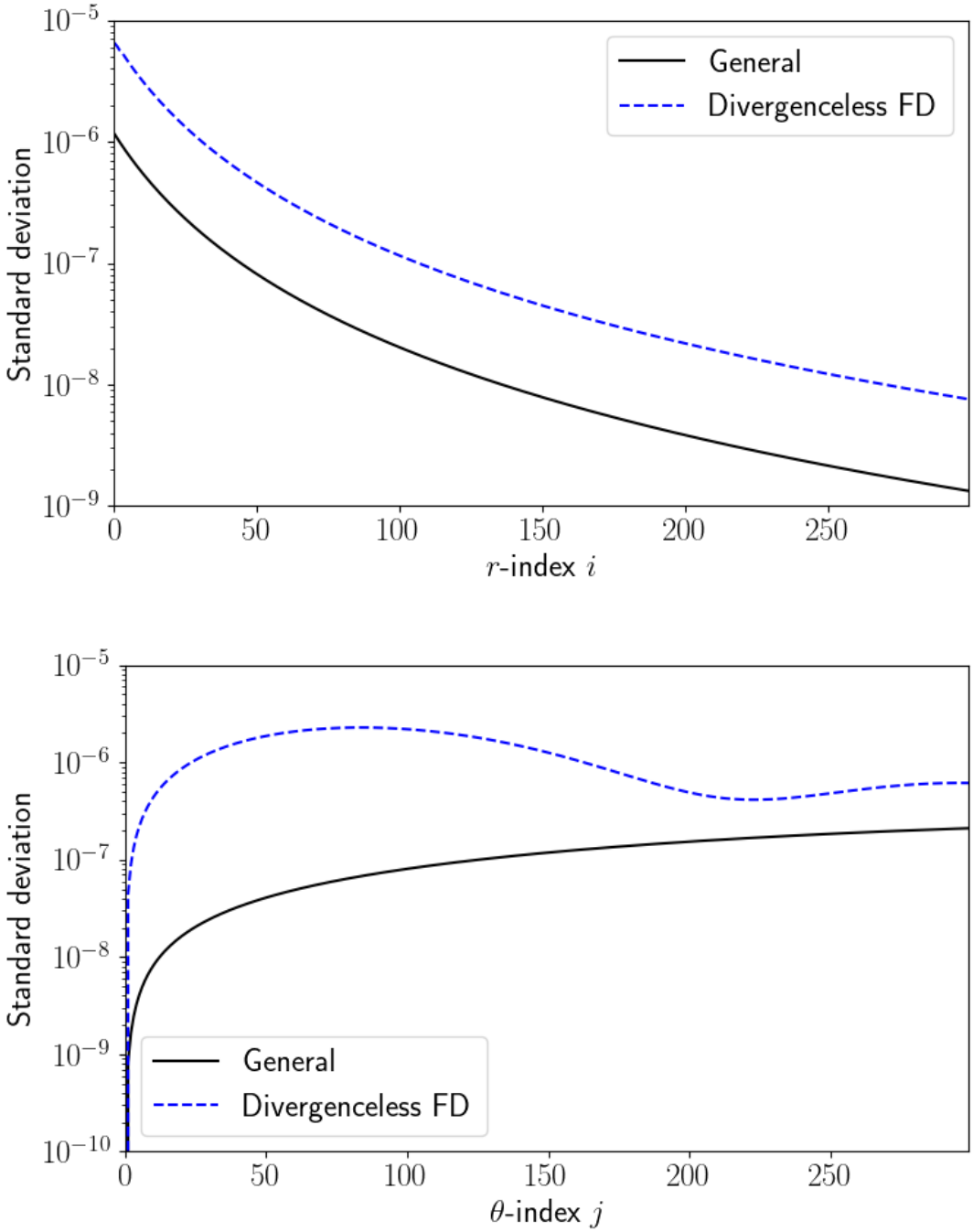

Figure 24.5: Standard deviation of θ-component of a vector with zero divergence, recalculated from its VSH series, where the VSH series was taken by three different methods as described in the text. *Top:* As a function of the radial coordinate (i.e. the standard deviation is taken across all θ coordinates for a given r). *Bottom:* As a function of the polar coordinate θ.

Perhaps surprisingly, it may be better *not* to use a specific expression for $A^{(1),\ell}$ if the vector is divergenceless. However, this introduces the risk of the recalculated vector having nonzero divergence. To that end, we now calculate the resulting divergence of the vector using the original VSH decomposition method. Again, this can be performed by either taking finite differences or by calculating the Chebyshev series of the coefficients. Note that this already introduces a numerical error which may even dwarf the error in the original decomposition, so we must be careful when drawing any conclusions.

In any case, we still require accurate calculation of numerical radial derivatives for the divergence and curl, so the second two methods mentioned above are still needed elsewhere in the code. Of the two, it appears that a finite-differencing method is the more accurate.

Recall from Corollary 19.5 that the $B^{(1),\ell}$ coefficients already contain radial derivatives, and so $\frac{dB^{(1),\ell}}{dr}$ contains second radial derivatives. This causes numerical uncertainties to be far higher than if only first derivatives were required.

The accuracy of a VSH decomposition depends on the truncation index $\ell_{\max}$, but from Proposition 22.2 we know that the coefficients for a static dipole are zero for $\ell > 1$. We can only test for an optimum $\ell_{\max}$ when we come to full evolutions of the rotating model. One simple way to tell is by how close the calculated coefficients for the chosen $\ell_{\max}$ to zero; if they are significantly nonzero, then increasing $\ell_{\max}$ is likely to have some effect, but further investigation will be required to determine whether this is noise or useful data. As an initial estimate, we find that $\ell_{\max} = 10$ gives reasonable accuracy.

24.4 Testing divergence and curl

With the accuracy of the VSH series itself established, let us now calculate $\boldsymbol{\nabla}\cdot\mathbf{B}$ and $\boldsymbol{\nabla}\times\mathbf{B}$ for a test vector $\mathbf{B}$. To this end, we write the test code

`Test codes/Test_divergence_and_curl_from_VSH_series_axisymmetric.cpp`

which allows us to specify a vector $\mathbf{B}$ along with exact expressions for its divergence and curl. We then choose which of the three methods outlined at the top of the chapter that we wish to use. However, we have already discounted Chebyshev series as an option due to their poor accuracy; all that remains is a comparison between the finite-difference expressions for the general case and the divergenceless case.

24.4.1 Divergenceless function

First, let us define a vector whose divergence is zero but whose curl is nonzero, allowing us to use both expressions to test the accuracy of a calculation of the curl. We use a magnetic dipole with $A(r,\theta) = r^3\,\sin^2(\theta)$, such that $B_\phi(r,\theta) = r^2\,\sin(\theta)$ and

$$\boldsymbol{\nabla}\times\mathbf{B} \;=\; 2\,r\,\cos(\theta)\,\mathbf{e}_r - 3\,r\,\sin(\theta)\,\mathbf{e}_\theta. \tag{24.2}$$

The standard deviation in $\boldsymbol{\nabla}\cdot\mathbf{B}$ and $\boldsymbol{\nabla}\times\mathbf{B}$ using both methods is given in Table 24.4. Recall from Proposition 19.6 that $B^{(1),\ell}$ only appears in the ϕ-component of $\boldsymbol{\nabla}\times\mathbf{B}$, so we should expect the same results for the r- and θ-components.

Both methods give around the same accuracy in $\big[\boldsymbol{\nabla}\times\mathbf{B}\big]_\phi$. The divergenceless method agrees with $\boldsymbol{\nabla}\cdot\mathbf{B} = 0$ to within floating-point accuracy, as expected; the general method yields a standard deviation in $\boldsymbol{\nabla}\cdot\mathbf{B}$ of around 10^{-4}.

Table 24.4: Standard deviation of numerically calculated divergence and curl of a divergenceless function described in the text, using the VSH series with and without the specific expression for divergenceless functions.

Function	General	Divergenceless
$\boldsymbol{\nabla}\cdot\mathbf{B}$	0.0001806640383908	4.39443763815612e-13
$\big[\boldsymbol{\nabla}\times\mathbf{B}\big]_r$	1.36013607233612e-05	1.36013607233612e-05
$\big[\boldsymbol{\nabla}\times\mathbf{B}\big]_\theta$	2.0382825340586e-05	2.0382825340586e-05
$\big[\boldsymbol{\nabla}\times\mathbf{B}\big]_\phi$	3.74476451401511e-05	3.63320888329485e-05

24.4.2 Function with nonzero divergence

It remains to test the accuracy of the expression for $\boldsymbol{\nabla}\cdot\mathbf{B}$ itself, which only applies to the general method above. We use the function

$$\mathbf{B}(r,\theta) \;=\; \frac{\cos(\theta)}{r^3}\,\mathbf{e}_r + \frac{2\,\sin(\theta)}{r^3}\,\mathbf{e}_\theta + \frac{\sin(2\theta)}{r^3}\,\mathbf{e}_\phi, \tag{24.3}$$

which has a similar form to a dipole and hence is reasonably realistic, but its divergence is nonzero:

$$\boldsymbol{\nabla}\cdot\mathbf{B} \;=\; \frac{3\,\cos(\theta)}{r^4}. \tag{24.4}$$

We obtain a standard deviation in $\boldsymbol{\nabla}\cdot\mathbf{B}$ of 9.52748878910725e-05, which is around the same accuracy as the value of $\boldsymbol{\nabla}\times\mathbf{B}$.

Finally, note that the divergenceless method will always yield $\boldsymbol{\nabla}\cdot\mathbf{B} = 0$ to within floating point accuracy, even if the original function is divergenceless. This follows because we do not calculate $B^{(1),\ell}(r)$ in its own right, but only as a derivative of $B^{r,\ell}(r)$.

24.5 Monitoring divergence under time evolution

In our simulation, we will begin with a divergenceless function (the magnetic field $\mathbf{B}$ of a dipole) and allow the values of this function to change over time as the system evolves. Even if the function should remain divergenceless at all times, inaccuracies in the method may introduce a spurious nonzero value. Since $\boldsymbol{\nabla} \cdot \mathbf{B}$ depends on the coordinates, we will in principle need to monitor its value at each gridpoint over time. Let us instead develop a method of averaging the values over the entire domain, producing a single value that can be plotted as a function of time. It will also be useful if this value is a dimensionless number, so that it is unaffected by our choice of non-dimensionalisation made in §20.3.

Because the divergence is a first spatial derivative, we have $[\boldsymbol{\nabla} \cdot \mathbf{B}] = [B]\,[L]^{-1}$, where the square brackets mean "dimensions of" and in particular $[L]^{-1}$ means "dimensions of length". We divide by the magnitude of the magnetic field and multiply by the radial coordinate at the same gridpoint at which $\boldsymbol{\nabla} \cdot \mathbf{B}$ is being evaluated, to give the dimensionless quantity $\frac{r\,\boldsymbol{\nabla}\cdot\mathbf{B}}{B}$.

Now, to account for all gridpoints, let us perform a volume integral of this quantity over the entire domain. This will introduce a dimensional factor $[L]^3$, so let us return to a pure number by dividing by the volume of the domain V. Our quantity is then

$$\frac{1}{V}\int \frac{r\,\boldsymbol{\nabla}\cdot\mathbf{B}}{B}\,dV. \tag{24.5}$$

For a spherical domain stretching from radial coordinate $r_{\min}$ to $r_{\max}$,

$$V \;=\; \frac{4\pi}{3}\left(r_{\max}^3 - r_{\min}^3\right). \tag{24.6}$$

Since the functional form of the magnetic field will no longer be known once rotation starts, we will of course have to calculate $\boldsymbol{\nabla}\cdot\mathbf{B}$ numerically at each gridpoint (discussed above), and we will have to calculate the volume integral numerically over our finite region (discussed in §10.4 and §10.5). Naturally then, we do not expect the result to be accurate, but what is more important is that it remains constant in time. The differentiation and integration methods, and the coordinate themselves, do not change with time, so we can say with certainty that any observed time-variation in our calculated quantity is due to $\boldsymbol{\nabla} \cdot \mathbf{B}$ changing with time (at some or all gridpoints), hence a sign that spurious nonzero divergence has been introduced by our evolutionary scheme.

This in itself cannot tell us *where* in the domain the nonzero divergence was introduced, but we will regularly send values of $\boldsymbol{\nabla} \cdot \mathbf{B}$ across the domain to the output CSV files, which can be analysed separately if the need arises.

24.6 Conclusion

We have shown that Chebyshev decompositions are not accurate enough to be used in our code, despite the potential advantages they typically represent.

We have settled on a finite-difference scheme working to $\mathcal{O}(h^8)$ for gridpoints within four steps of the inner and outer boundaries, and to $\mathcal{O}(h^6)$ for all intermediate points. The latter method uses a symmetrical expression, while the former methods are offset. Using this hybrid system of finite difference expressions ensures a roughly even numerical error across the domain.

We have shown that VSH decompositions for grid spacings typically used in our code yield standard deviations in the curl and divergence of a vector of around 10^{-5} in code units.

Our fears at the beginning of the chapter about using second derivatives for the divergenceless VSH decomposition have been eased. Since we have the capability to calculate first and second derivatives to any accuracy required, we are able to get around this issue and use the divergenceless expression without fear of introducing numerical error. This conclusion does not contradict the observation in §24.3 that the numerical error from a divergenceless expression can be larger than the for general case; in that section, we were not considering any other variables. Here, the numerical error in the θ-direction dominates over that in the r-direction, so effectively the two radial finite differencing methods have the same accuracy. The knowledge that errors are now dominated by the θ-direction validates our choice of maximum accuracy of radial derivatives in §24.1.

Were we to persist with the general expression, we see from the analysis in this chapter that the introduced divergence would be around 10^{-4}. This is reasonably low, but an order of magnitude higher than the error in the curl, so may well become the dominant source of error when compounded over multiple timesteps.

We have developed a simple method of monitoring how $\boldsymbol{\nabla} \cdot \mathbf{B}$ evolves over time, which may be used as an early warning sign of inaccuracies in our numerical calculation of spatial derivatives.

We conclude that finite-difference VSH decompositions should be used at all times, and when we expect the vector to be divergenceless, we should use the divergenceless-specific decomposition.

25 Implementing magnetospheric twists

In this chapter, we briefly describe how a magnetospheric twist might be modelled within the code. Twists are implemented by altering the rotation rate Ω within a finite region of the magnetosphere $r \in (r_{\mathrm{L}}, r_{\mathrm{U}})$ and $\theta \in (\theta_{\mathrm{L}}, \theta_{\mathrm{U}})$, where the subscripts L and U refer respectively to lower and upper limits. The twist can take arbitrary values at any gridpoint within this region, allowing for complex angular dependencies to be trialled in order to better approximate physical models of a twist.

As with rotation, the twist is implemented linearly over a finite number of timesteps in order to simulate a gradual onset, and better preserve smooth evolution of the magnetosphere. We recommend to only begin the ramp-up of the twist after full rotation has been reached, but the code is built so that the rotation and twist ramp-up intervals are independent of each other. If the star is slowly rotating, it may be preferable to approximate it as non-rotating; the independence of the ramp intervals easily allows this.

We are especially interested in twists that break the north-south symmetry of the magnetosphere, so recommend that twists are only located within the northern hemisphere to minimise any risk of "cancelling out" between points above and below the equator. This is achieved simply by maintaining $\theta_{\mathrm{U}} \leq \frac{\pi}{2}$.

Let us note that the importance of smoothing the twist grows with resolution of the code: if we have few gridpoints, the action of a smoothing function across any particular gridpoint will be diluted across its extent. We will see in Chapter 26 that numerical instabilities limit the resolution at which our code runs effectively, so perhaps the implementation of a smoothing region may not be so crucial in our situation.

25.1 Smoothing the twist over the coordinates in 1D

Simply imposing a twist in the defined region by adding it to the base rotation rate may cause issues with jump discontinuities in Ω near the boundaries of the twisted region. To counter this, let us develop a few rudimentary smoothing functions which provide a more gradual onset as one moves from outside to inside the twisted region in any direction.

In the following examples, suppose that we initially have $\Omega = 0$ everywhere, and wish to impose a twist that will yield $\Omega = A = \text{const}$ for some finite range of x-values. If there were no smoothing method, the graph of $f(x)$ would appear as a step function. We will describe functions which smooth the transition from $\Omega = 0$ to $\Omega = A$ and the transition back to $\Omega = 0$ by applying the change in Ω gradually across a finite region of x either side of the twisted region. Our functions may be easily extended to situations where Ω is an arbitrary function of x both in the twisted and non-twisted regions.

25.1.1 Linear ramp in 1D

Perhaps the simplest smoothing function is a linear increase from $x = 0$ at the boundary to $x = A$ at some chosen point within the region, with a similar linear decrease to zero at the opposite boundary.

Proposition 25.1. *Let $A \in \mathbb{R}$ and let $x_{\rm L}, x_{A-}, x_{A+}, x_{\rm U} \in \mathbb{R}$ such that $x_{\rm L} < x_{A-} < x_{A+} < x_{\rm U}$. Then,*

$$f(x) \;=\; \begin{cases} \dfrac{A}{x_{A-} - x_{\rm L}}\left(x - x_{\rm L}\right) & x_{\rm L} \le x \le x_{A-}, \\ A & x_{A-} \le x \le x_{A+}, \\ \dfrac{A}{x_{A_+} - x_{\rm U}}\left(x - x_{\rm U}\right) & x_{A+} \le x \le x_{\rm U}, \\ 0 & \text{otherwise}, \end{cases} \tag{25.1}$$

represents a function which rises linearly from 0 at $x = x_{\rm L}$ to A at $x = x_{A-}$, maintains this value until $x = x_{A+}$, and falls linearly to 0 at $x = x_{\rm U}$.

Proof. The straight line between two pairs of gridpoints (x_1, y_1) and (x_2, y_2) has equation

$$y - y_1 \;=\; m(x - x_1), \tag{25.2}$$

or equivalently $y - y_2 = m(x - x_2)$, where

$$m \;=\; \frac{y_2 - y_1}{x_2 - x_1} \tag{25.3}$$

is the gradient of the line.
Let us first fit a straight line between the points $(x_{\rm L}, 0)$ and (x_{A-}, A). It is simplest if we use the first point, not the second, to find the equation of the line:

$$y - 0 \;=\; m(x - x_{\rm L}), \tag{25.4}$$
$$\Rightarrow\; y \;=\; m(x - x_{\rm L}). \tag{25.5}$$

The gradient is

$$m \;=\; \frac{A - 0}{x_{A-} - x_{\rm L}} \;=\; \frac{A}{x_{A-} - x_{\rm L}}. \tag{25.6}$$

These combine to give the expression for the region $x \in [x_{\rm L}, x_{A-}]$.
Second, we fit a straight line between (x_{A+}, A) and $(x_{\rm U}, 0)$. It is easier to use the second point in the equation for the line:

$$y - 0 \;=\; m(x - x_{\rm U}), \tag{25.7}$$
$$\Rightarrow\; y \;=\; m(x - x_{\rm U}). \tag{25.8}$$

The gradient is

$$m \;=\; \frac{0 - A}{x_{\rm U} - x_{A+}} \;=\; \frac{A}{x_{\rm U} - x_{A+}}. \tag{25.9}$$

These combine the give the expression for the region $x \in [x_{A+}, x_{\rm U}]$. Finally, the function is simply constant for $x \le x_{\rm L}$, for $x \in [x_{A-}, x_{A+}]$ and for $x \ge x_{\rm U}$. Combining the function behaviour for all of these subintervals, and noting that the expressions indeed match $f(x) = 0$ at the transition points between intervals x_{A-} and x_{A+}, we obtain the given result. □

While such a function is simple to understand and implement, we see that

$$\frac{df}{dx} = \begin{cases} \dfrac{A}{x_{A-} - x_{\rm L}}\, x & x_{\rm L} \le x \le x_{A-}, \\ 0 & x_{A-} \le x \le x_{A+}, \\ \dfrac{A}{x_{A_+} - x_{\rm U}}\, x & x_{A+} \le x \le x_{\rm U}, \\ 0 & \text{otherwise}, \end{cases} \tag{25.10}$$

and so we will encounter four discontinuities in the derivative at $x \in \{x_{\rm L}, x_{A-}, x_{A+}, x_{\rm U}\}$. This is easily seen from the graph of the function in Figure 25.1.

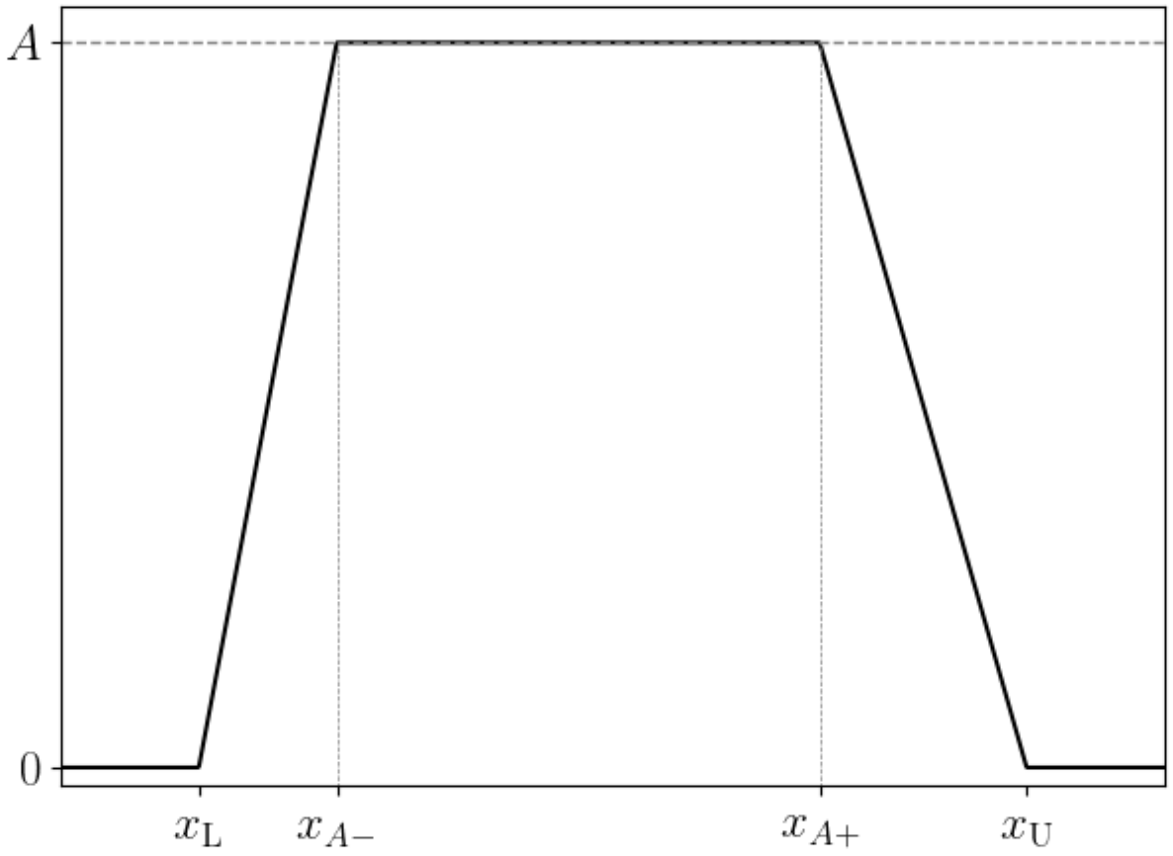

Figure 25.1: Example of a linear smoothing function over one dimension, generated by Eq. (25.1). The ramp-up and ramp-down rates are exaggeratedly slowed and deliberately chosen to be different, for illustrative purposes. Produced by code: `Plot_Smooth_increase_and_decrease_1D_linear.py`

25.1.2 Exponential decay in 1D

We can partially counter the discontinuities in the derivatives above by using an exponential decay to smooth the end of the ramp-up before the function reaches A, and the start of the ramp-down once the function begins to decrease.

Proposition 25.2. *Let $A \in \mathbb{R}$, let $p, q \in (0,1)$ and let $x_{\mathrm{L}}, x_p, x_q, x_{\mathrm{U}} \in \mathbb{R}$ such that $x_{\mathrm{L}} < x_p < x_q < x_{\mathrm{U}}$. Then,*

$$f(x) \;=\; \begin{cases} A\left[1 - e^{-B(x-x_{\mathrm{L}})}\right]\left[1 - e^{-C(-x+x_{\mathrm{U}})}\right], & x_{\mathrm{L}} \le x \le x_{\mathrm{U}}, \\ 0 & \text{otherwise}, \end{cases} \tag{25.11}$$

where

$$B \;=\; -\frac{1}{x_p - x_{\mathrm{L}}}\ln(1-p), \tag{25.12}$$

$$C \;=\; -\frac{1}{-x_q + x_{\mathrm{U}}}\ln(1-q), \tag{25.13}$$

represents a function which rises from 0, tails off arbitrarily close to A (from below), maintains values close to A, before falling back to 0, all between lower and upper limits x_{L} and x_{U}, with ramp-up and ramp-down controlled by $f(x_p) = pA$ and $f(x_q) = qA$.

Proof. We know that the function

$$f(x) \;=\; A\left[1 - e^{-Bx}\right] \tag{25.14}$$

represents a smooth increase from 0 to A for $x \ge 0$, with ramp length controlled by B: larger B causes a more sudden ramp. To avoid the function falling to $-\infty$ for $x \ll 0$, we can use cases to set $f(x) = 0$ for $x < 0$. To begin the ramp at a particular lower limit x_{L}, we simply use the rule for shifting graphs to the right and replace x by $x - x_{\mathrm{L}}$. This yields

$$f(x) \;=\; \begin{cases} A\left[1 - e^{-B(x-x_{\mathrm{L}})}\right] & x \ge x_{\mathrm{L}}, \\ 0 & \text{otherwise}. \end{cases} \tag{25.15}$$

To control the value of B, and hence the sharpness of the ramp-up, suppose that we wish for the function to rise to some fraction[1] $p \in (0,1)$ of its maximum value A at the value $x = x_p > x_{\mathrm{L}}$, where p, x_0 are both free parameters. Then, we have

$$f(x_p) \;=\; pA \;=\; A\left[1 - e^{-B(x_p - x_{\mathrm{L}})}\right], \tag{25.16}$$

which solves for B to give the result in Eq. (25.12). Now, functions are reflected in the y-axis by replacing $f(x)$ with $f(-x)$. Then,

$$g(x) \;=\; \begin{cases} A\left[1 - e^{-C(-x+x_{\mathrm{U}})}\right] & x \le x_{\mathrm{U}}, \\ 0 & \text{otherwise}, \end{cases} \tag{25.17}$$

represents a smooth decrease from A to 0 at upper limit $x = x_{\mathrm{U}}$. To control the tailing-off length, suppose that we wish the function to fall to some fraction $q \in (0,1)$ of its maximum value A at $x = x_q < x_{\mathrm{U}}$, so that $g(x_q) = qA$. This yields the value of C in Eq. (25.13). Finally, we can multiply Eqs. (25.15) and (25.17) with only a single occurrence of A to obtain a function which smoothly ramps from 0 at $x = x_{\mathrm{L}}$ toward A, and then smoothly decreases to zero at $x = x_{\mathrm{U}}$. The ramp-up speed is controlled by B and the ramp-down speed by C. □

[1] E.g. $p = 0.99$ means the function will rise to 99% of its maximum value.

This function is plotted in Figure 25.2 with its significant points highlighted, and in Figure 25.3 with various ramping rates. Higher B and C yield faster, less smooth ramps: since $B \sim \frac{1}{x_p}$, increasing B causes x_p to decrease and so close values to peak are reached sooner, and the same is true for C. In particular, as $B, C \to \infty$, the exponentials tend to zero and we recover the non-smoothed function $f(x) = A$ for $x_1 \leq x \leq x_2$.

We have $f(x_\mathrm{L}) = f(x_\mathrm{U}) = 0$, so the field values will be smooth across the boundary if they are roughly zero outside the twisted region. If there is some residual field value outside the region, this may be accounted by by simply adding this residual value to all points within the twisted region. However, there is a potential discontinuity in the derivative at the boundary since we might expect the fields to be roughly constant (zero derivative) in a small region surrounding the edge of the boundary, but there is a constant ramp within the twisted region. This is a potential limitation of our developed smoothing function. The potential discontinuity in the derivative can be appreciated from Figures 25.2 and 25.3.

A further limitation is that, although the function reaches 0 at the endpoints, it never actually reaches $f(x) = A$ because the function is asymptotic with the line $y = A$. We must be satisfied to get "arbitrarily close" to A within the twisted region; we do not attempt to quantify how close the function gets.

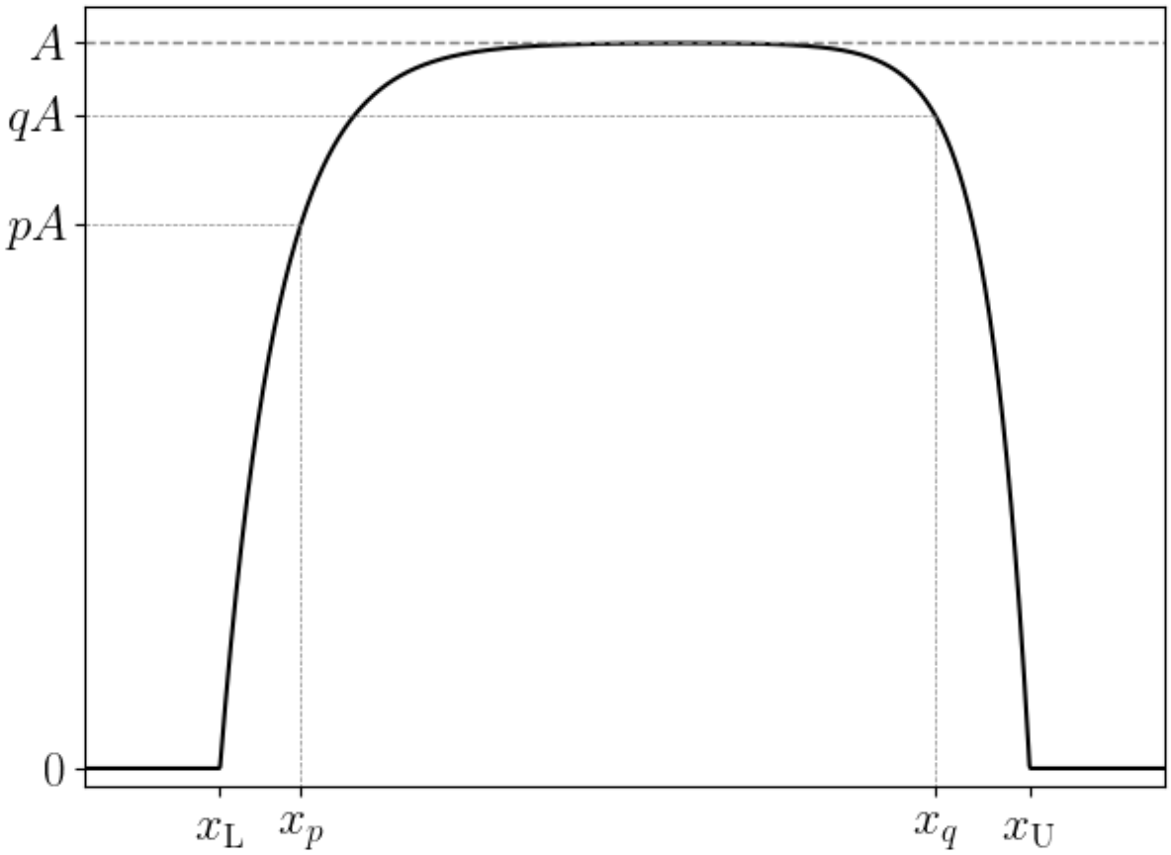

Figure 25.2: Example of an exponential smoothing function over one dimension, generated by Eq. (25.11). The ramp-up and ramp-down rates are exaggeratedly slowed and deliberately chosen to be different, for illustrative purposes.
Produced by code: `Plot_Smooth_increase_and_decrease_1D_exponential_single.py`

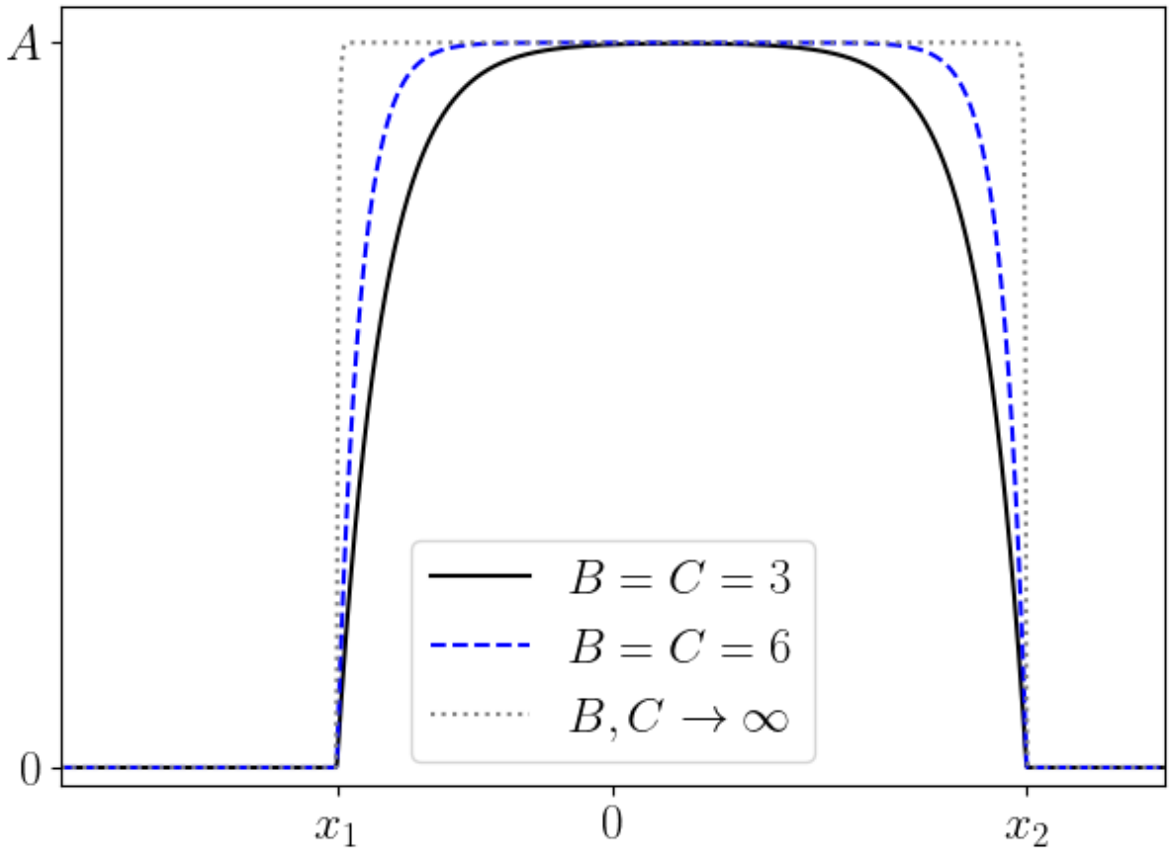

Figure 25.3: Example of an exponential smoothing function over one dimension, generated by Eq. (25.11). various values of the parameters B and C are plotted, to demonstrate their effect on the ramping rate. Produced by code: `Plot_Smooth_increase_and_decrease_exponential_1D_multiple.py`

25.1.3 Hyperbolic tangent in 1D

Let us try to address the limitations of the exponential smoothing function above by constructing an alternative using hyperbolic tangents, which offer both a smooth onset and a smooth cessation of the ramp.

Proposition 25.3. *Let $A \in \mathbb{R} : A > 0$, let $p, q \in (0,1)$ and let $x_{\mathrm{L}}, x_{p-}, x_{p+}, x_{q-}, x_{q+}, x_{\mathrm{U}} \in \mathbb{R}$ such that $x_{\mathrm{L}} < x_{p-} < x_{p+} < x_{q-} < x_{q+} < x_{\mathrm{U}}$. Then, a smooth ramp-up from $x \approx 0$ to some value $x \approx A$ and back to $x \approx 0$, roughly within the interval (x_{p-}, x_{q+}), is given by*

$$f(x) = \begin{cases} \frac{A}{4}\Big[\tanh\big[B(x-x_p)\big]+1\Big]\Big[\tanh\big[C(-x+x_q)\big]+1\Big] & x_{\mathrm{L}} \le x \le x_{\mathrm{U}}, \\ 0 & \text{otherwise}, \end{cases} \tag{25.18}$$

where

$$x_p = \frac{x_{p-}+x_{p+}}{2}, \tag{25.19}$$

$$x_q = \frac{x_{q-}+x_{q+}}{2} \tag{25.20}$$

are the two values for which $f(x) = \frac{A}{2}$, and

$$B = \frac{2}{x_{p-}-x_{p+}} \tanh^{-1}(2p-1), \tag{25.21}$$

$$C = \frac{2}{x_{q-}-x_{q+}} \tanh^{-1}(2q-1). \tag{25.22}$$

Proof. We know that the function

$$f(x) = A\tanh(Bx) \tag{25.23}$$

represents a smooth increase from $-A$ at $x \to -\infty$ to A at $x \to \infty$, crossing through 0 at $x = 0$, with ramp length controlled by B: larger B causes a more sudden ramp. Adding a constant factor 1 and scaling by $\frac{1}{2}$, the function

$$f(x) = \frac{A}{2}\Big[\tanh(Bx)+1\Big] \tag{25.24}$$

represents a smooth increase from 0 to A. As before, we translate the function by replacing the argument by $x - x_p$, but this time x_p refers to the value of x at which $f(x_p) = \frac{1}{2}$, not the value at which we wish ramp-up to begin. Since the function is only asymptotic with $f(x) = 0$ and $f(x) = A$, we can only choose a value of x at which the ramp has already progressed by a certain fraction $p \in (0,1)$. Define x_{p-} to be the value of x at which the function has increased to p of its maximum, $f(x_{p-}) = pA$, and x_{p+} to be the value at which the function is below its maximum by the same factor, $f(x_{p+}) = (1-p)A$. These are illustrated in Figure 25.4. Now, since the hyperbolic tangent is odd, the "midpoint" x_p of the rampup is simply the mean of these points:

$$x_p = \frac{x_{p-}+x_{p+}}{2}, \tag{25.25}$$

$$f(x_p) = \frac{A}{2}. \tag{25.26}$$

We can use this to find the required value of B given chosen p, x_{p-}, x_{p+}:

$$f(x_{p-}) = pA = \frac{A}{2}\Big[\tanh\big[B(x_{p-}-x_p)\big]+1\Big]. \tag{25.27}$$

Use that

$$x_{p-} - x_p \;=\; x_{p-} - \frac{x_{p-} + x_{p+}}{2} \;=\; \frac{x_{p-} - x_{p+}}{2} \tag{25.28}$$

and solve for B to obtain Eq. (25.21).
A function which ramps down from A to 0 is given by reflecting in the y-axis, that is, by replacing $f(x)$ by $f(-x)$. Allowing for new variables q, x_{q-}, x_{q+} such that[2] $f(x_{q-}) = (1-q)A$ and $f(x_{q+}) = qA$, a smooth decrease is given by the function.

$$g(x) \;=\; \frac{A}{2}\Big[\tanh\big[C(-x + x_q)\big] + 1\Big] \tag{25.29}$$

The required value of C is found by setting $f(x_{q+}) = qA$. We multiply these functions together to obtain a single function which begins with values near 0, smoothly ramps up to maintain values around A, and then smoothly ramps down to near 0. Finally, we apply cases to ensure that the function is only defined for $x \in (x_{\mathrm{L}}, x_{\mathrm{U}})$. □

The function described by Eq. (25.18) is plotted in Figure 25.4. The ramp-up rate is controlled by three variables p, x_{p-}, x_{p+} such that $f(x_{p-}) = pA$ and $f(x_{p+}) = (1-p)A$; that is, we choose some fraction p of the ramped-up value A and then decide on the values x_{p-}, x_{p+} at which we wish the function to obtain this fraction. Similarly, the ramp-down rate is controlled by three variables q, x_{q-}, x_{q+} such that $f(x_{q-}) = (1-q)A$ and $f(x_{q+}) = (1-q)A$.

The requirement of more variables makes this slightly more complicated than the exponential function in §25.1.2, but both the ramp-up and ramp-down have smooth onsets and ends. This avoids the potential issue of a discontinuity in the first derivative at the boundaries of the twisted region that we discussed for the exponential function. However, we have now introduced a discontinuity in the field values themselves at the boundary because $f(x_{\mathrm{L}}) > 0$ and $f(x_{\mathrm{U}}) > 0$.

This hyperbolic tangent function is asymptotic with the lines $y = A$ and $y = 0$, so not only does it not address the issue of the exponential function never reaching A, we actually never reach 0 either. Although this may introduce a jump discontinuity in the fields at the boundaries, we claim without proof that the function gets "close enough" to 0 and to A for this not to become an issue. Moreover, the fact that our six parameters $p, x_{p-}, x_{p+}, q, x_{q-}, x_{q+}$ offer so much control over the ramp means we are likely to find values for which this effect is insignificant. Potential jump discontinuities can be mitigated by choosing small p, q (at the cost of a sharper ramp-up and ramp-down) and choosing x_{p-}, x_{q+} to be far from the boundaries of the twisted region (at the cost of a smaller region over which $f(x) \approx A$).

[2] This may seem "the other way around" to our definition of x_{p-}, x_{p+}. We find it the most logical in practice to define the quantities such that $x_{p-} < x_p < x_{p+}$ and $x_{q-} < x_q < x_{q+}$.

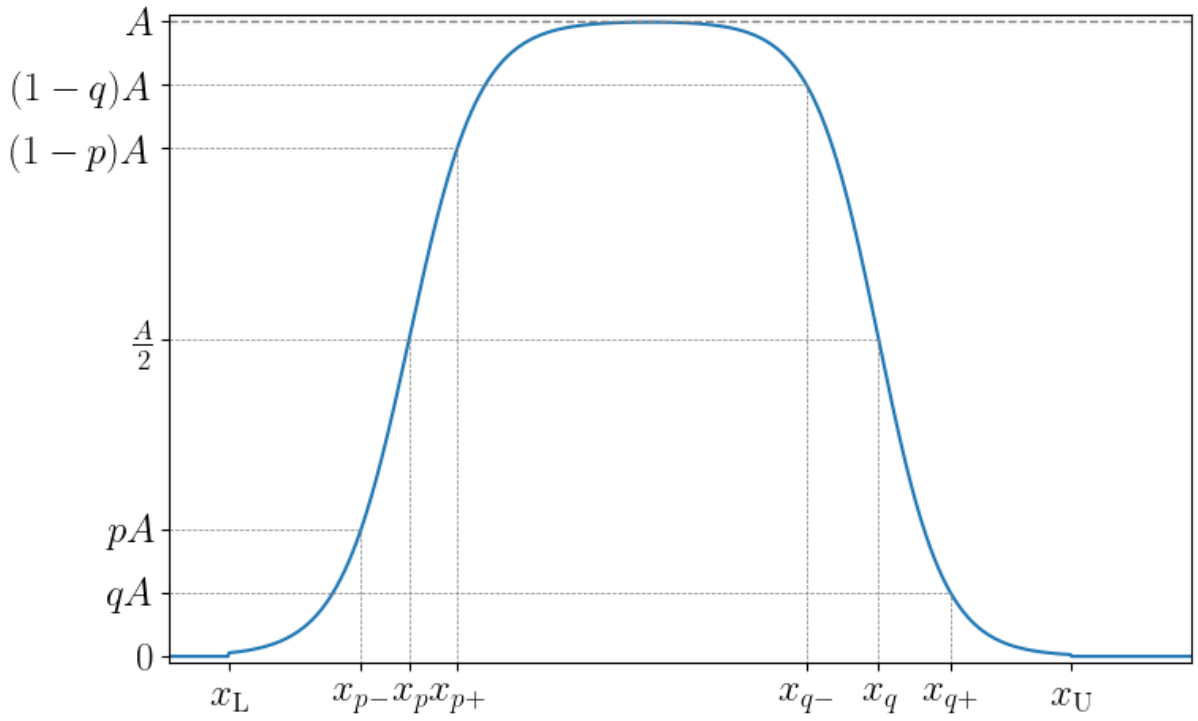

Figure 25.4: Example of a function which smoothly rises from near 0 to A and back over a chosen interval, in one dimension, using hyperbolic tangents, generated by Eq. (25.18). The ramp-up and ramp-down rates are exaggeratedly slowed and deliberately chosen to be different, for illustrative purposes.
Produced by code: `Plot_Smooth_increase_and_decrease_1D_tanh.py`

25.1.4 Gaussian function in 1D

Perhaps the quintessential function which smoothly ramps from 0 to a maximum A and back is a Gaussian

$$f(x) \;=\; A\,\exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right), \tag{25.30}$$

with maximum obtained at $x = x_0$ such that $f(x_0) = A$, and ramp-up controlled by σ. However, as with the hyperbolic tangent in §25.1.3, the function is asymptotic with $f(x) = 0$ and so the enduring issue of jump discontinuities near the boundary is not addressed; further, the values at which the ramp-up begins and the ramp-down ends are difficult to quantify, as before. The greatest limitation of the Gaussian for our application is that it does not plateau near its maximum value A as our previous two constructed functions do, instead reaching it at the single point x_0 and immediately decreasing for higher x. We do not consider the Gaussian function further.

25.1.5 Choice of 1D function

To decide upon a smoothing function to use, we might wish to test various instances of the above functions against each other with a simple twist, and compare which smoothing function allows the strongest twist to persist for the longest time. However, there are so many variables at play (six per dimension, namely $p, q, x_{\rm L}, x_p, x_q, x_{\rm U}$, for the exponential decay and six per dimension, namely $p, q, x_{p-}, x_{p+}, x_{q-}, x_{q+}$, for the hyperbolic tangent) that it becomes increasingly difficult to guarantee a fair comparison. Further, the great number of variables means that either choice will still offer a large amount of fine-tuning to the user.

With this in mind, we forego quantitative tests and conclude based on our arguments in the preceding subsections that the exponential is the optimum smoothing function to implement in the evolution code. It has no jump discontinuities in the function itself, those in the derivative only appear at the onset of the ramp (i.e. not as it tails off near A) and its ramp rate is easily tuned by our defined parameters.

25.2 Smoothing the twist over the coordinates in multiple dimensions

For smoothing over multiple dimensions, we simply multiply by similar factors in the other coordinates. For example, consider a two-dimensional smoothing function in axisymmetric spherical polar coordinates $f(r,\theta)$. Choose the boundaries of the twisted region to be the points $r_1, r_2, \theta_1, \theta_2$. The rates of ramp-up and ramp-down are controlled by specifying the fractions p, q, s, t and then by choosing coordinate values $r_p, r_q, \theta_s, \theta_t$ with $r_1 < r_p < r_q < r_2$ and $\theta_1 < \theta_s < \theta_t < \theta_2$ over which the ramps occur.

These smoothing functions may be applied to *any* twist, not just one with constant Ω. If the twist is described by some function $g(r,\theta)$, this can be substituted for the constant leading factor A in the equation.

25.2.1 Exponential function in 2D

For smoothing with exponential functions,

$$f(r,\theta) = \begin{cases} A\left[1-e^{-B(r-r_1)}\right]\left[1-e^{-C(-r+r_2)}\right]\left[1-e^{-D(\theta-\theta_1)}\right]\left[1-e^{-E(-\theta+\theta_2)}\right] & r_1 \le r \le r_2 \text{ and } \theta_1 \le \theta \le \theta_2, \\ 0 & \text{otherwise,} \end{cases} \tag{25.31}$$

where

$$B = -\frac{1}{r_p - r_1}\ln(1-p), \tag{25.32}$$

$$C = -\frac{1}{-r_q + r_2}\ln(1-q), \tag{25.33}$$

$$D = -\frac{1}{\theta_s - \theta_1}\ln(1-s), \tag{25.34}$$

$$E = -\frac{1}{-\theta_t + \theta_2}\ln(1-t), \tag{25.35}$$

with free parameters $p, q, s, t, r_p, r_q, \theta_s, \theta_t, r_1, r_2, \theta_1, \theta_2$ controlling the ramp-up and ramp-down in both dimensions. Note that the asymptotic nature of the exponential smoothing function is exacerbated as we move from 1D to 2D, so we might expect comparatively less gridpoints within the twisted region to be "arbitrarily close" to A.

25.3 Testing implementation

The smoothing function Eq. (25.31) is used in our evolution code to smooth over the coordinates the contribution of the twist to the electric field, which is done at each timestep. This is achieved in the functions

```
void setup_twist_final_Omega()
void ramp_up_electric_fields_for_twist()
```

We set $p = q = s = t$ to be the same for simplicity. Similarly, we only allow the unsmoothed twist to have a constant value A, as opposed to being a function of the coordinates. The functions can easily be modified if required to relax these restrictions. The controlling parameters are set in the `Header_Initial_Conditions_xx.h`. For this test, let us choose $A = 0.01$, $n_r = 50$, $n_\theta = 101$, $p = q = s = t = 0.99$, $r_1 = r_{\min}$, $r_2 = r_{\max}$, $r_p = r_{\min} + 0.1(r_{\max} - r_{\min})$, $r_q = r_{\max} - 0.1(r_{\max} - r_{\min})$, $\theta_1 = 30°$, $\theta_2 = 40°$, $\theta_s = 34°$, $\theta_t = 36°$.

To verify that the function is implemented correctly, we write the test codes

```
Test codes/20241015_Test_twist_smoothing_function.cpp
Test codes/20241015_Test_twist_smoothing_function_plot.py
```

to calculate values of the smoothed twist using the functions from the evolution code, save them to a CSV file and plot a heatmap of their value normalised to the unsmoothed maximum A. The heatmap is shown in

Figure 25.5; note the gradual onset near all four boundaries of the region within which the twist is defined, and also the relatively large region at which the twist is "arbitrarily close" to its desired maximum value, i.e. the normalised value is near unity. The maximum smoothed function value is 0.00993643, which is around 99.3% of the chosen A, showing that the asymptotic nature of the smoothing function has little detrimental effect on the final values. Note that there is no twist outside the defined twist region, as required.

The code may easily be modified to instead accommodate other functions, by modifying the definition of the function `setup_twist_final_Omega()`.

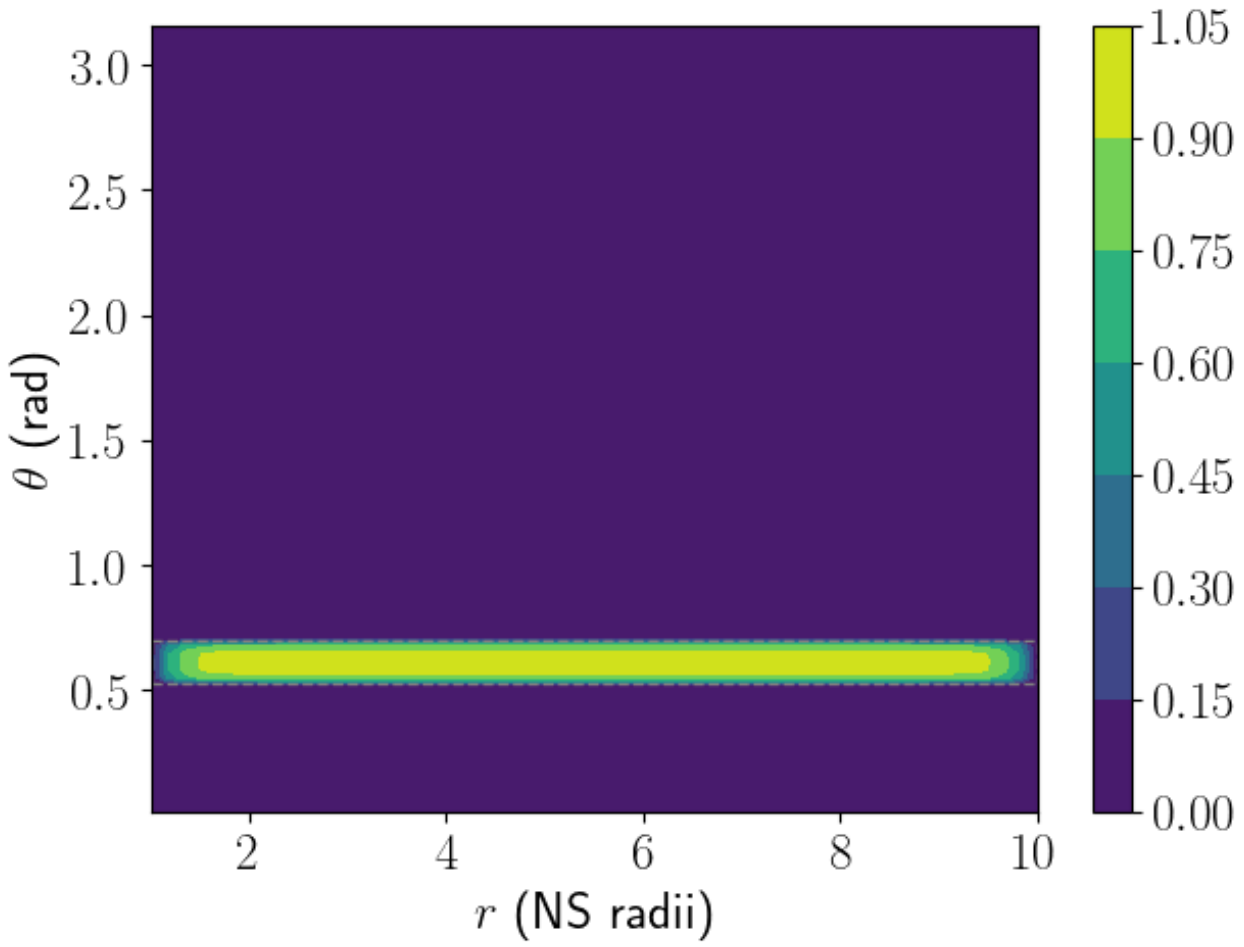

Figure 25.5: Value of 2D exponential smoothing function Eq. (25.31) normalised to its unsmoothed value, applied to a small region $(r_1, r_2, \theta_1, \theta_2)$ within the full domain of the evolution code. The twisted region is annotated with a grey dashed box.

26 Determining suitable number of gridpoints

Let us now decide on suitable numbers of gridpoints for both the radial and angular directions. There is always a tradeoff between execution speed (equivalently computer memory) and accuracy. In principle, one may wish to reduce the resolution if runtimes are too slow, especially for bulk simulations, or increase the resolution if the results are not sufficiently accurate. However, in reality it is not so simple.

For one, we expect diminishing returns as resolution increases, so at some point disproportionately more gridpoints will be needed for meaningful improvements in accuracy. However, we must also be aware of numerical instability. If there exists some instability in the method, then paradoxically we may expect more well-behaved evolution by using fewer gridpoints (Axelrad, 1998). It is easy to fall into the trap of "throwing more resolution at the problem" when results are not as expected, but if the underlying issue is due to instabilities then this can actually exacerbate the situation.

In the tests in this chapter, we consider a dipole which ramps-up to a constant angular velocity and with no twists. We hold the following parameters constant, and vary N_r and N_θ.

```
Parameter values (given in code units, then in SI units)
P_final                       : 44.9688687          0.0015
Omega_final                   : 0.139723001463446   4188.79020478639
r_min, r_max                  : 1                   10.0020751953125
Timestep delta_T              : 0.0030517578125     1.01795683349045e-07
Length of simulation          : 6                   0.000200138457118891 steps: 2000
ell_max                       : 20
Ramp start                    : 1 0.0030517578125
Ramp stop                     : 150 0.457763671875
use_outer_sponge_layer: 1
sigma_0, gamma, beta    : 0.8 6 4
```

26.1 Number of radial gridpoints

We perform three evolutions with $N_r = 25$, $N_r = 50$ and $N_r = 100$. In all of these we use $N_\theta = 1001$, far higher than the values later considered in §26.2; this is a relic of attempting to overcome troublesome evolutions by increasing resolution, as alluded to above.

The variation of total energy with time is plotted in Figure 26.1. The value is calculated by the volume integrals in Eqs. (22.31) and (22.34), and we can calculate the exact value by Eqs. (22.44) and (22.37).

At $t = 0$, the relative error of $U(t)$ goes as N_r^{-2} (Table 26.1), to be expected since it is calculated by a numerical volume integral. At $t = 0$ the relative error for $N_r = 25$ is around 4.17 times that for $N_r = 50$, which in turn is around 4.08 times that for $N_r = 100$. This is less accurate than the $\mathcal{O}(\Delta r^6)$ radial finite difference scheme we chose in §24.1, so our choice of radial finite differencing method is not a limiting factor in the accuracy of our calculations.

As evolution progresses, the graph shows that the value of U is still generally closer to the exact value as resolution increases, and that there are diminishing returns. In all cases, the calculated total energy lags behind the exact value once ramp-up starts, and the lagging persists for longer as more gridpoints are used. Inevitably, some sort of blow-up occurs where the calculated value rapidly increases with time. This blow-up occurs later for higher resolution, and is also less intense (with the energy suddenly spiking to lower values). All of these point to the accuracy of the evolution increasing with N_r, but the inevitable blow-up points to some underlying numerical instability which cannot be avoided by increasing radial resolution.

Table 26.1: Relative error in total energy for various numbers of radial gridpoints.

	Initial, $t = 0$		After ramp-up, $t = 0.457763671875$	
N_r	Calculated U	Relative error	Calculated U	Relative error
25	5.777291159	0.3806059101	8.901958533	0.6600056765
50	4.566346718	0.09122512492	5.377648162	0.002804769694
100	4.278138909	0.02235177348	4.397623266	−0.1799467996
Exact	4.184605554		5.362607284	

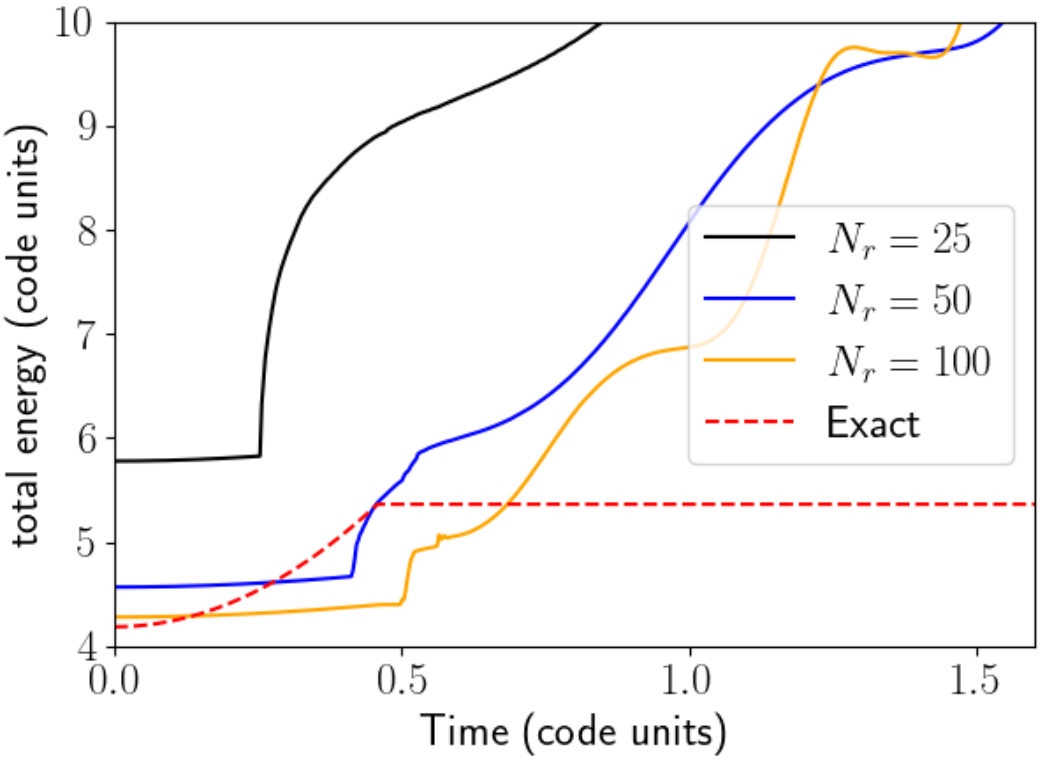

Figure 26.1: Total energy as a function of time for various numbers of radial gridpoints.
Produced by code: `Plot_History_Multiple.py`

Code accuracy can also be measured by ensuring that no divergence leakage occurs, which we quantify across the entire domain by a volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$, Eq. (24.5). The results are plotted in Figure 26.2. We plot the absolute value since the corrective mechanisms from our third-order Adams-Bashforth time integration method lead to flips in sign at each timestep. We expect values to be close to zero and remain constant over time. As Table 26.2 shows, the result at $t = 0$ is near zero in all cases; it is nearer to zero as N_r *decreases*, not increases, but this may simply be due to the volume integral being less accurate for fewer gridpoints. At the end of ramp-up ($t \approx 0.46$ s), the value has already grown significantly to $10^{-4} - 10^{-3}$ and there is no longer a clear trend with N_r. At twice the ramp-up time ($t \approx 0.92$ s), the value appears to have decreased for $N_r = 25$ and $N_r = 50$, but the figure shows that fluctuations are beginning to dominate as the fully-signed value oscillates around zero. Beyond this, there is no discernable difference between the three N_r tested, although interestingly the simulation for $N_r = 100$ had to be terminated early since the evolution became dominated by `nan` values. All of this is consistent with some instability in the evolution, which cannot be prevented by increasing radial resolution.

Table 26.2: Absolute value of dimensionless volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ as a function of time, for various numbers of radial gridpoints.

N_r	$t = 0$	$t = 0.457763671875$	$t = 0.91552734375$
25	3.29513987898665e-15	0.0019056898700025	0.00076294184054962
50	1.88484521320713e-15	0.00471827173856188	0.000350499547531265
100	1.30098354243529e-14	0.000601661204676431	0.131333682616569

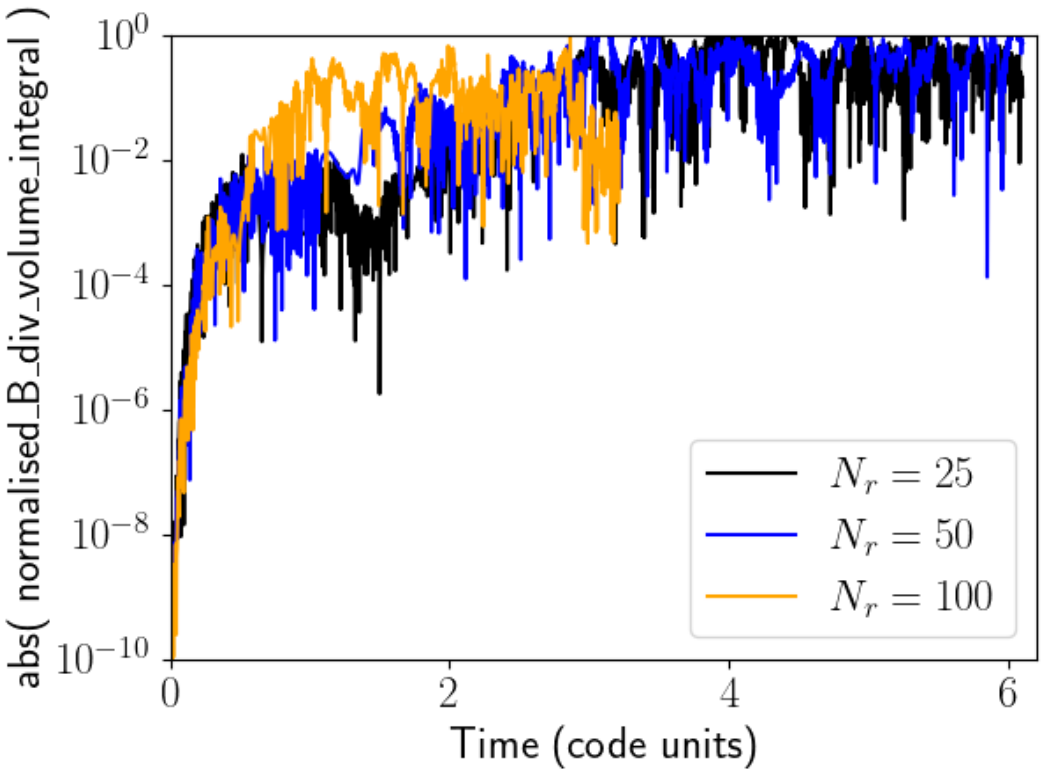

Figure 26.2: Dimensionless volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ as a function of time for various numbers of radial gridpoints.

Finally, let us check the standard deviation of the VSH decomposition of each field component, compared to its calculated value, as a function of time. The VSH decompositions depend on integrals over θ and we take radial derivatives as part of the evolution, so the accuracy of evolution depends on both coordinates. The standard deviation is plotted as a function of time for each vector component in Figure 26.3. At early times there is little discernible difference between the simulations for $N_r = 25$ and $N_r = 50$, but for $N_r = 50$ the standard deviation generally begins its uncontrollable increase earlier. The run with $N_r = 100$ causes the VSH decompositions to become far less accurate as time progresses. Again, this is consistent with a numerical instability.

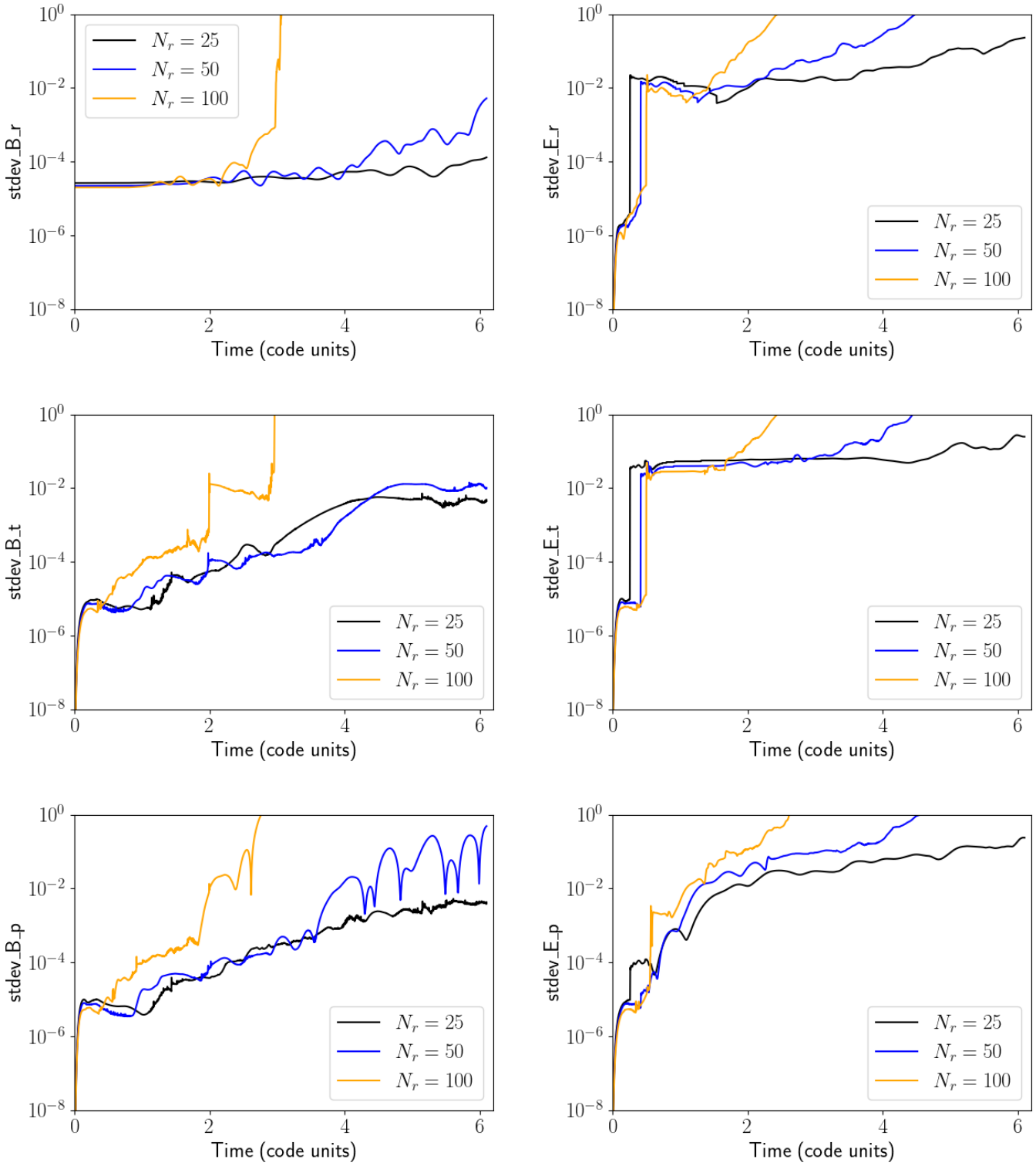

Figure 26.3: Standard deviation of VSH decomposition of field components, compared to the current numerical value, as a function of time for various numbers of radial gridpoints.

26.2 Number of angular gridpoints

We perform four evolutions with $N_\theta = 25$, $N_\theta = 49$, $N_\theta = 101$ and $N_\theta = 201$. These values are all of the form $4n + 1,\ n \in \mathbb{Z}$, which we find gives the best performance.

The variation of total energy with time is plotted in Figure 26.4. At $t = 0$, the four simulations yield similar results but all displaced from the exact value, indicating that error at this point is dominated by the radial coordinate. If we correct for radial error by taking the values for $N_\theta = 201$ to be exact, then the relative error roughly goes as N_θ^{-2}: that for $N_\theta = 25$ is around 4.18 times that of $N_\theta = 49$, which is around 5.45 times that of $N_\theta = 101$.

The blow-up in energy highlighted in §26.1 is delayed as N_θ increases, but its intensity appears unaffected. We suggest that the numerical instability is radial in nature, and that increasing angular resolution may help to stave it off, but it is still inevitable. The total energy at the end of ramp-up is closer to the exact value as N_θ increases, but is largely affected by having already begun to blow up. Still, the relative error taking $N_\theta = 201$ to be exact, is around 3.49 times higher for $N_\theta = 25$ than for $N_\theta = 49$, which is around 4.72 times higher than for $N_\theta = 101$, so the scaling as N_θ^{-2} appears to be preserved as evolution progresses.

Table 26.3: Relative error in total energy for various numbers of angular gridpoints.

	Initial, $t = 0$			After ramp-up, $t = 0.457763671875$		
N_θ	Calculated U	Relative error	R.e. to $N_\theta = 201$	Calculated U	Relative error	R.e. to $N_\theta = 201$
25	4.559840711	0.08967037685	−0.001405061965	7.701469166	0.4361426742	0.344365137
49	4.564721062	0.09083664014	−0.0003362757984	6.29414423	0.1737097081	0.09870310299
101	4.565974941	0.09113628098	−0.0000616791601	5.848491222	0.09060591469	0.02091010612
201	4.566256584	0.0912035855		5.728703426	0.06826831095	
Exact	4.184605554			5.362607284		

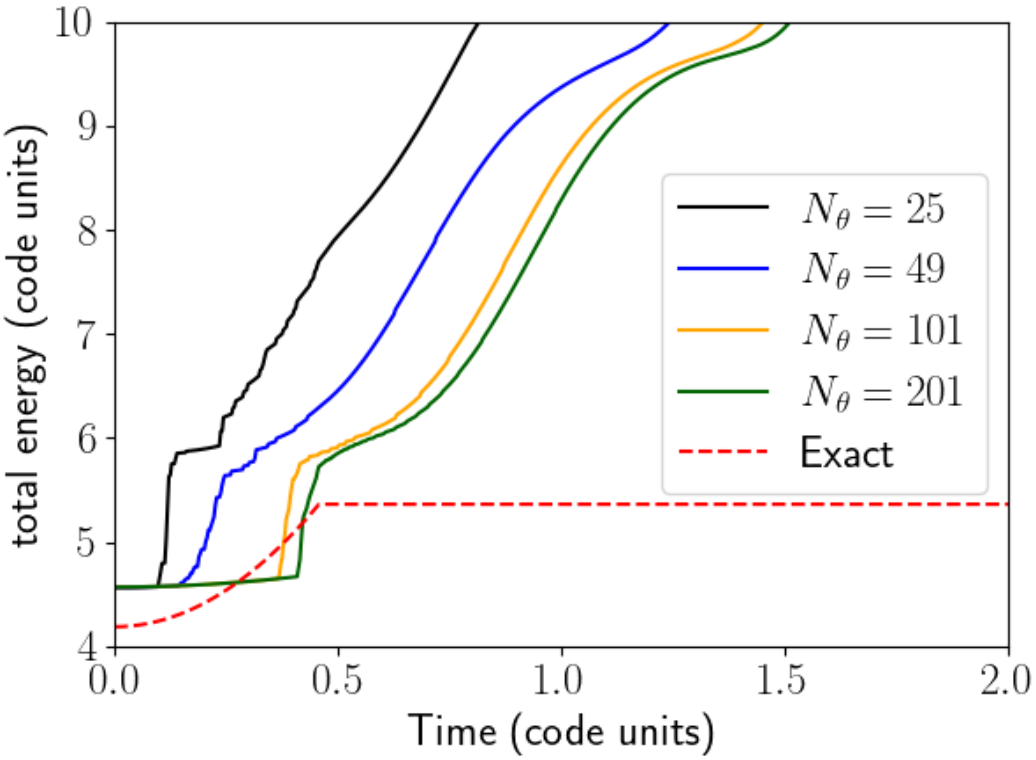

Figure 26.4: Total energy as a function of time for various numbers of angular gridpoints.

Let us now check the constancy of our volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ with time. At $= 0$, the result is near zero for all cases except $N_\theta = 25$ (Table 26.4), and is nearer to zero as N_θ increases, as expected. At later times times, fluctuations are dominating as we saw for the radial test. Perhaps Figure 26.5 may suggest that the value grows at a slower rate as N_θ increases, and hence remains more well-behaved for longer, but the table shows that fluctuations make direct comparisons between runs at any single timestep impossible. We conclude that the angular resolution is not the limiting factor in the behaviour of $\boldsymbol{\nabla} \cdot \mathbf{B}$ with time.

Table 26.4: Absolute value of dimensionless volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ as a function of time, for various numbers of angular gridpoints.

N_θ	$t = 0$	$t = 0.457763671875$	$t = 0.91552734375$
25	0.000191827912407487	0.001188842800387	0.237620633666942
49	6.26071545501263e-16	0.00542484313333733	0.0956445523892594
101	3.83966479295651e-15	0.0263119555645653	0.068276040844469
201	1.90131289900207e-15	0.00702932670973856	0.0190707697404965

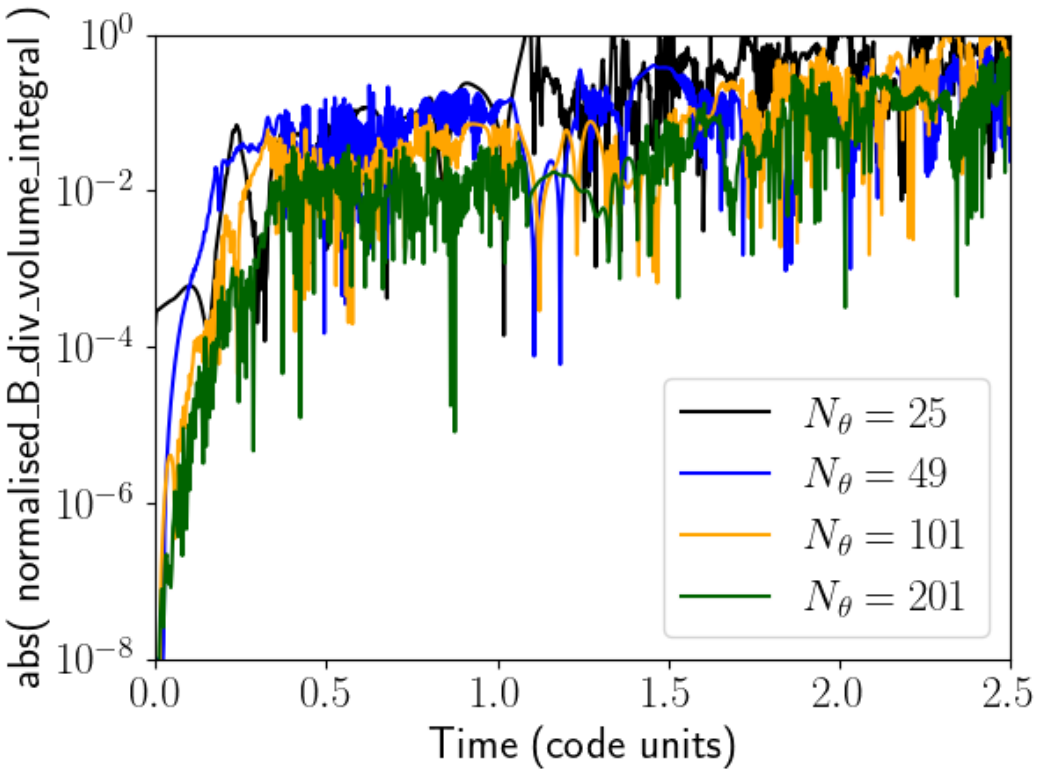

Figure 26.5: Dimensionless volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ as a function of time for various numbers of angular gridpoints.

The standard deviation of the VSH decomposition of each vector component, compared to its calculated numerical value, is plotted in Figure 26.6. The accuracy of the VSH decomposition remains higher for longer as N_θ increases, but we see diminishing returns between $N_\theta = 101$ and $N_\theta = 201$.

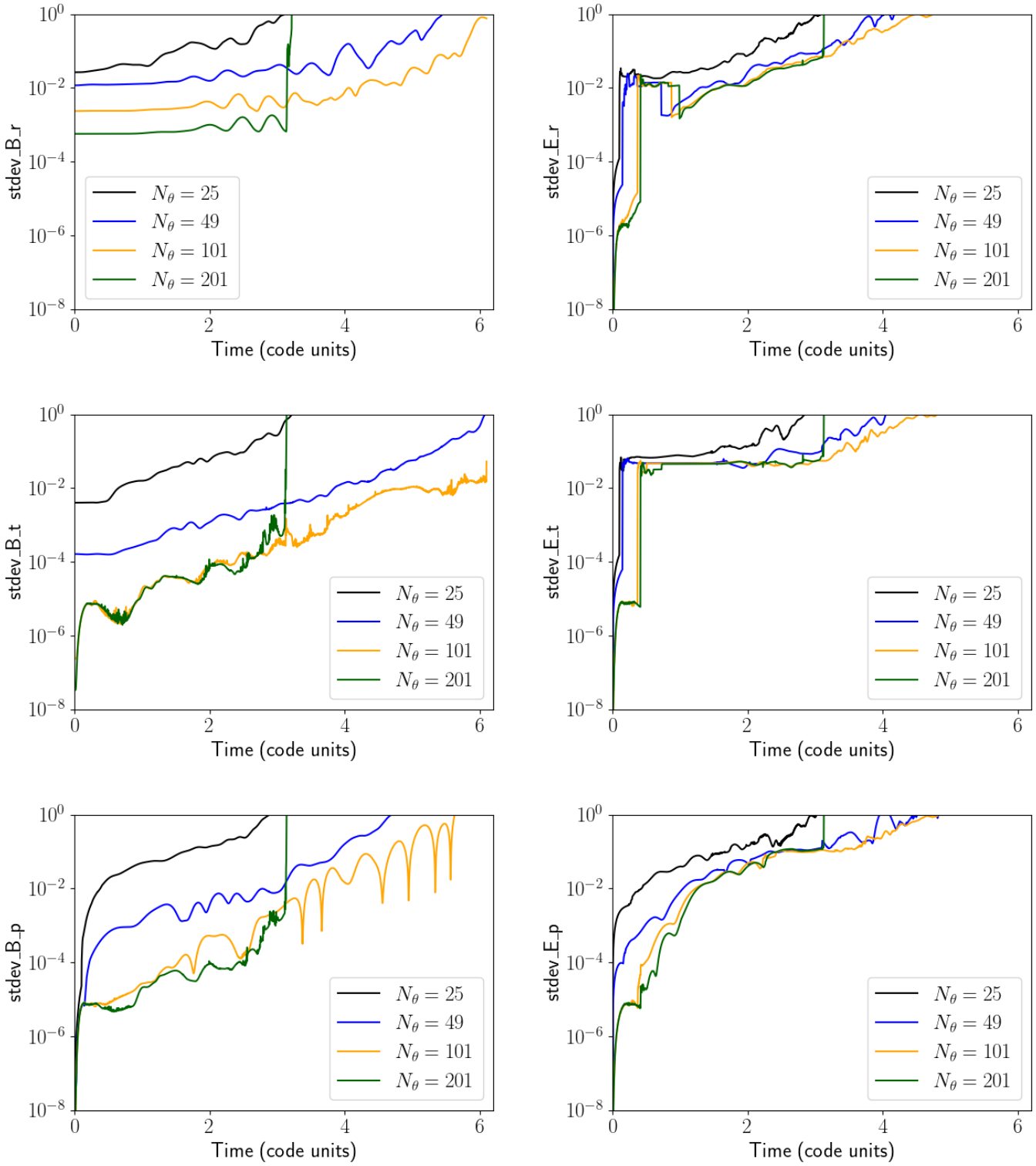

Figure 26.6: Standard deviation of VSH decomposition of field components, compared to the current numerical value, as a function of time for various numbers of angular gridpoints.

26.3 Conclusion

We choose $N_r = 50$ and $N_\theta = 101$ as a compromise between code performance and accuracy. Accuracy may be slightly improved by increasing N_θ, but increasing N_r may cause further issues. The code is limited by the existence of some numerical instability. These coordinate choices correspond to a CFL maximum timestep of $(\Delta t)_{\max} \approx 0.0314$ by Eq. (23.35), comfortably above our chosen timestep.

Part IV

Simulations and results

27 Stationary dipole

In these chapters, we describe the results of three rudimentary simulations: a stationary magnetic dipole, a rotating dipole (simulating a pulsar) and a stationary dipole which develops a simple magnetospheric twist (simulating a magnetar).

First, let us consider a magnetic dipole without rotation. In reality this is a steady, stable configuration which we expect to persist for all times without changing. Thus, a simulation which persists for long times without significant deviation in the values of the magnetic or electric fields may be used as an indicator that the numerical scheme is stable. We use the following parameters:

```
P_final                     : 44.9688687          0.0015
Omega_final                 : 0                   0
R_LC_max                    : 0                   0
r_min, r_max                : 1                   10.0020751953125
n_r, delta_r                : 50                  0.1837158203125
n_t, delta_t                : 101                 0.0314159265358979
Timestep delta_T            : 0.0030517578125     1.01795683349045e-07
Length of simulation        : 9                   0.000300207685678337 steps: 3000
ell_max                     : 20
Ramp start                  : 100001 305.178833007812
Ramp stop                   : 100150 305.633544921875

use_outer_sponge_layer: 1
sigma_0, gamma, beta    : 0.8 6 4
```

To disable rotation, we set the times for the start of rotation and twist ramp-up to after the simulation has finished, e.g.

```
T_index_rotation_ramp_start = T_index_max + 1;
T_index_rotation_ramp_stop  = T_index_max + 1;
T_index_twist_ramp_start    = T_index_max + 1;
T_index_twist_ramp_stop     = T_index_max + 1;
```

It may be tempting to simply set `P_SI_final = 0` and `twist_final_Omega = 0`, but recall from §20.5 that a separate set of outer boundary conditions are used once rotation ramp-up begins.

In addition to $\ell_{\max} = 20$, we also perform a simulation with $\ell_{\max} = 1$, since from Proposition 22.2 this should be enough to exactly reproduce the magnetic field. Comparing the two, we can characterise the extent to which numerical error from overfitting with high values of ℓ affects simulations.

First, let us consider the values of the fields at an arbitrarily chosen gridpoint as a function of time,

```
( r[20], theta[20] ) = ( 4.67431640625, 0.644026493985908 )
```

These can be compared to the known exact values for a non-rotating aligned dipole, Eq. (22.2) with $A(\Psi) = 0$. Figure 27.1 shows the time-evolution of B_r, B_θ and E_ϕ at this gridpoint; all other field components remain zero throughout the evolution, as expected.

The values of B_r and B_θ remain close to their theoretical values for around two light-crossing times, before which numerical instability begins to affect the evolution. Our third-order Adams-Bashforth time integration works well to minimise the growth of noise by causing the field values to oscillate about their exact values. However, after around four light-crossing times the instability becomes too great. The introduction of a nonzero E_ϕ is a further indicator of this instability. Limiting $\ell_{\max} = 1$ yields little qualitative difference in the evolution, giving confidence that the numerical scheme is relatively insensitive to the choice of $\ell_{\max}$.

This result shows that the code is capable of relatively stable evolution, at least for a few light-crossing times, and that the time-evolution method works to reduce the growth of numerical errors.

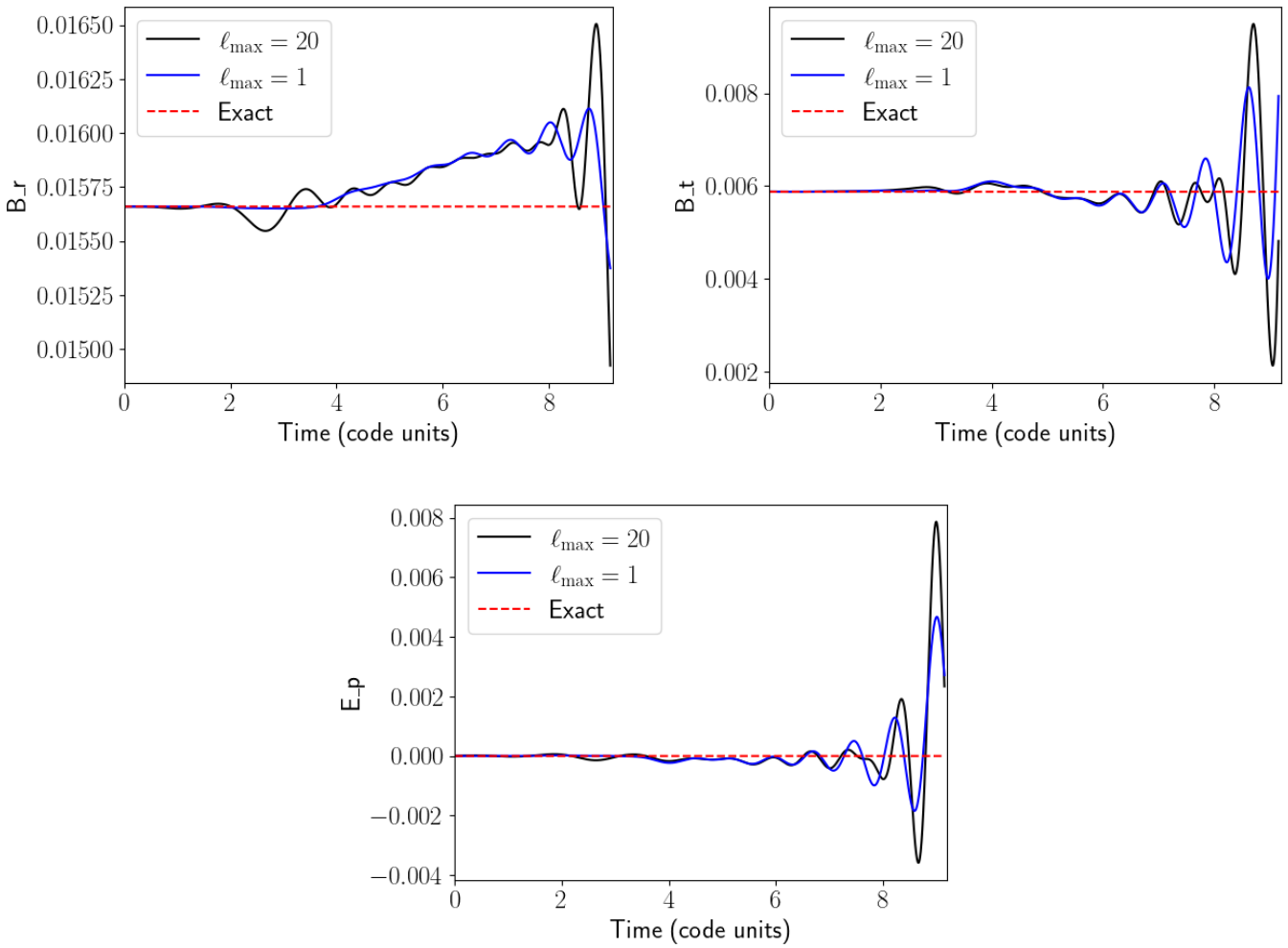

Figure 27.1: Time-evolution of B_r, B_θ and E_ϕ at a single gridpoint for the simulated non-rotating dipole.

A similar story is told by the standard deviation of the VSH decomposition, Figure 27.2. Here, we see that the uncontrollable rise occurs first in E_ϕ, which should not be surprising since nonzero E_ϕ is a direct result of numerical error in the time-evolution. We also see that B_θ begins to lose accuracy long before B_r, after around 2 light-crossing times. In these figures, $\ell_{\max} = 1$ yields far more accurate VSH decompositions.

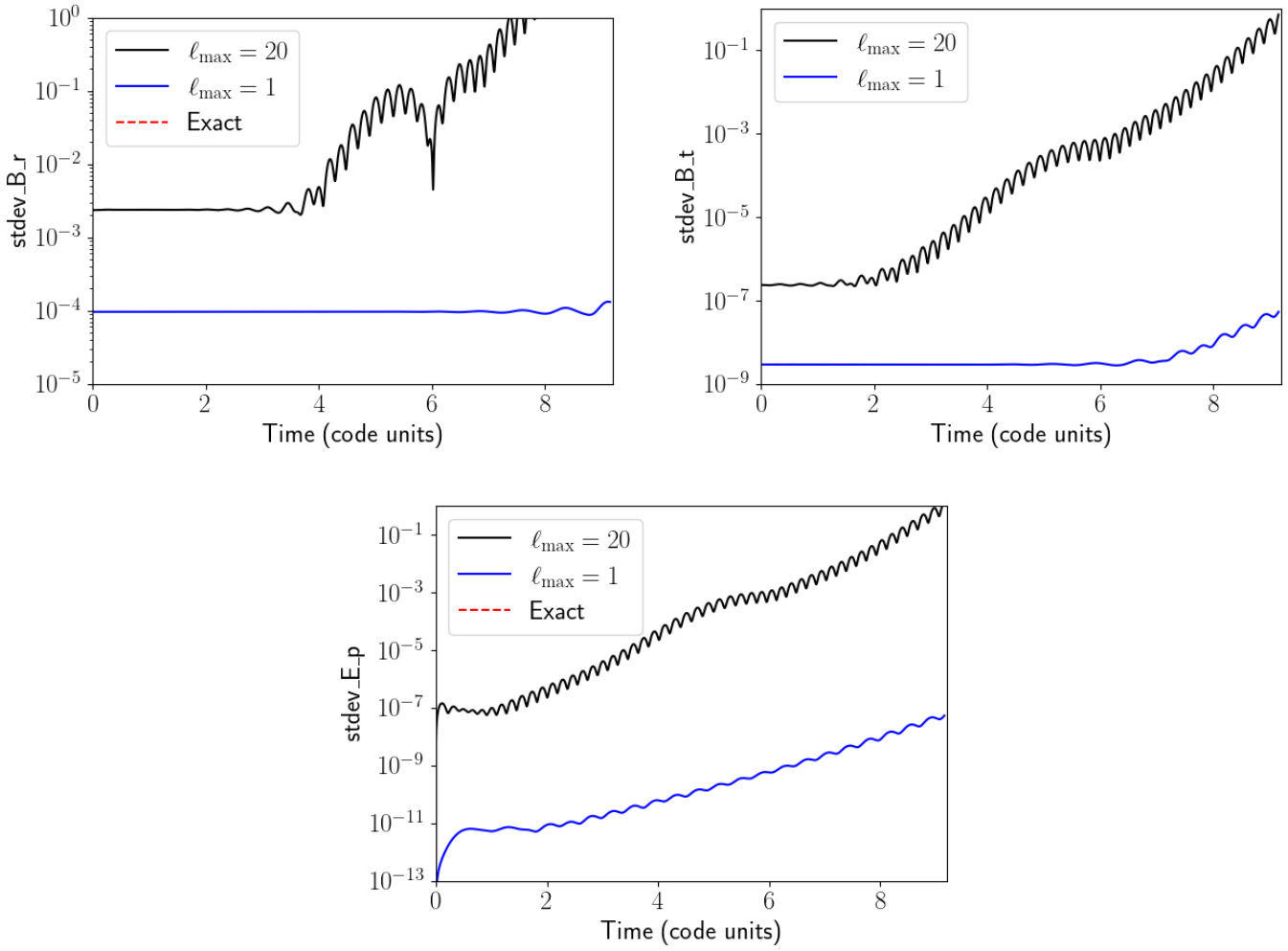

Figure 27.2: Standard deviation of VSH decomposition of field components, compared to the current numerical value, as a function of time, for the simulated non-rotating dipole.

Figures 27.3, 27.4 and 27.5 explore this in more detail. Note that the north pole $\theta = 0$ is not included because we offset the θ values by half a gridpoint (§23.2). The configuration remains symmetric about the equatorial line $\theta = \frac{\pi}{2}$ at all times.

For B_r, the field is largely unchanged until around four light-crossing times, after which the values near the outer boundary begin to oscillate widly around zero. This region of gridpoints with high $|B_r|$ begins to proliferate outwards.

For B_θ, the onset is much sooner. There also develops a region of large B_θ near the outer boundary after four light-crossing times, which grows in magnitude and extent.

Very early in the evolution, regions of significantly nonzero E_ϕ develop near both the inner and the outer boundary. Although $E_\phi = 0$ at $r_{\min}$ and $r_{\max}$ are boundary conditions we enforce (§20.4 and §20.5), adjacent values are not controlled for. These grow both in magnitude and in extent over time, eventually meeting at around two light crossing times.

One potential cause of this finer structure that appears may be an unrealistically high $\ell_{\max}$. Note however that it is concentrated near the inner boundary, where the density of gridpoints is far higher due to our linear spacing (§23.2). Perhaps a smoother evolution may be gained by employing some other coordinate system; Parfrey (2012) explores various mappings in both the radial and angular directions which concentrate gridpoints where they are needed.

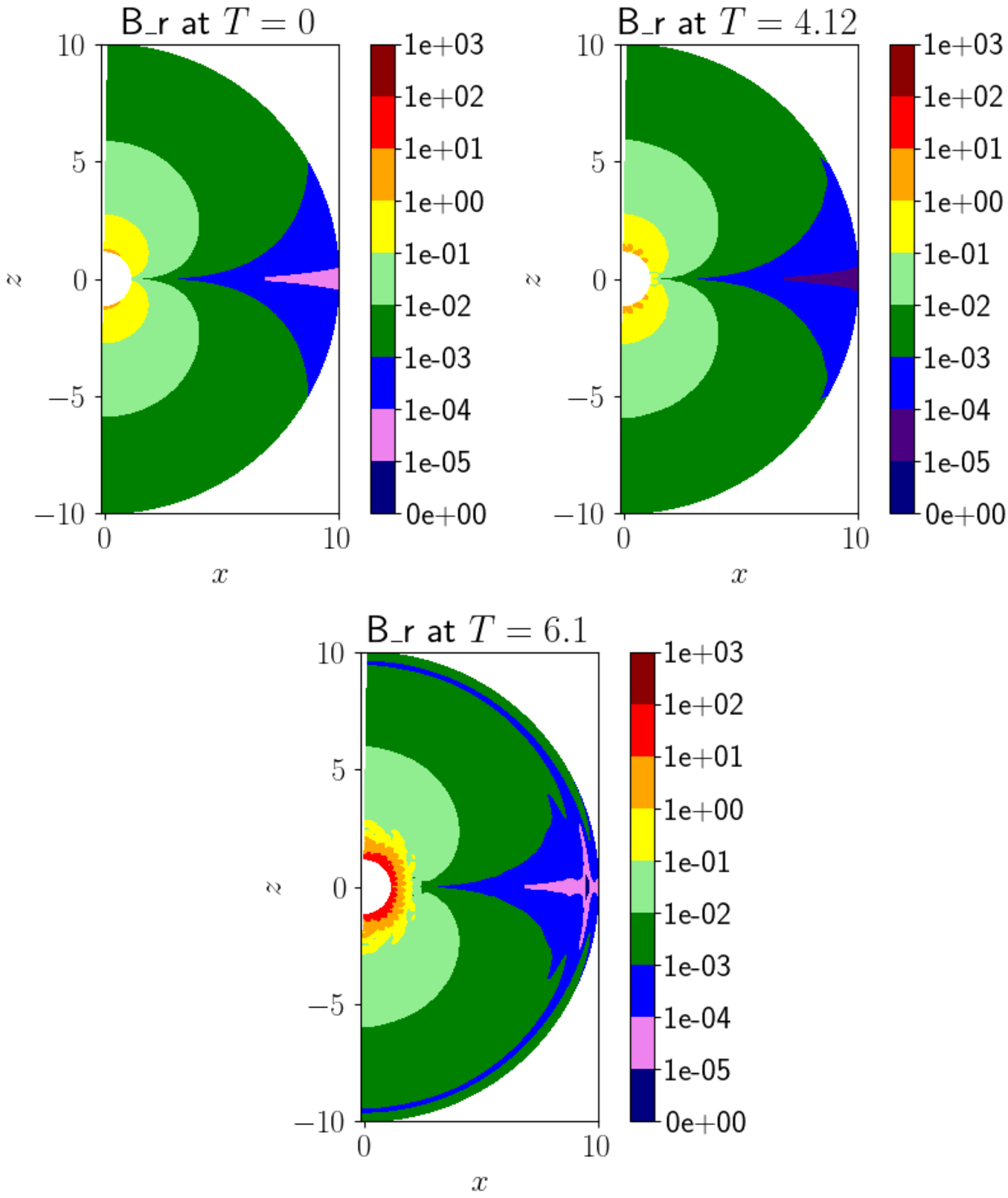

Figure 27.3: Contour plot of B_r at various timepoints for the non-rotating model.

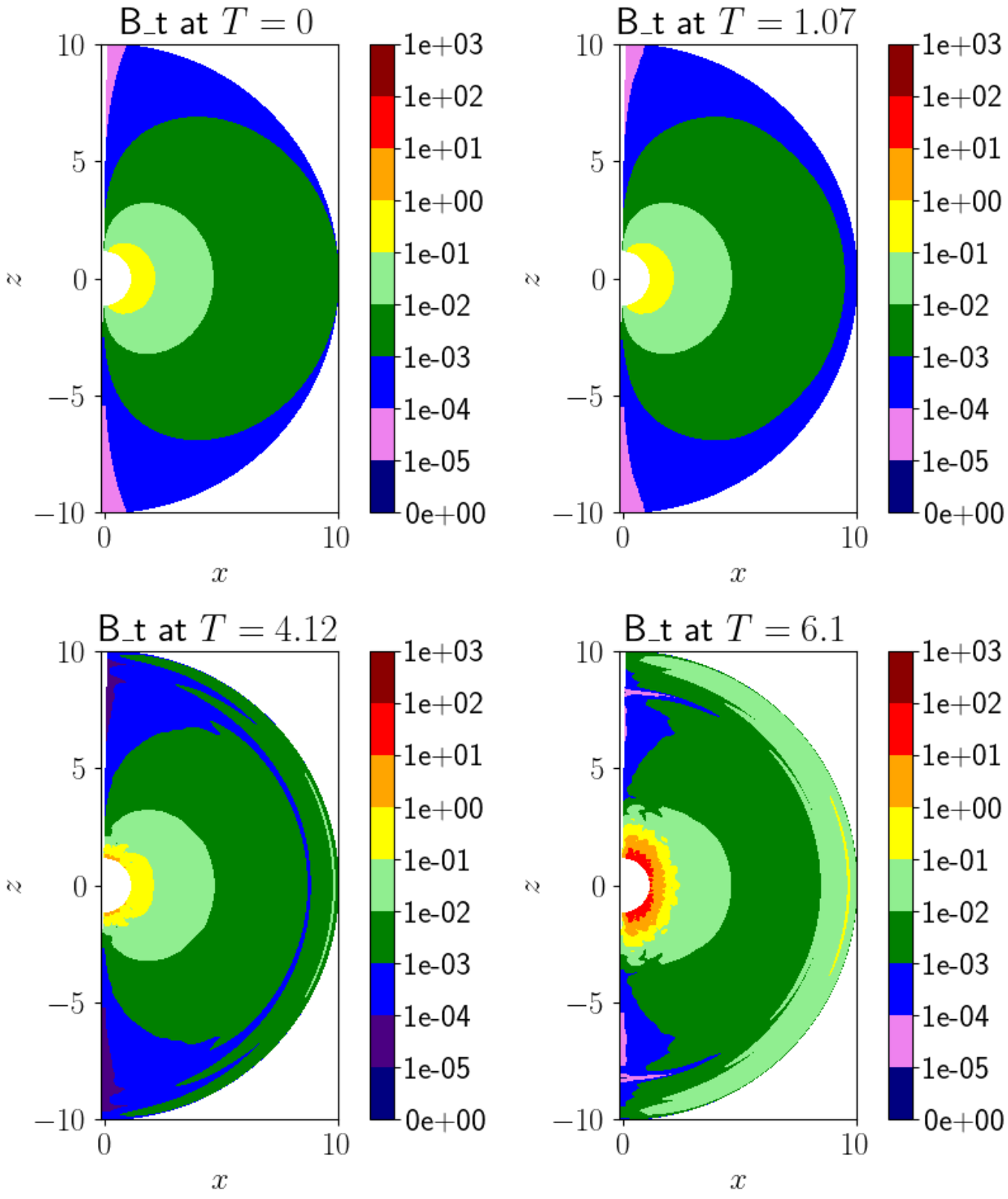

Figure 27.4: Contour plot of B_θ at various timepoints for the non-rotating model.

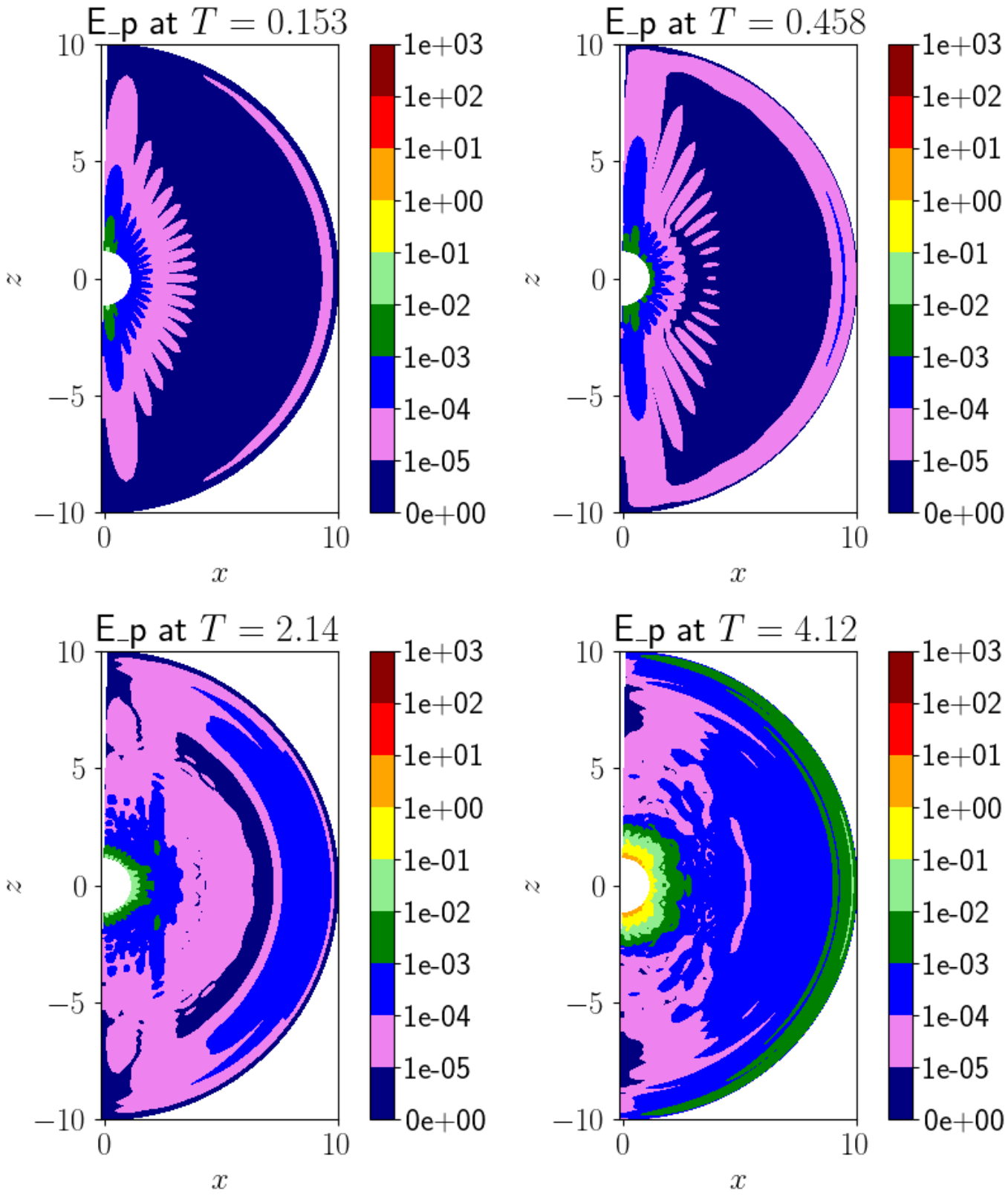

Figure 27.5: Contour plot of E_ϕ at various timepoints for the non-rotating model.

Figures 27.6 and 27.7 plot the total energy and the non-dimensionalised volume integral of $\nabla \cdot \mathbf{B}$, tests of the numerical stability of the simulation and its susceptibility to leaking of energy out of the domain. Both figures support our earlier findings that evolution is stable for a few light-crossing times. They also suggest that refining $\ell_{\max}$ can have a stabilising effect if physical arguments can be made to limit it, but it cannot prevent the simulation from blowing up, only delaying it.

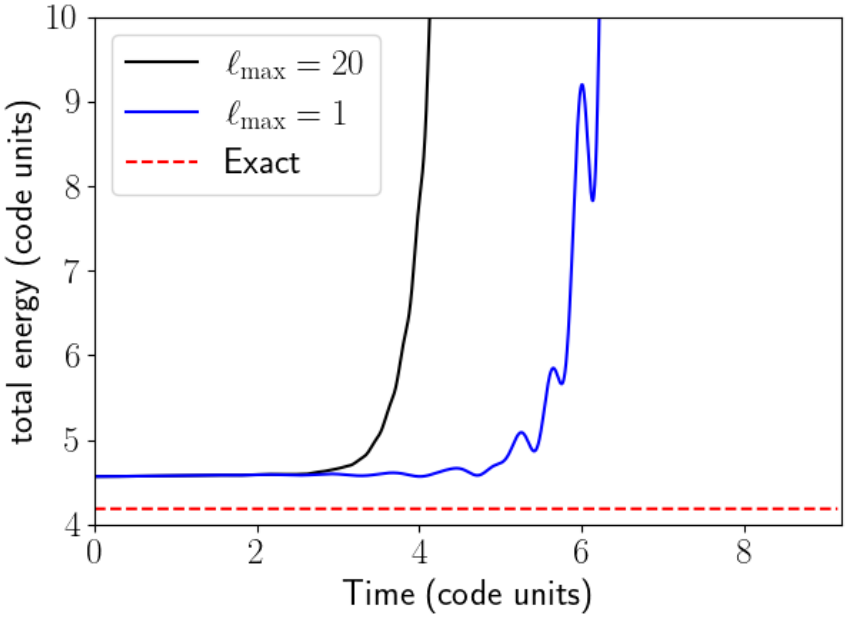

Figure 27.6: Total energy as a function of time for the non-rotating model.

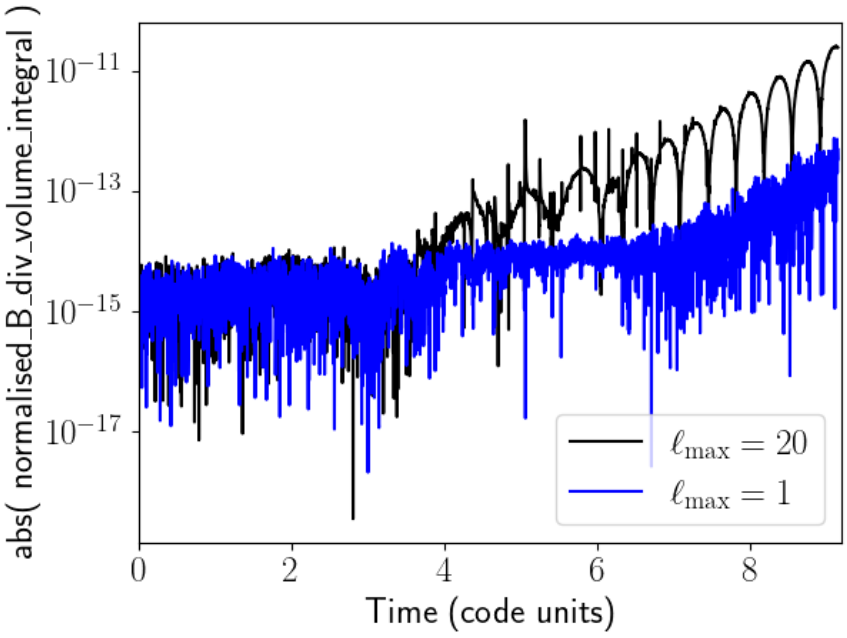

Figure 27.7: Normalised volume integral of $\nabla \cdot \mathbf{B}$ across the domain as a function of time for the non-rotating model.

28 Pulsar model (rotating dipole)

With the stability of the code demonstrated, let us now allow the model to rotate in order to simulate a Goldreich-Julian pulsar. We use the following parameters:

```
P_final                     : 44.9688687          0.0015
Omega_final                 : 0.139723001463446   4188.79020478639
R_LC_max                    : 7.15701773885541    71570.1773885541
r_min, r_max                : 1                   10.0020751953125
n_r, delta_r                : 50                  0.1837158203125
n_t, delta_t                : 101                 0.0314159265358979
Timestep delta_T            : 0.0030517578125     1.01795683349045e-07
Length of simulation        : 9                   0.000300207685678337 steps: 3000
ell_max                     : 20
Ramp start                  : 1 0.0030517578125
Ramp stop                   : 150 0.457763671875

use_outer_sponge_layer: 1
sigma_0, gamma, beta    : 0.8 6 4
```

Figure 28.1 shows the field components evaluated the same gridpoint chosen in the previous evolution (Chapter 27). Interestingly, the simulation appears to run stably for longer than the non-rotating case, with significant deviations in the magnetic field values only after around 6 light-crossing times. The electric field follows the theoretical value Eq. (22.25) very closely during the ramp-up, and remains stable at this value for around two light-crossing times before some numerical instability sets in.

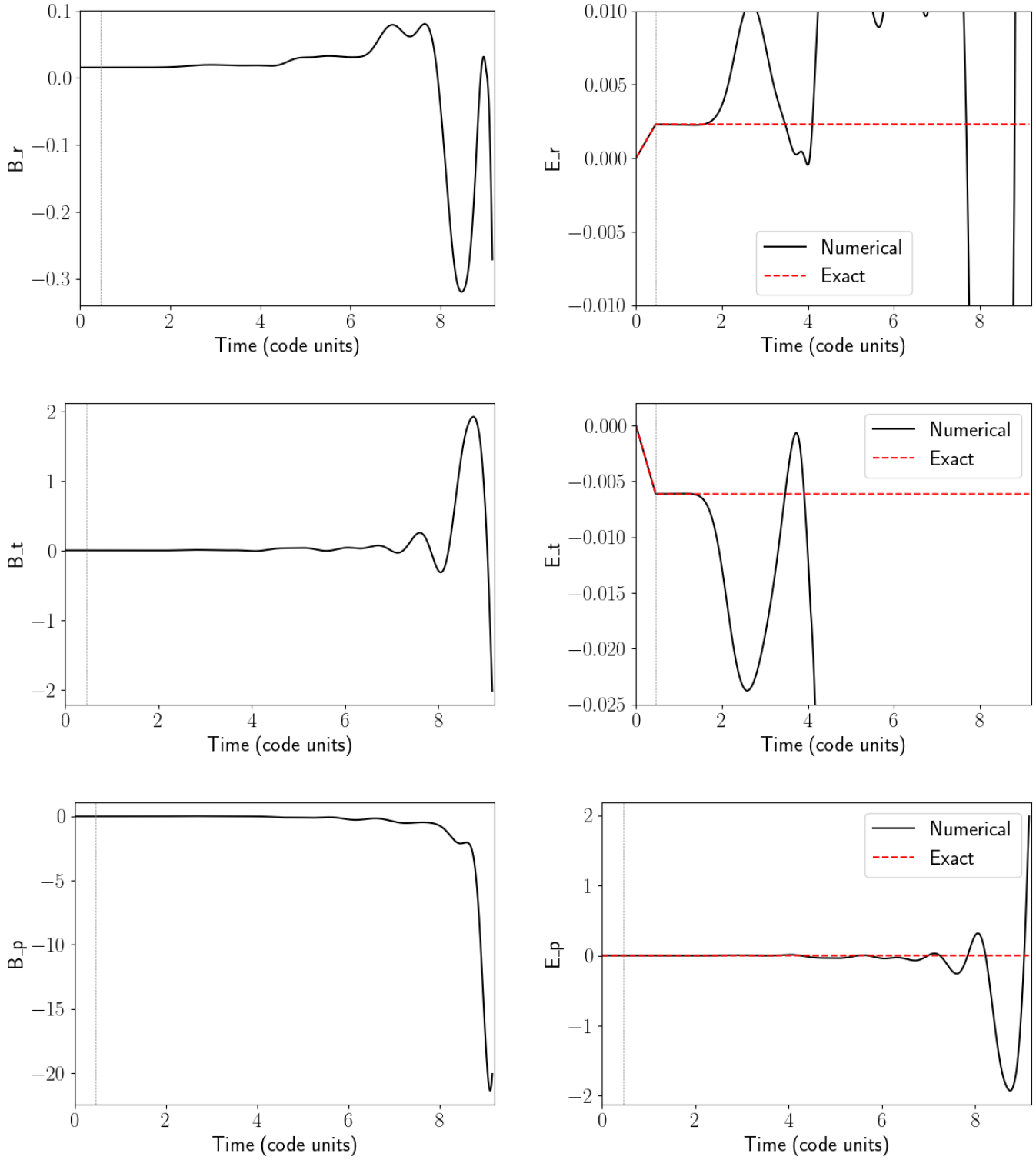

Figure 28.1: Time-evolution of field components at a single gridpoint for the simulated rotating dipole. The dashed grey vertical lines represent the end of the ramp-up.

The standard deviation of the VSH decomposition is shown in Figure 28.2. That of B_r is most poorly represented, even at the start of the evolution, which is surprising since it appears very stable in Figure 28.1. There is a spike in the components of **E** just before the end of ramp-up, after which the values briefly stabilise before beginning an uncontrollable rise. The standard deviations of the components of **B** rise steadily throughout the evolution.

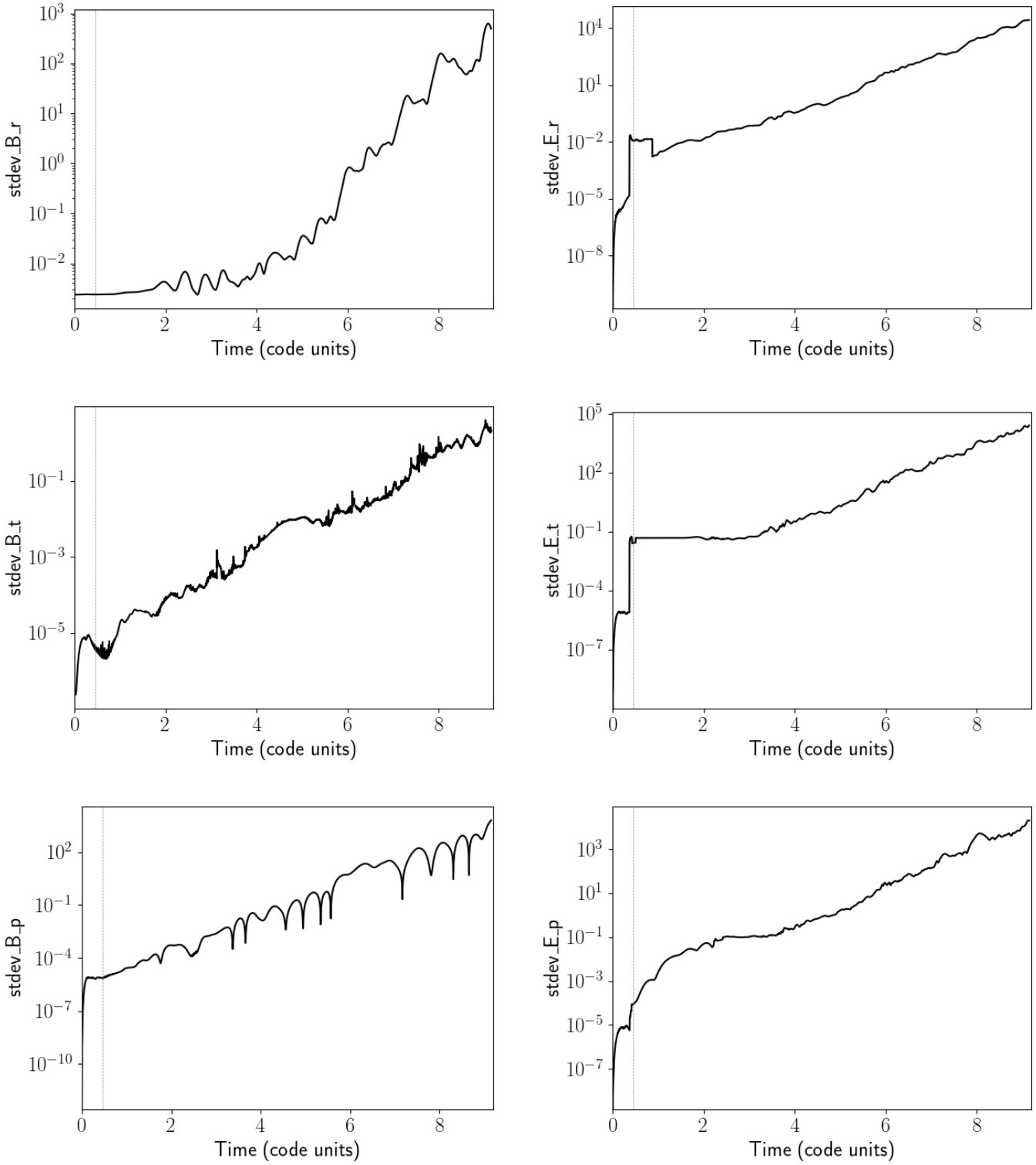

Figure 28.2: Time-evolution of standard deviation of VSH decomposition of field components at a single grid-point for the simulated rotating dipole. The dashed grey vertical lines represent the end of the ramp-up.

Figures 28.3 to 28.8 show contour plots of the field components at various points in time. B_r only begins to show variation after around two light-crossing times, at which point large values appear at the inner boundary and the smooth structure begins to become more complex. After around four light-crossing times, the value at the outer boundary also begins to increase. B_θ shows variation almost immediately, increasing in the region around the equatorial line at the outer boundary and decreasing around the poles at the outer boundary. Nonzero B_ϕ appears around both boundaries almost immediately, growing in both magnitude and physical extent until a relatively complex structure is produced after around a light-crossing time. E_r ramps up to its expected value and soon begins to increase near the inner boundary and decrease near the outer boundary. E_θ retains its structure for longer, before increasing near the inner and outer boundaries. E_ϕ shows a complicated evolution, growing quickly during the ramp-up around both boundaries and continuing to rise after.

The evolution of all field components appears strongly influenced by the inner and outer boundaries; efforts to improve code performance in the future might begin by considering these in greater detail. As for the non-rotating case, the complicated structures that develop may simply be due to the choice of $\ell_{\max}$ or the chosen gridpoint distribution.

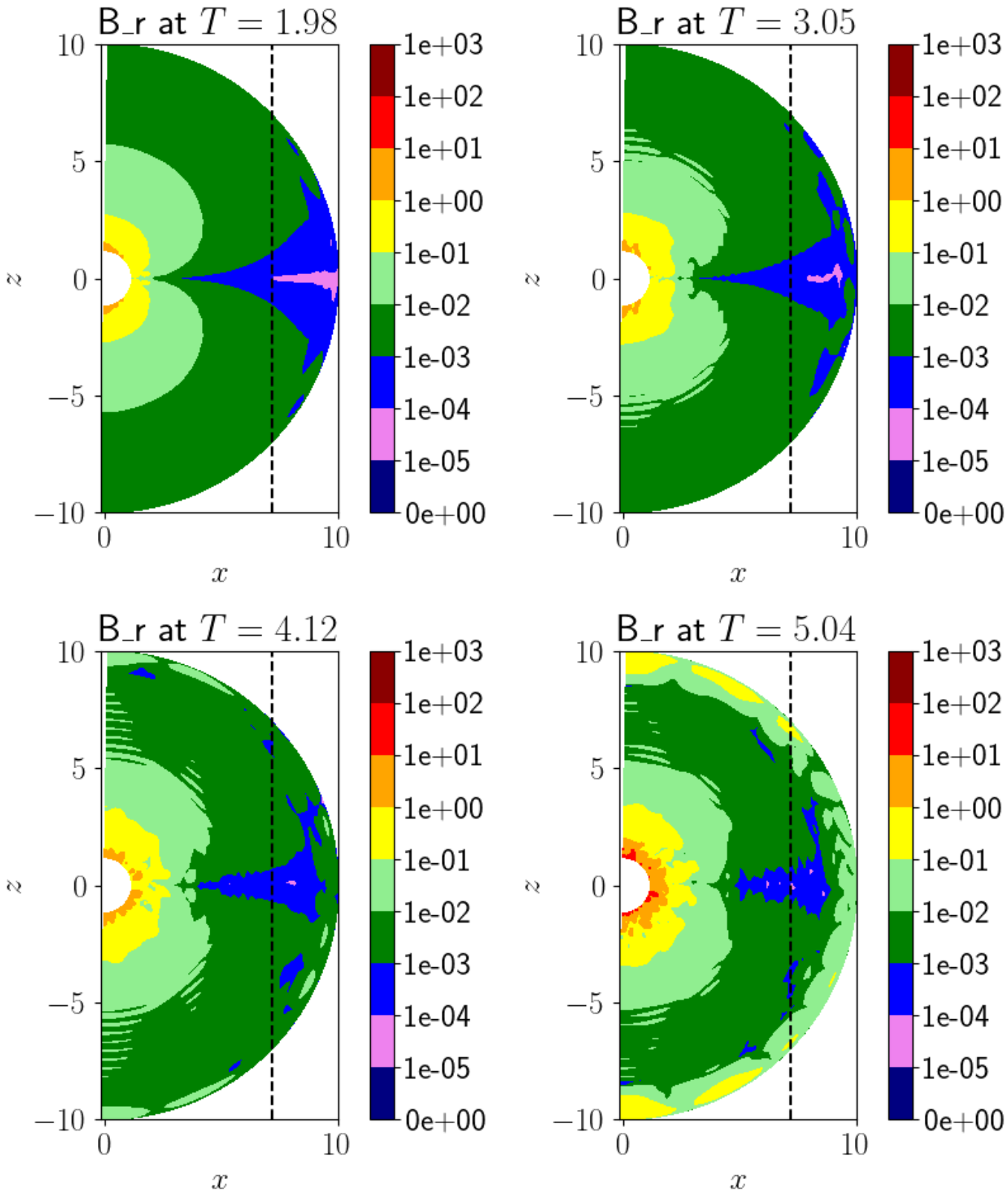

Figure 28.3: Contour plot of B_r at various timepoints for the rotating model. The dashed black line represents the light cylinder.

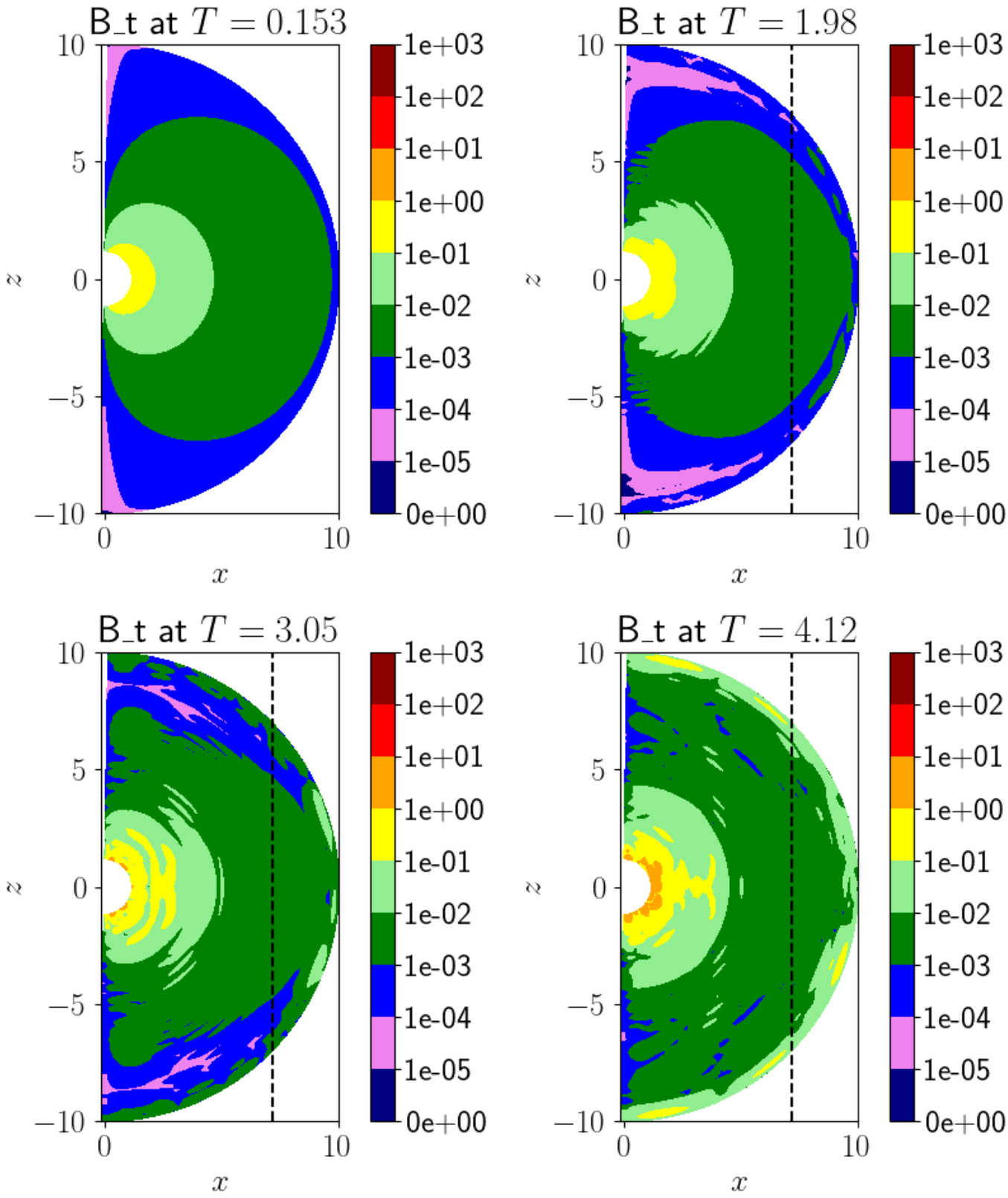

Figure 28.4: Contour plot of B_θ at various timepoints for the rotating model. The dashed black line represents the light cylinder.

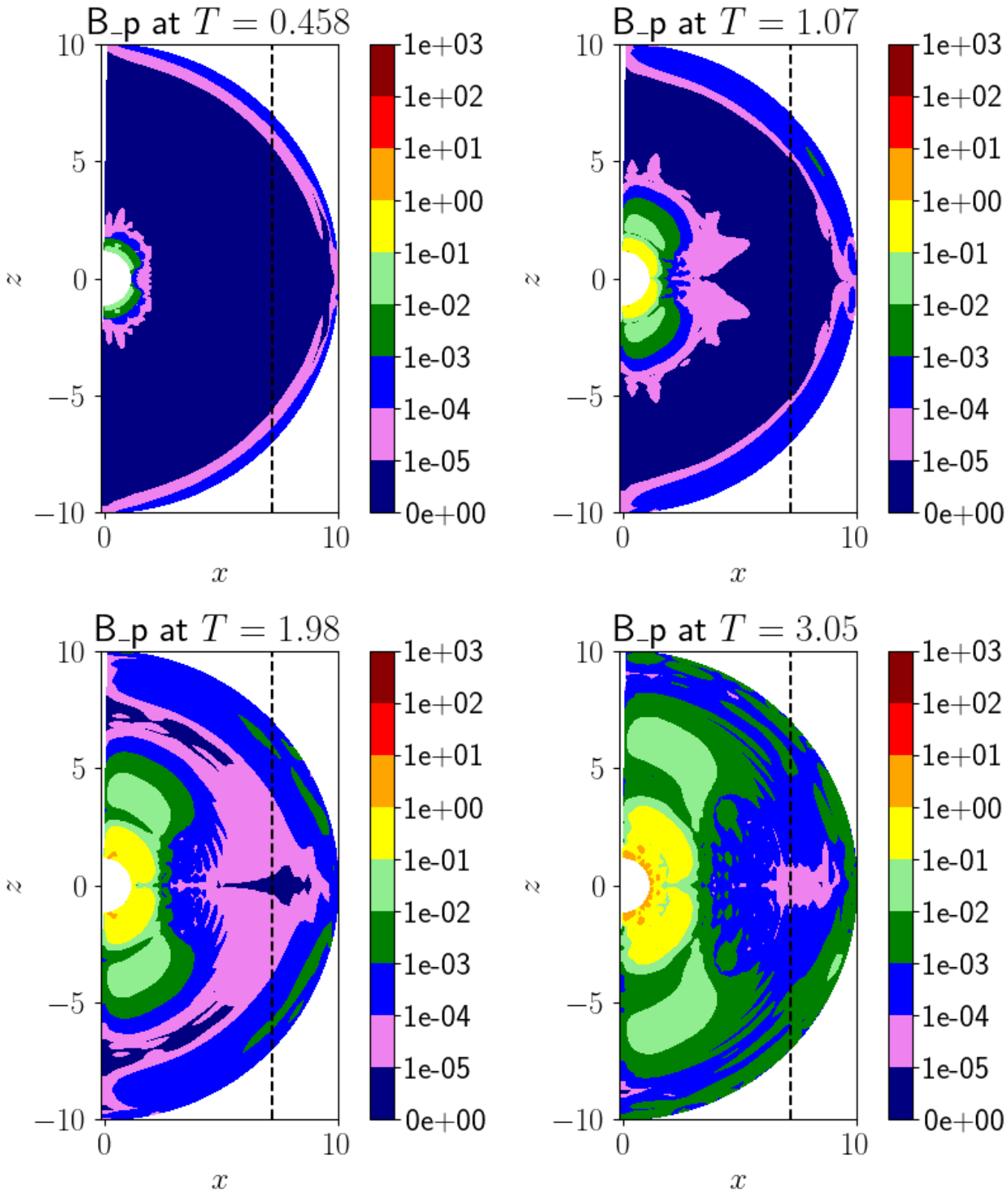

Figure 28.5: Contour plot of B_ϕ at various timepoints for the rotating model. The dashed black line represents the light cylinder.

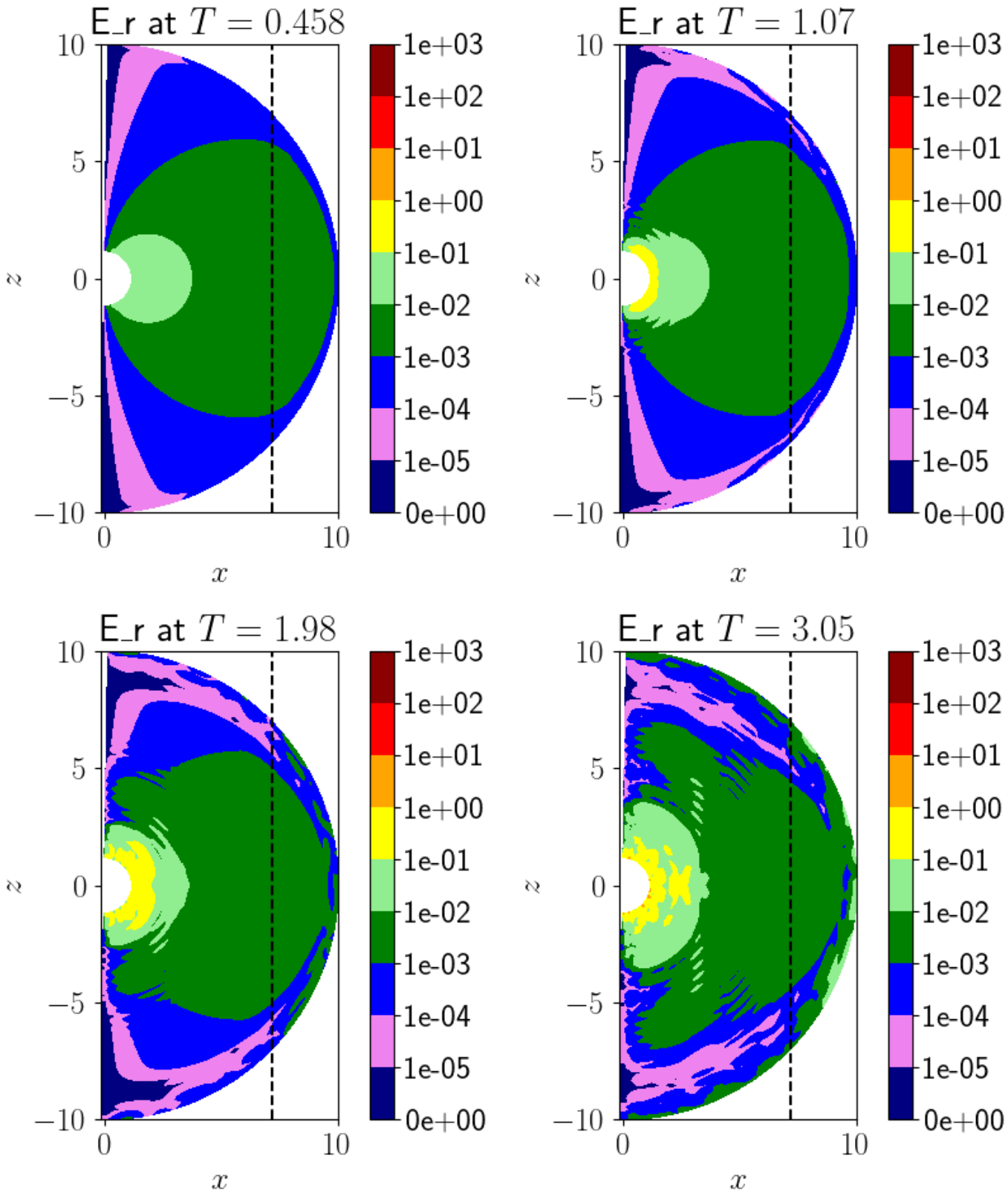

Figure 28.6: Contour plot of E_r at various timepoints for the rotating model. The dashed black line represents the light cylinder.

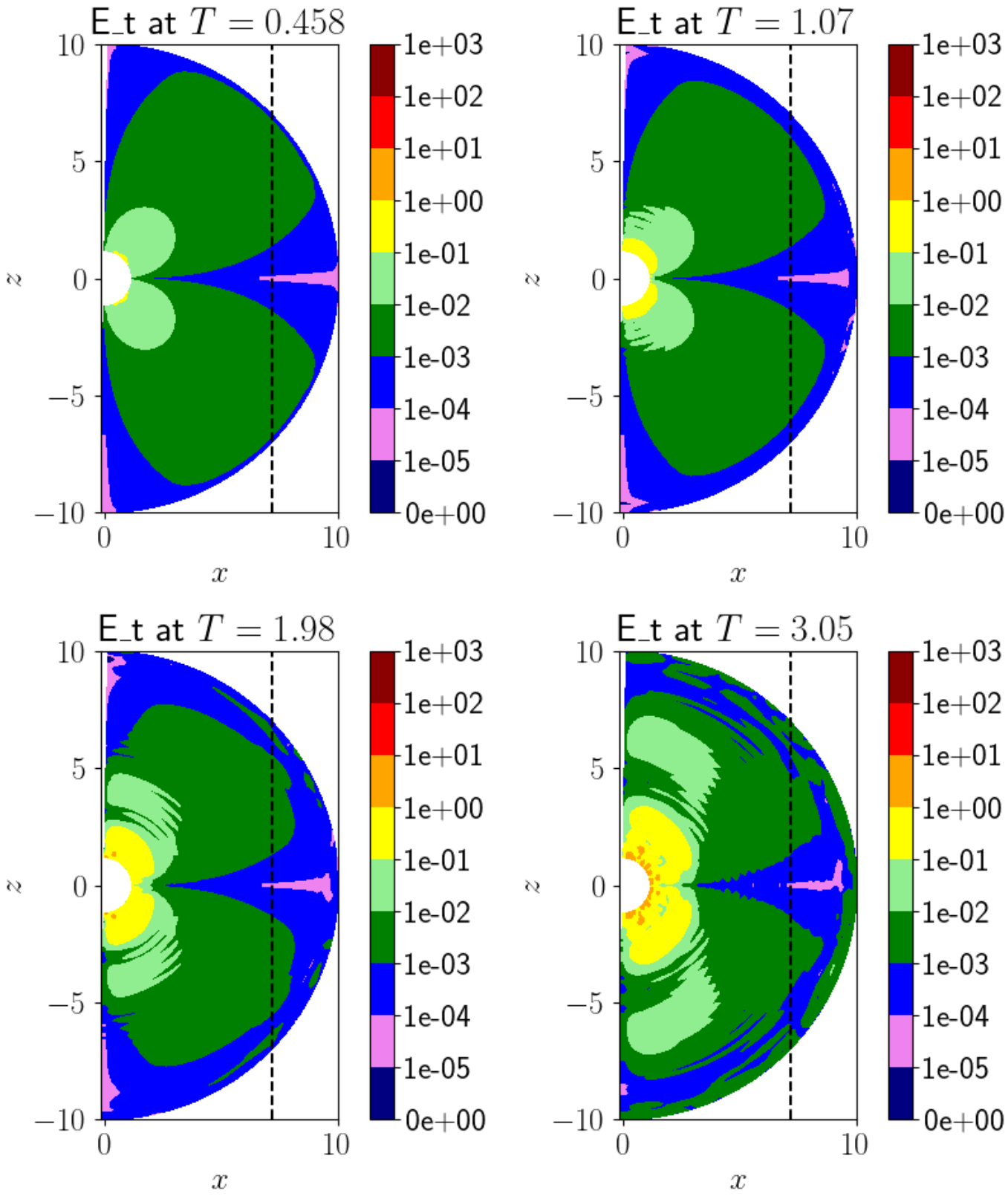

Figure 28.7: Contour plot of E_θ at various timepoints for the rotating model. The dashed black line represents the light cylinder.

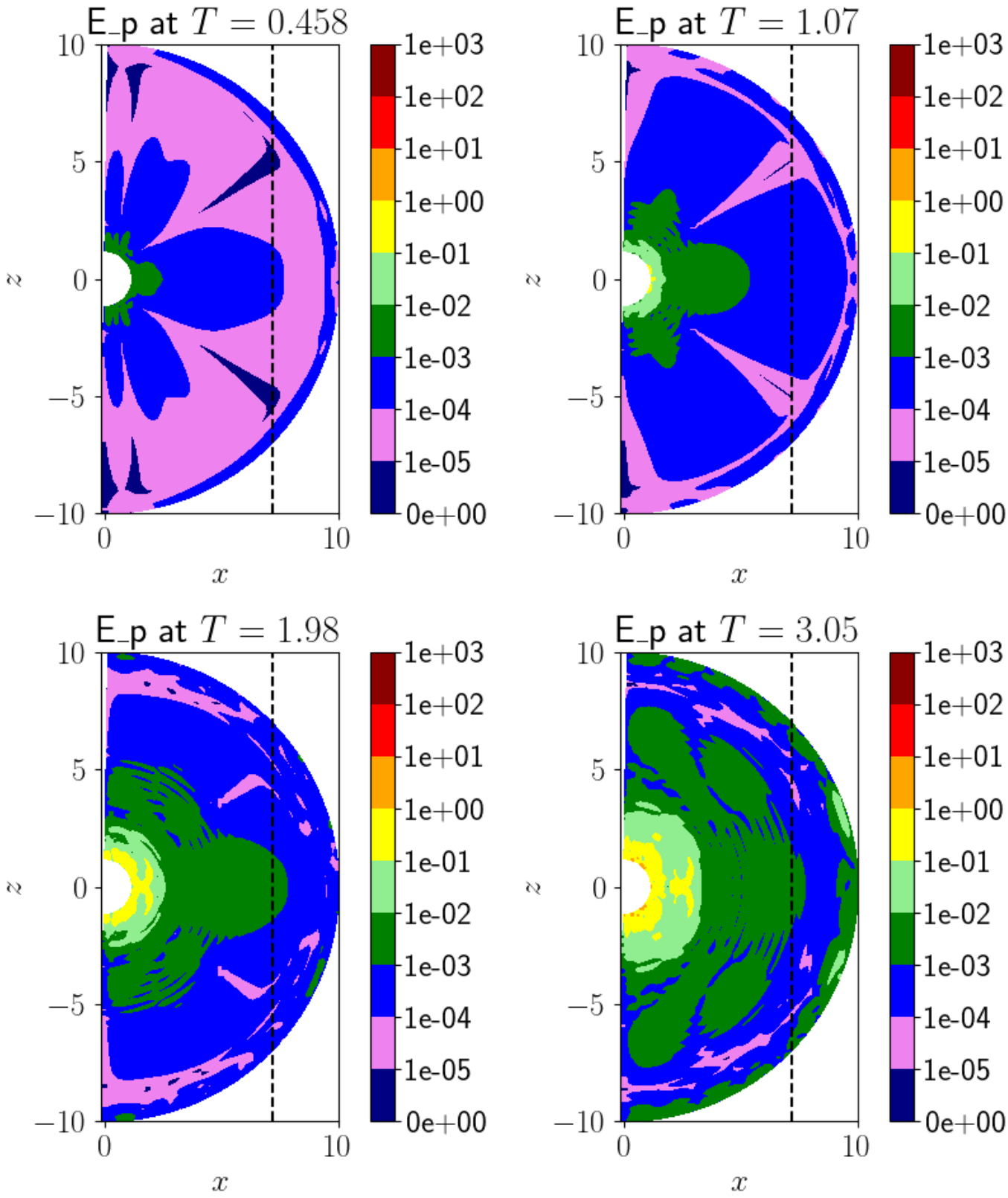

Figure 28.8: Contour plot of E_ϕ at various timepoints for the rotating model. The dashed black line represents the light cylinder.

Figure 28.9 shows the total energy as a function of time. As with the VSH decomposition, there is a sudden jump in the electric component just before the end of rampup, after which both the electric and magnetic components grow uncontrollably.

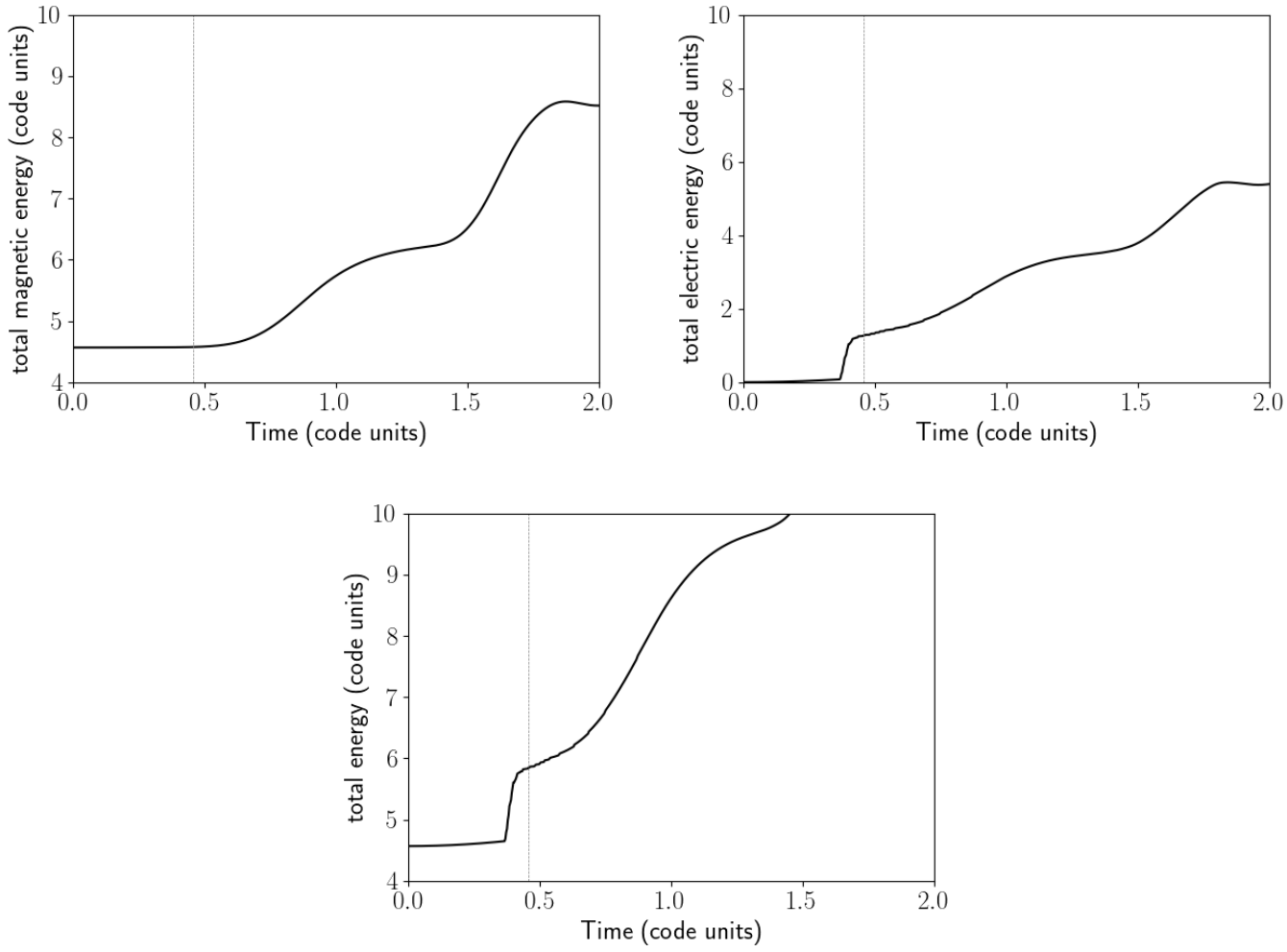

Figure 28.9: Total energy as a function of time for the rotating model.

The volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ (Figure 28.10) is not as close to zero as for the stationary dipole, nor does it remain at low values for as long: even by the end of ramp-up, the value has risen considerably from that at $t = 0$. However, we do not see the same pronounced rise over time as for the stationary dipole, indicating that the scheme is somewhat successful in maintaining the divergencelessness of $\mathbf{B}$.

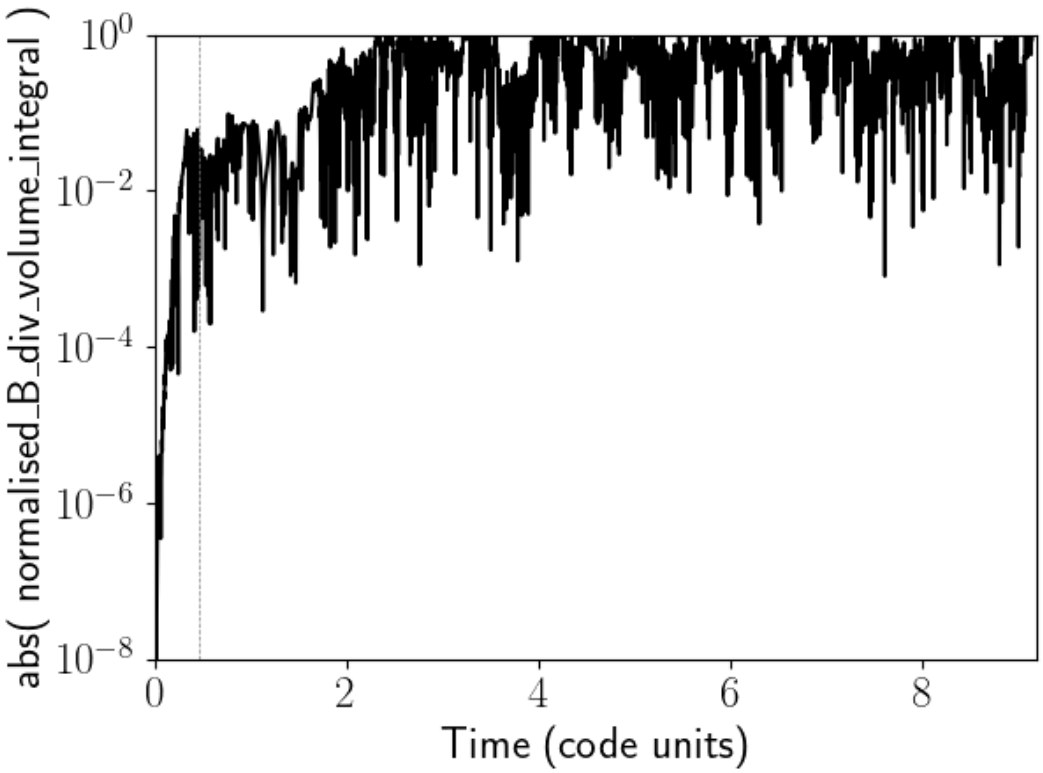

Figure 28.10: Normalised volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ across the domain as a function of time for the rotating model.

This relative consistency in $\boldsymbol{\nabla} \cdot \mathbf{B}$ compared to the stationary case may be due to our enforcement of the force-free conditions (§20.2 of this paper and §3.2 of Pétri (2012)). To test this, we plot in Figure 28.11 the percentage of gridpoints in the force-free region that are modified at each timestep to satisfy the force-free conditions. The value begins relatively low, indicating that the system initially obeys the force-free conditions quite well, and is stable until around 2 light-crossing times. After, the percentage of gridpoints begins to rise, indicating a growing instability. The rise exhibits some degree of oscillation, showing that the scheme employed is at least somewhat effective in delaying the breakdown of the force-free conditions, but as discussed in §20.2 we do not expect the breakdown to be prevented altogether.

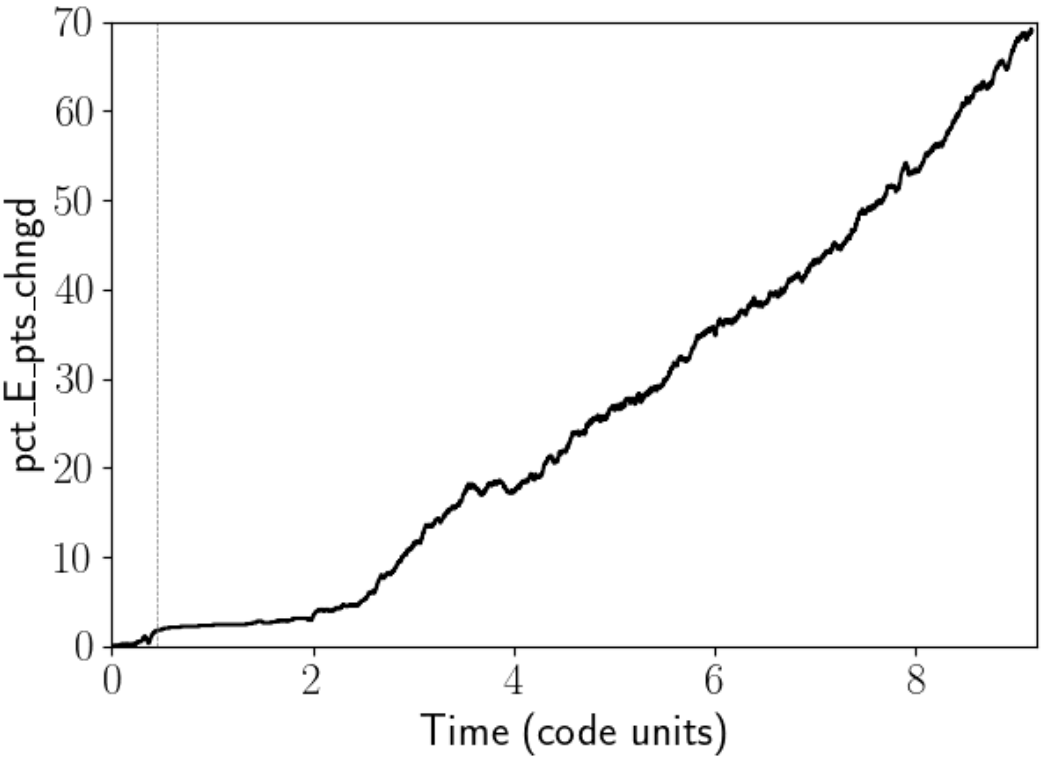

Figure 28.11: Percentage of gridpoints within the force-free region for which the force-free conditions must be enforced, as a function of time, for the rotating model.

Figure 28.12 shows the gridpoints at which the force-free conditions need to be applied at various points in the evolution. There is still a rotational symmetry about the equatorial line. The violating gridpoints appear not to persist to later times, indicating that our force-free condition enforcement is working. However, eventually the number of violating points escalates as the instability in the code grows, especially so near the inner and outer boundaries.

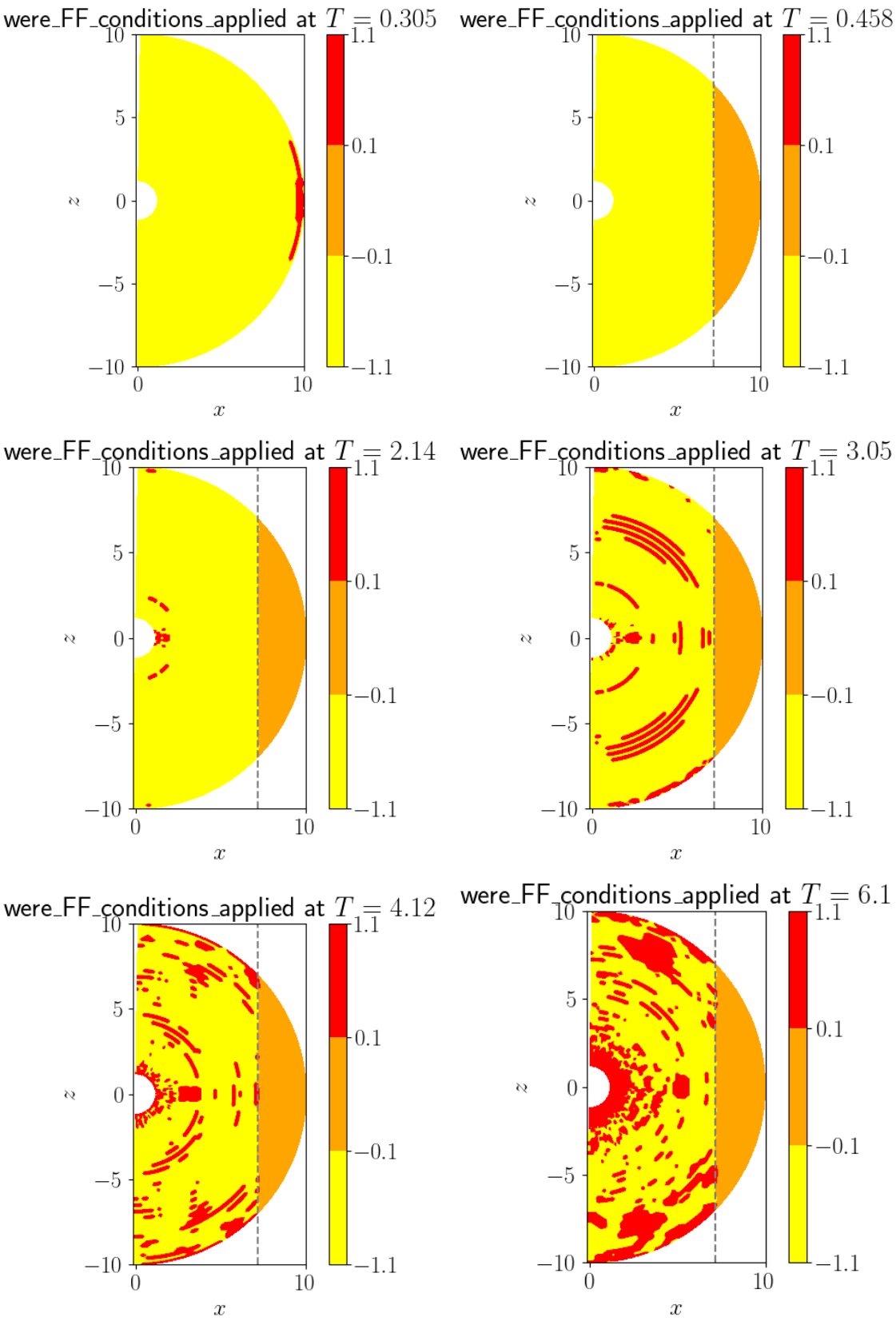

Figure 28.12: Gridpoints at which the force-free conditions are violated, and hence must be enforced, at various timepoints for the rotating dipole. *Red (+1)*: Gridpoint within FF region and FF conditions violated. *Yellow (-1)*: Gridpoint within FF region and FF conditions fulfilled. *Orange (0)*: Gridpoint outside FF region. The dashed grey vertical lines represent the light cylinder, which is still outside the domain at $T = 0.305$.

29 Magnetar model (constant magnetospheric twist)

Finally, we add a slight twist to the magnetosphere. Since we are most concerned with magnetars, which rotate slowly, we may approximate the star itself as non-rotating and only the twist as rotating. This may help to prevent issues from the previous test (Chapter 28) causing further numerical errors in the results.

We consider two configurations: first, a twist contained solely within the northern hemisphere $\theta \in [30°, 44°]$ in order to produce a rotational shear, preventing "cancelling-out" of its effect above and below the equatorial line. Second, we consider a twist symmetric about the equatorial line $\theta \in [30°, \pi - 30°]$ in order to allow comparisons with previous studies such as those by Contopoulos et al. (1999). We use the following parameters:

```
P_final                      : 44.9688687          0.0015
Omega_final                  : 0                   0
r_min, r_max                 : 1                   10.0020751953125
n_r, delta_r                 : 50                  0.1837158203125
n_t, delta_t                 : 101                 0.0314159265358979
Timestep delta_T             : 0.0030517578125     1.01795683349045e-07
Length of simulation         : 9                   0.000300207685678337 steps: 3000
ell_max                      : 20
Ramp start                   : 100001 305.178833007812
Ramp stop                    : 100150 305.633544921875

Twist start                : 1 0.0030517578125
Twist stop                   : 150 0.457763671875
Twist magnitude              : 0.15
Twist min radius             : 1
Twist ramp    99% radius     : 1.9
Twist de-ramp 99% radius     : 9.1
Twist max radius             : 10
Twist min angle (rad)        : 0.523598775598299
Twist ramp    99% angle (rad): 0.593411945678072

use_outer_sponge_layer: 1
sigma_0, gamma, beta    : 0.8 6 4
```

For the northern-hemisphere twist,

```
Twist de-ramp 99% angle (rad): 0.698131700797732
Twist max angle (rad)        : 0.767944870877505
```

For the symmetric twist,

```
Twist de-ramp 99% angle (rad): 2.54818070791172
Twist max angle (rad)        : 2.61799387799149
```

The twist angular velocity $\Omega_{\text{twist}} = 0.15$ was chosen to be comparable with the rotation rate of our rotating dipole (Chapter 28). The twist angles were chosen so that the gridpoint at which we evaluate the fields falls roughly midway into the twisted region and we can analyse the effects of the twist in full. We will directly compare this evolution with the stationary dipole (Chapter 27).

Figure 29.1 shows the field components evaluated at the same gridpoint as the previous evolutions, which indeed falls within the twisted region. Both twists have qualitatively similar behaviour, driving rises in $|B_r|, |E_r|, |E_\theta|, |B_\phi|, |E_\phi|$ which the code appears to work to restabilise before overshooting and losing out to numerical noise. The model with a symmetric twist survives for longer, perhaps because the large twisted region makes the configuration similar to our fully rotating model, which was also relatively stable (Chapter 28).

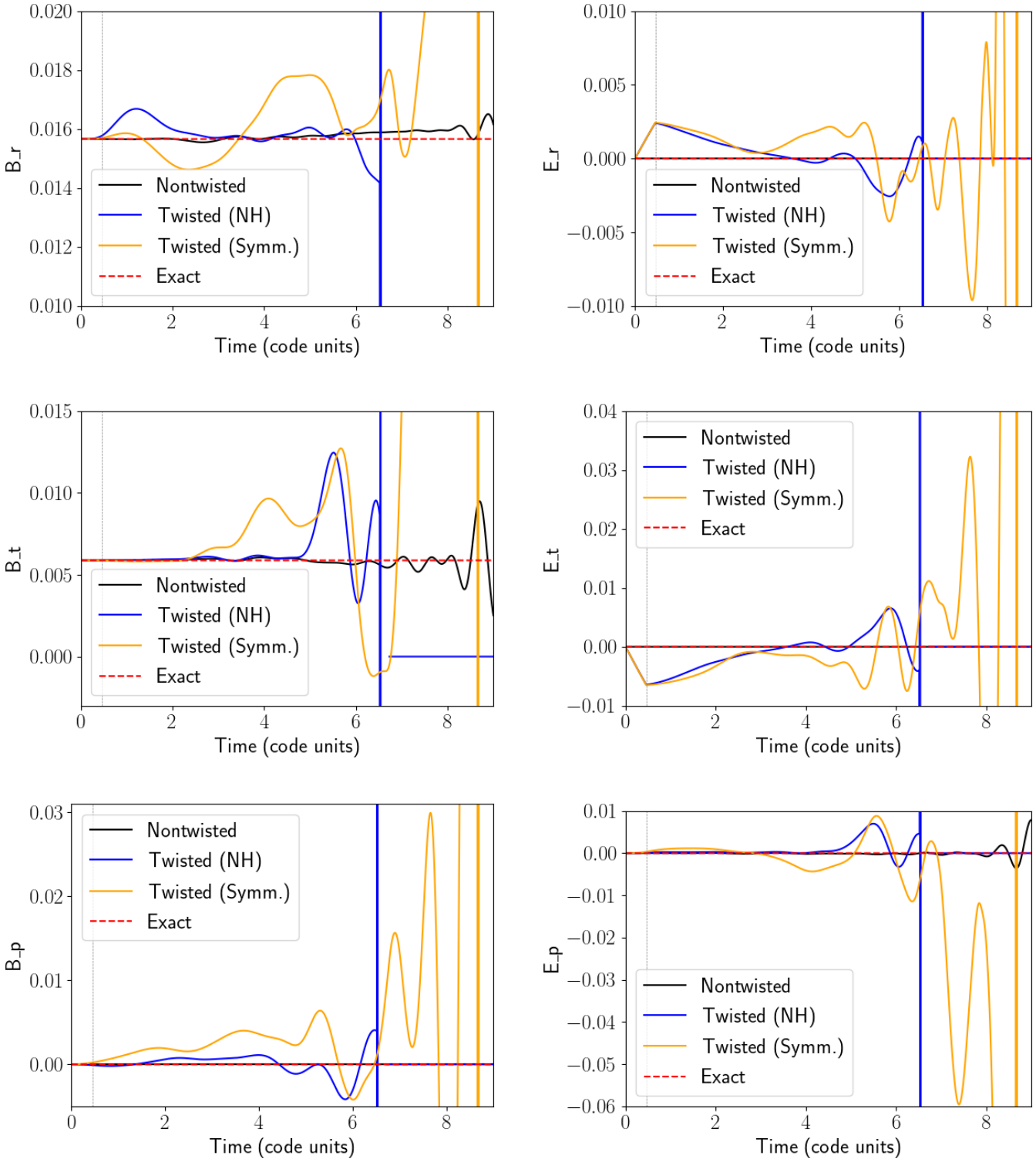

Figure 29.1: Time-evolution of field components at a single gridpoint for the simulated twisted dipoles. "NH" denotes the twist localised within the northern hemisphere; "Symm." denotes the twist symmetric about the equatorial line. The dashed grey vertical lines represent the end of the twist ramp-up.

Figure 29.2 shows that standard deviation of the field components. All models are similar in B_r, B_θ. The twists introduce nonzero E_r, E_θ, E_ϕ, but note that the standard deviation in the electric field components remains roughly stable until around 4 light-crossing times for both twists.

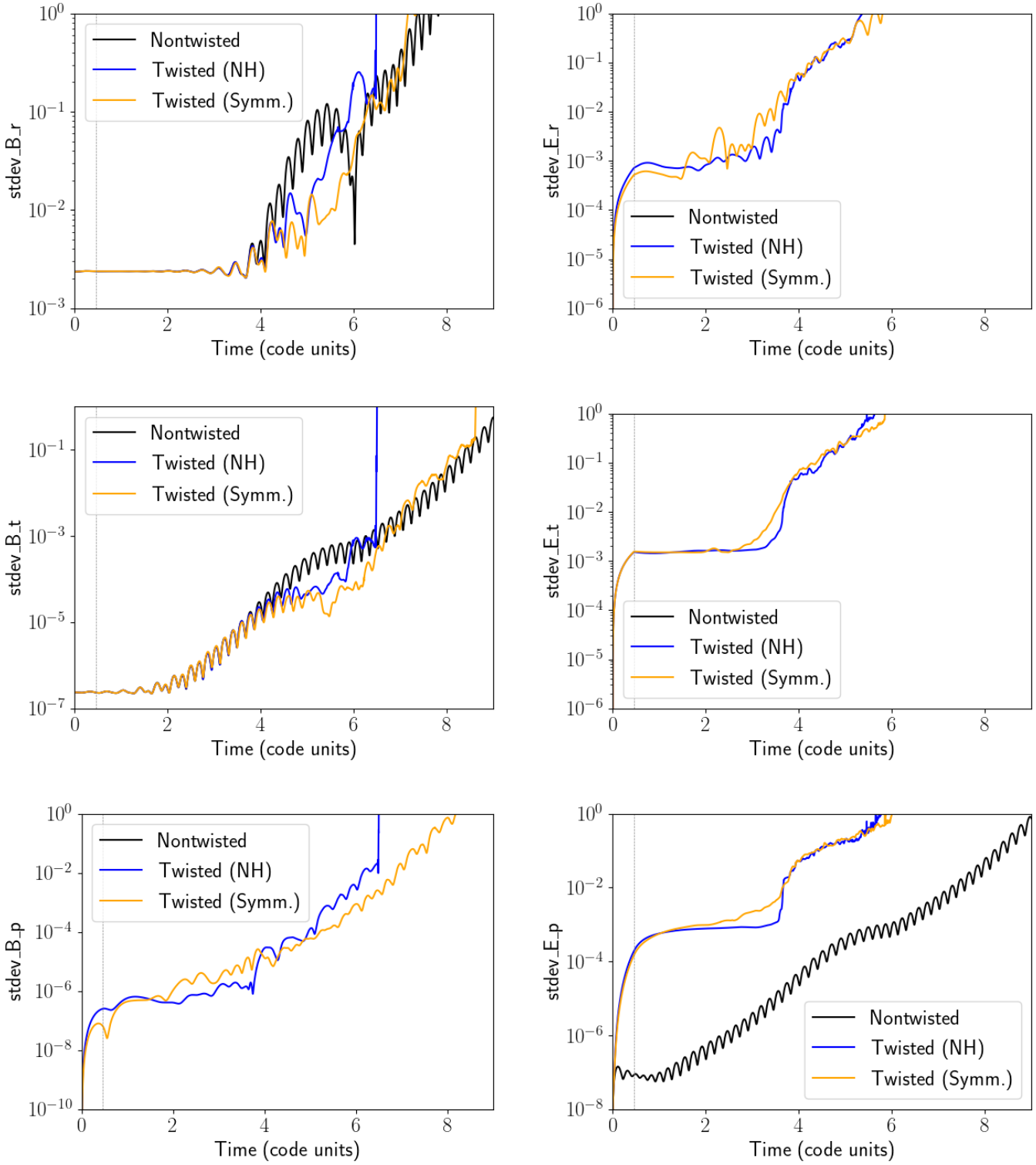

Figure 29.2: Time-evolution of standard deviation of VSH decomposition of field components at a single grid-point for the simulated twisted dipoles. "NH" denotes the twist localised within the northern hemisphere; "Symm." denotes the twist symmetric about the equatorial line. The dashed grey vertical lines represent the end of the twist ramp-up.

Figures 29.3 to 29.8 show heatmaps of the vector components at the same timesteps as for the stationary dipole (Figures 27.3 to 27.5) for the northern-hemisphere twist. The influence of the twist on B_r, B_θ is initially quite subtle; the model diverges at later times but numerical error is beginning to dominate by then. There is a significant region of nonzero B_ϕ coinciding with the twist, which begins to proliferate throughout the entire domain. The twist also induces an electric field, which also spreads throughout the domain as evolution progresses. In both the graphs of E_r and E_θ there is a noticeable decrease in the intensity within the twisted region between $T = 0.458$ and $T = 1.07$, suggesting that the code is attempting to return to a stable configuration, but it cannot abate the spread of nonzero E_r, E_θ to the rest of the domain.

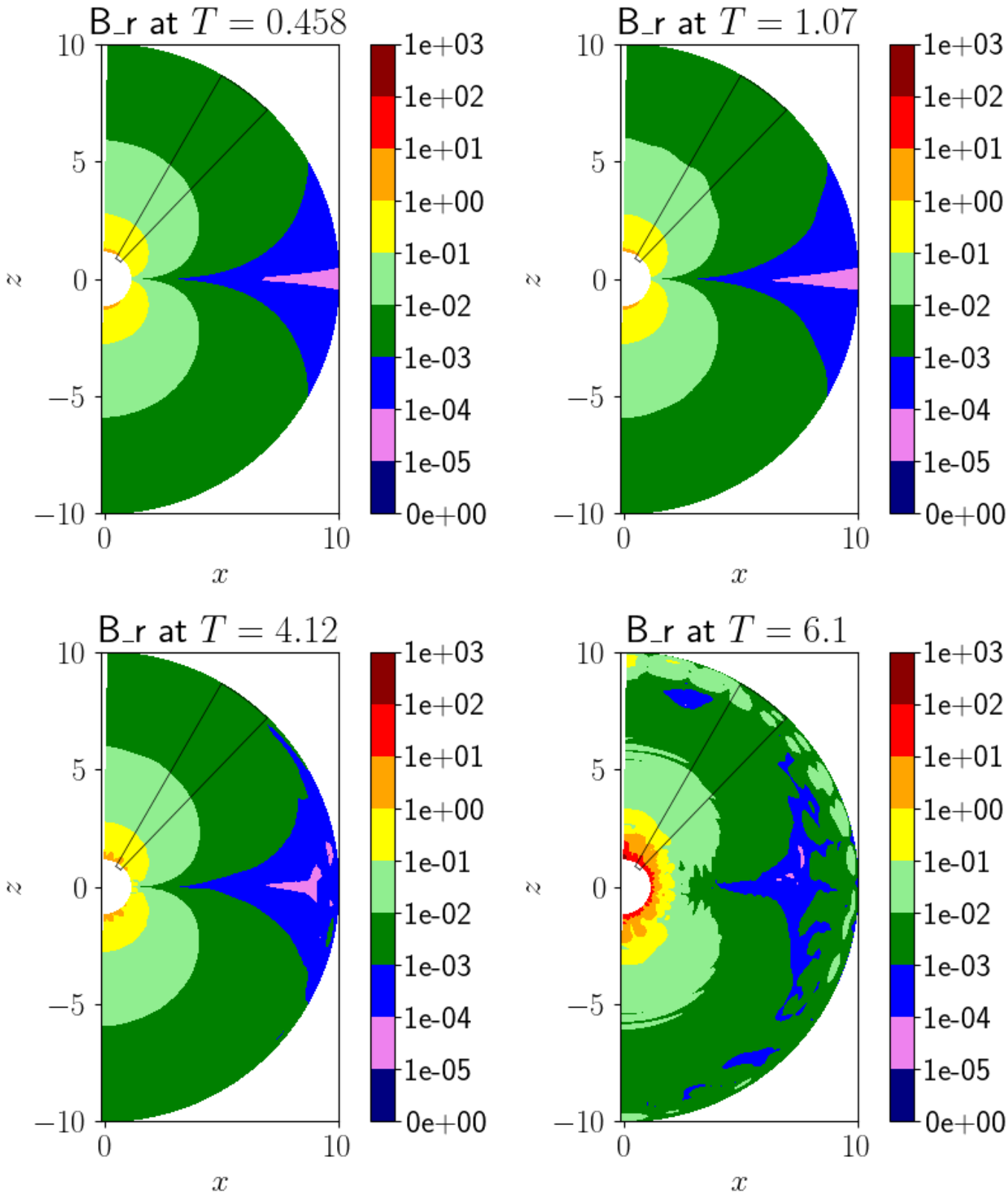

Figure 29.3: Contour plot of B_r at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

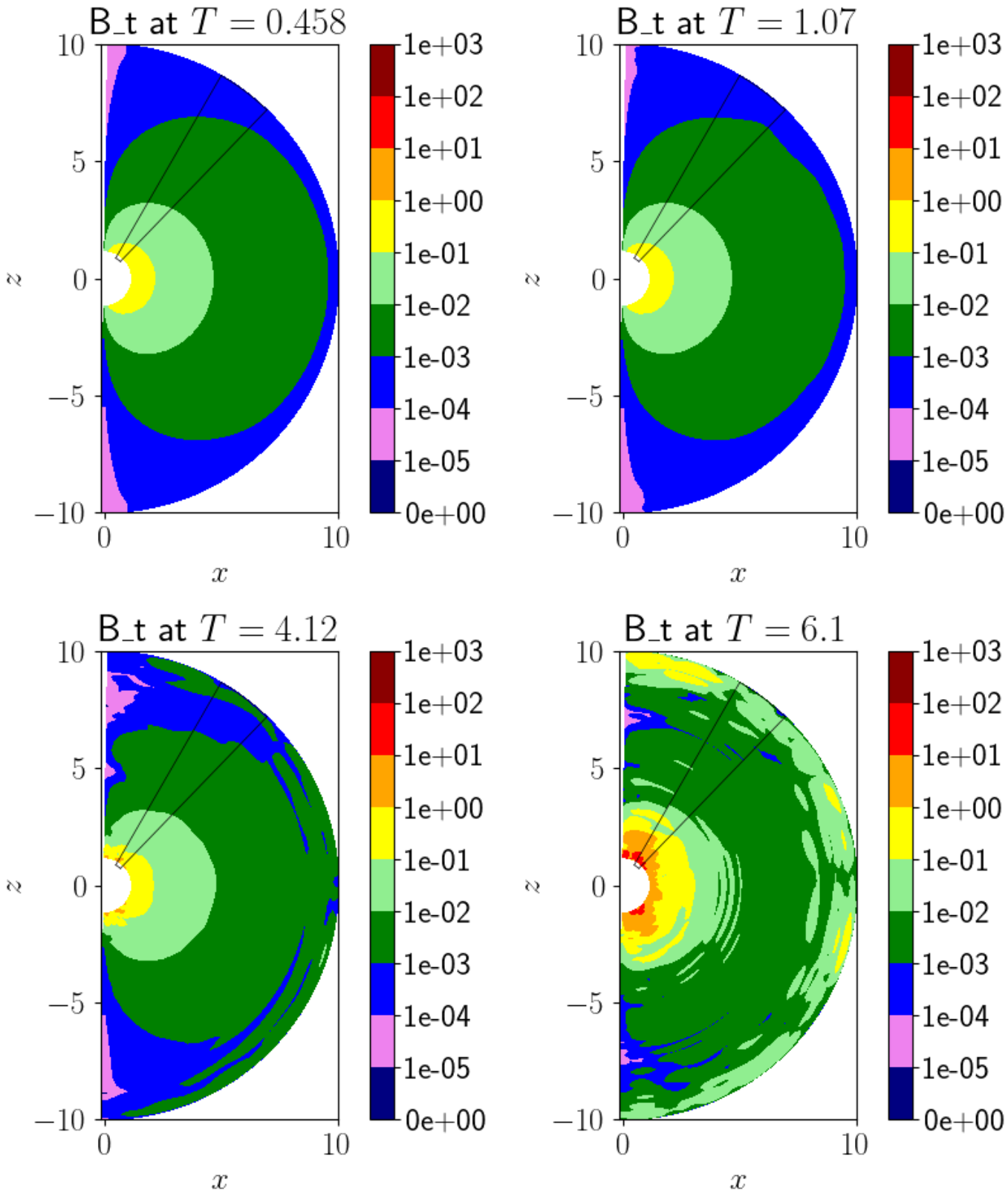

Figure 29.4: Contour plot of B_θ at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

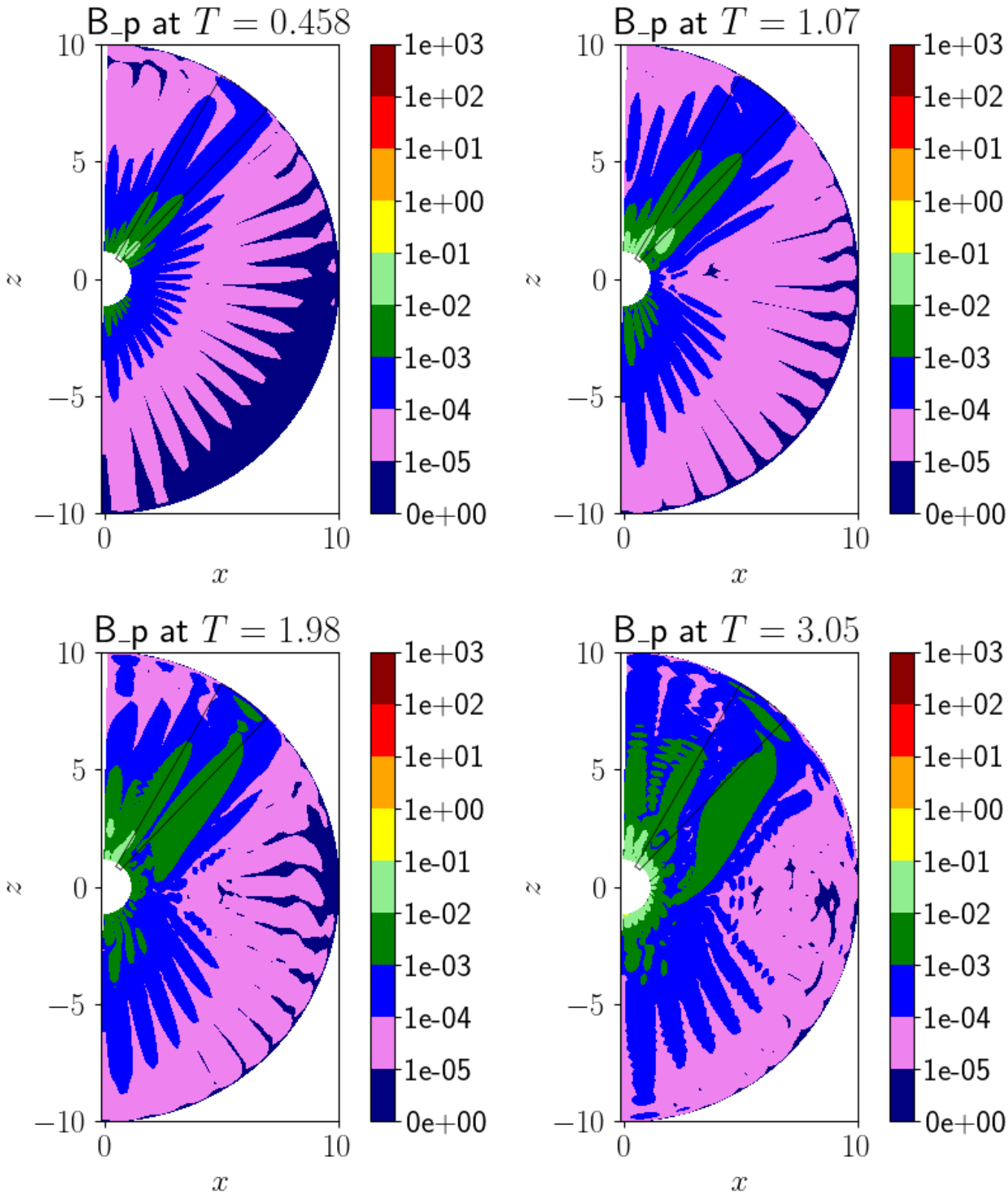

Figure 29.5: Contour plot of B_ϕ at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

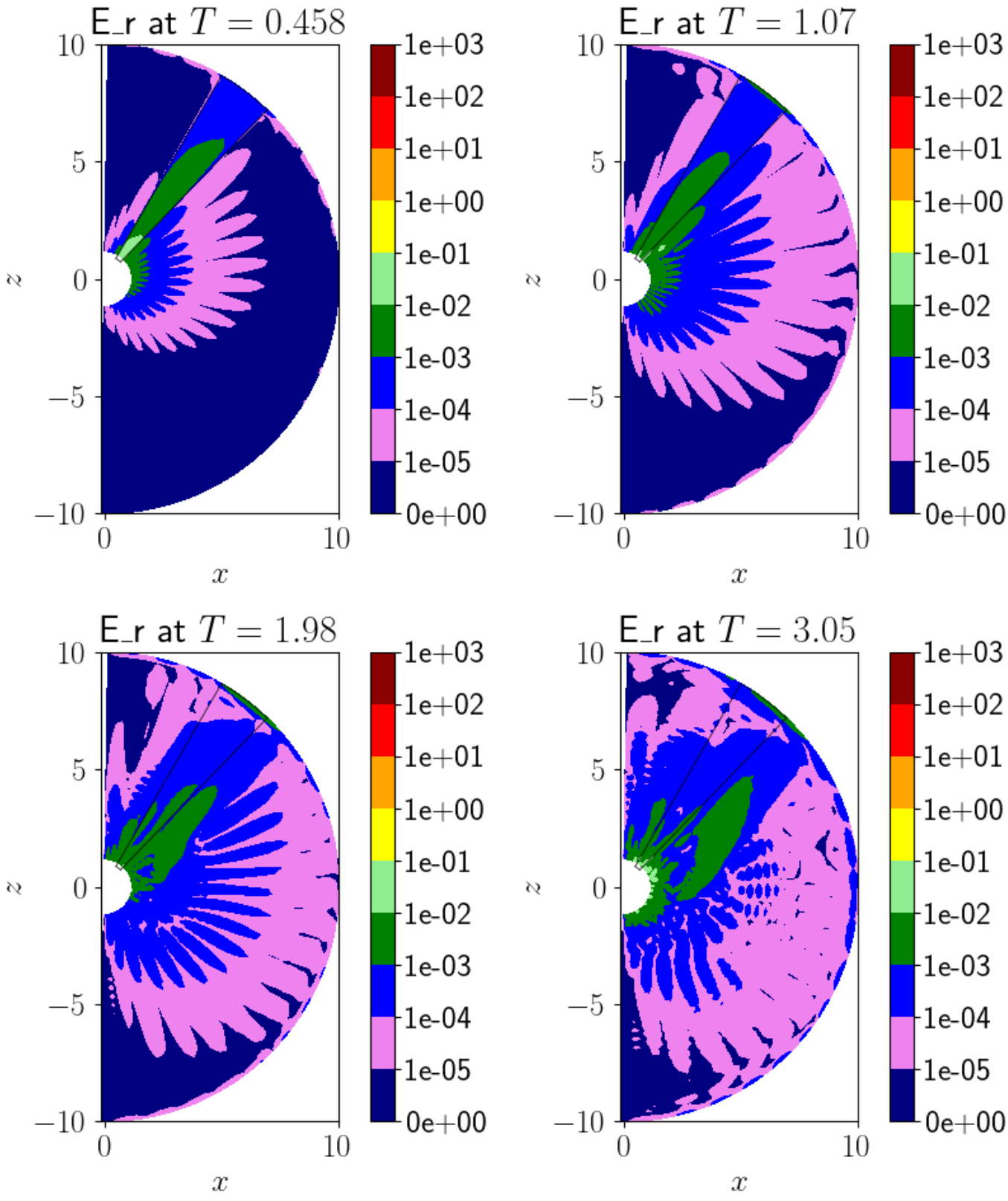

Figure 29.6: Contour plot of E_r at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

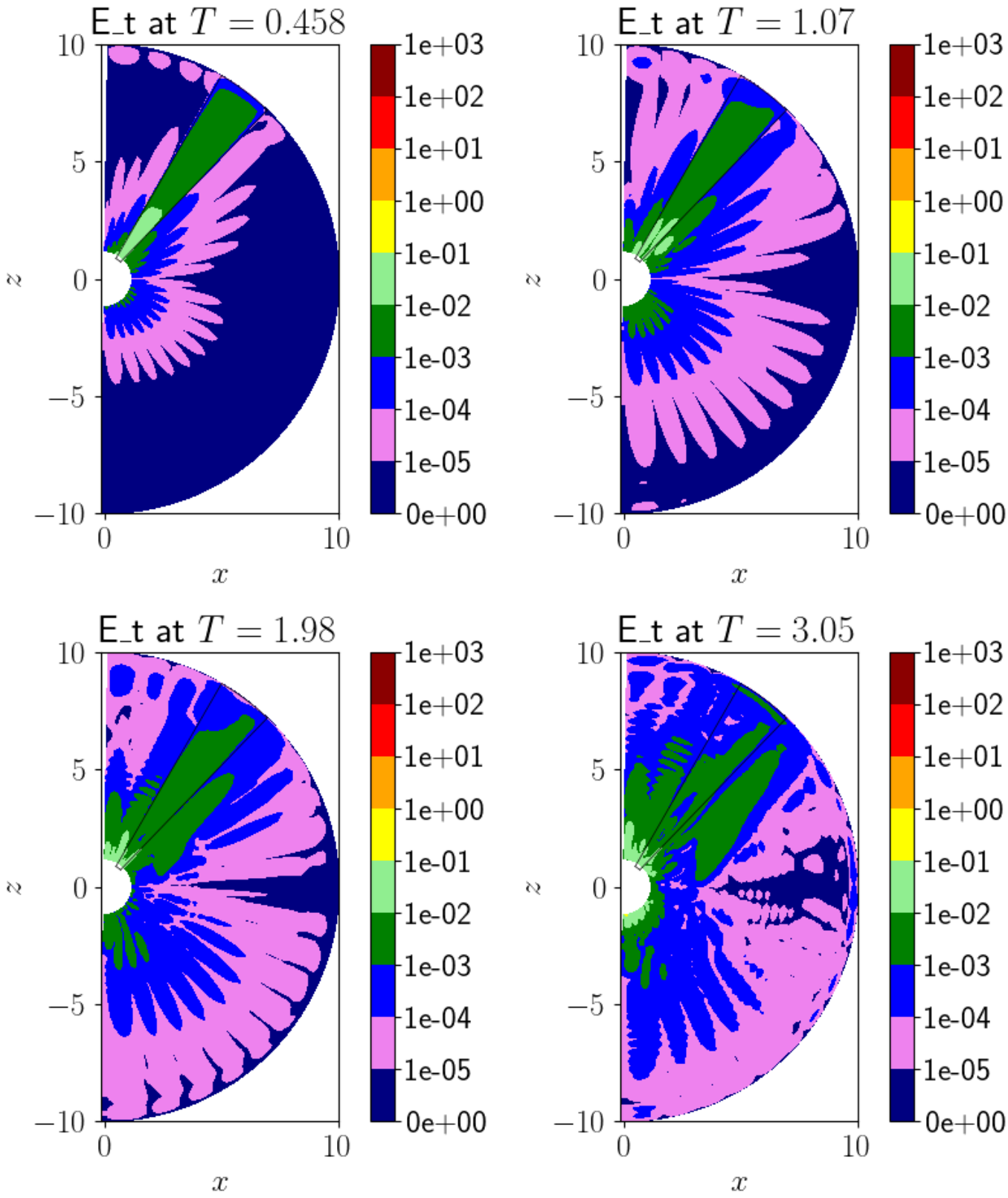

Figure 29.7: Contour plot of E_θ at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

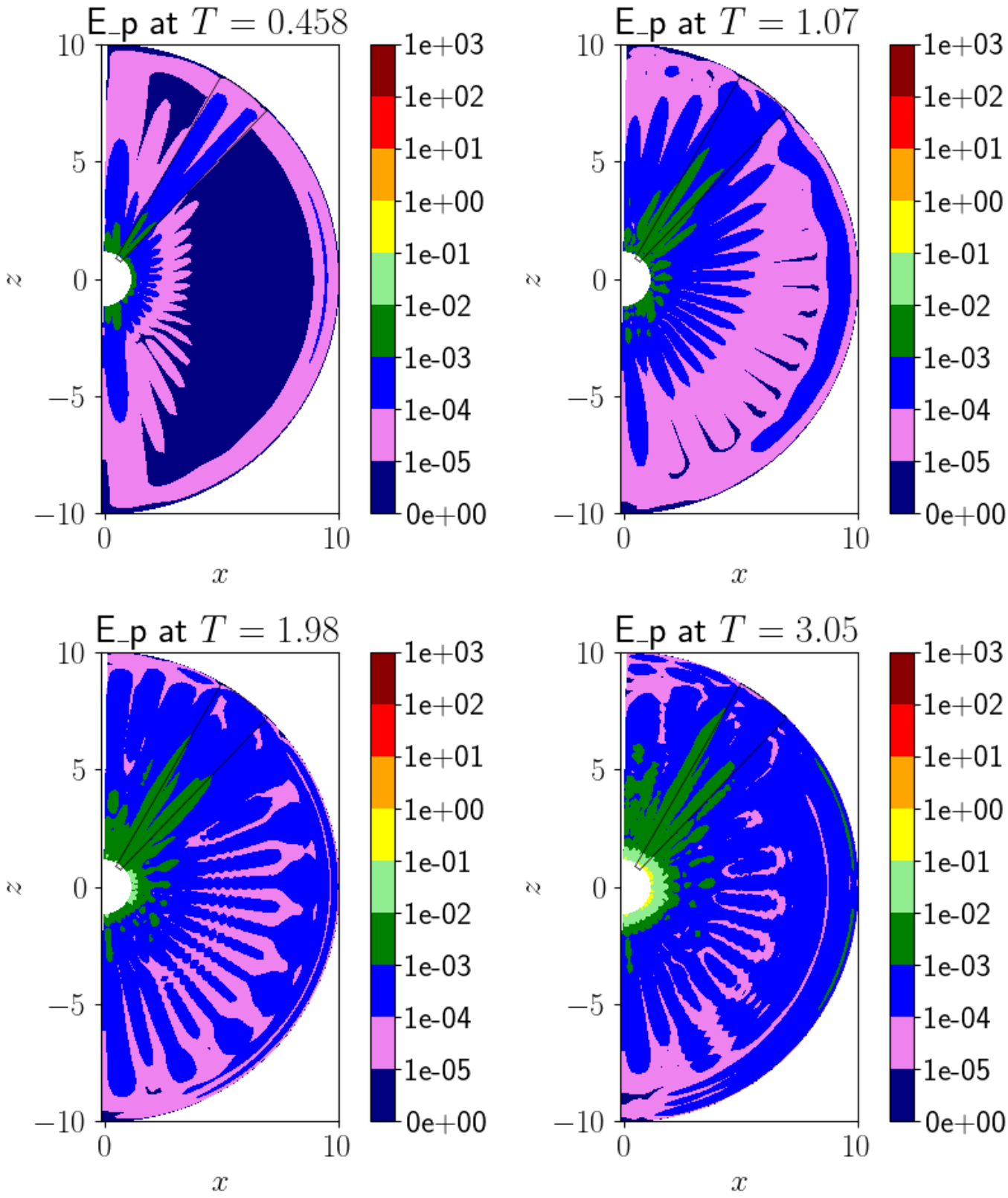

Figure 29.8: Contour plot of E_ϕ at various timepoints for the model with a twist localised in the northern hemisphere. The black box represents the twisted region.

Figures 29.9 to 29.14 show the corresponding heatmaps for the symmetric twist. The behaviour is similar to before. The nonzero electric field within the twisted region is qualitatively similar to that of the rotating dipole (Figures 28.6 to 28.8), perhaps explaining why the configuration remains relatively stable. Note that the domain only shows small deviations from equatorial symmetric even at late times.

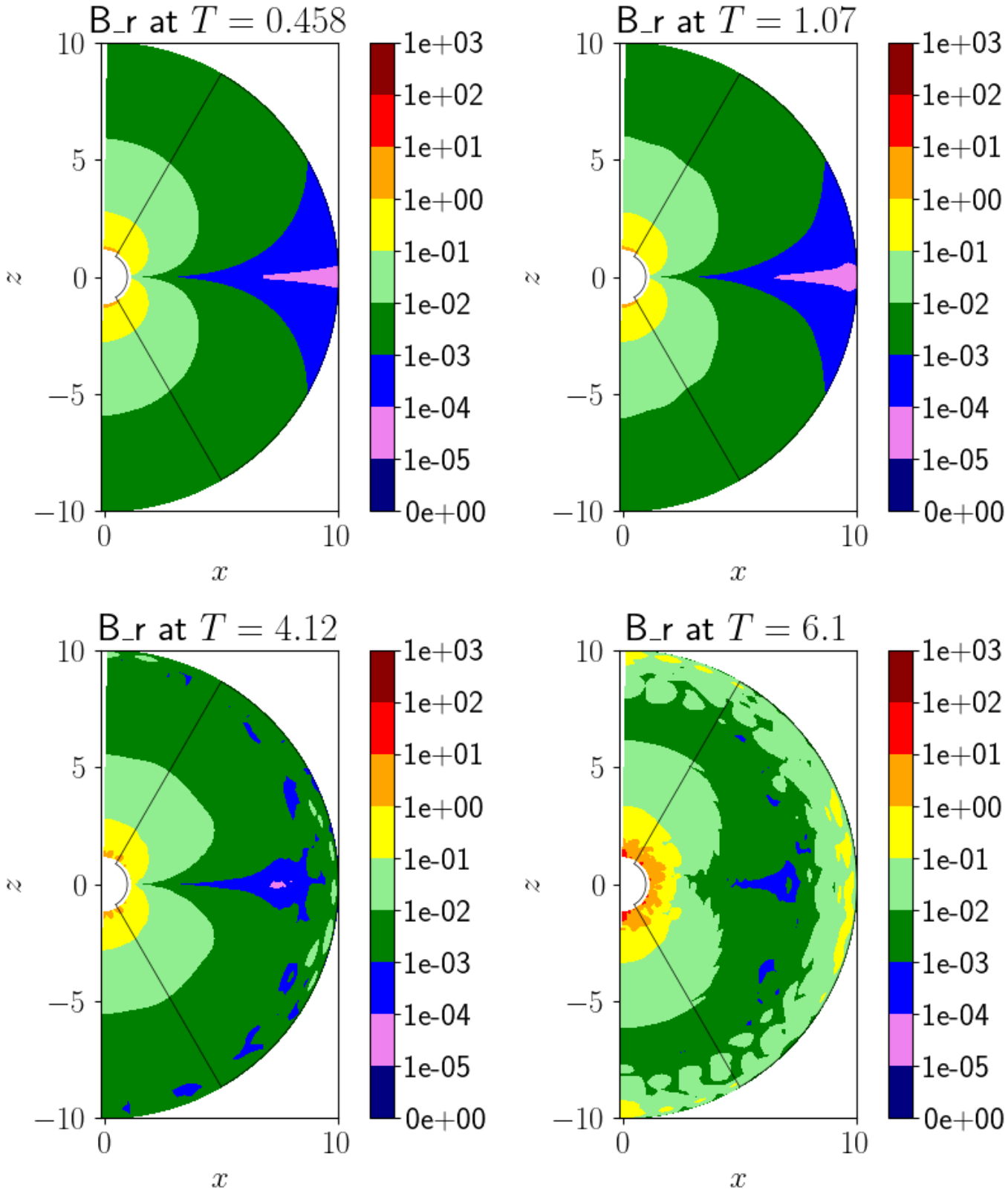

Figure 29.9: Contour plot of B_r at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

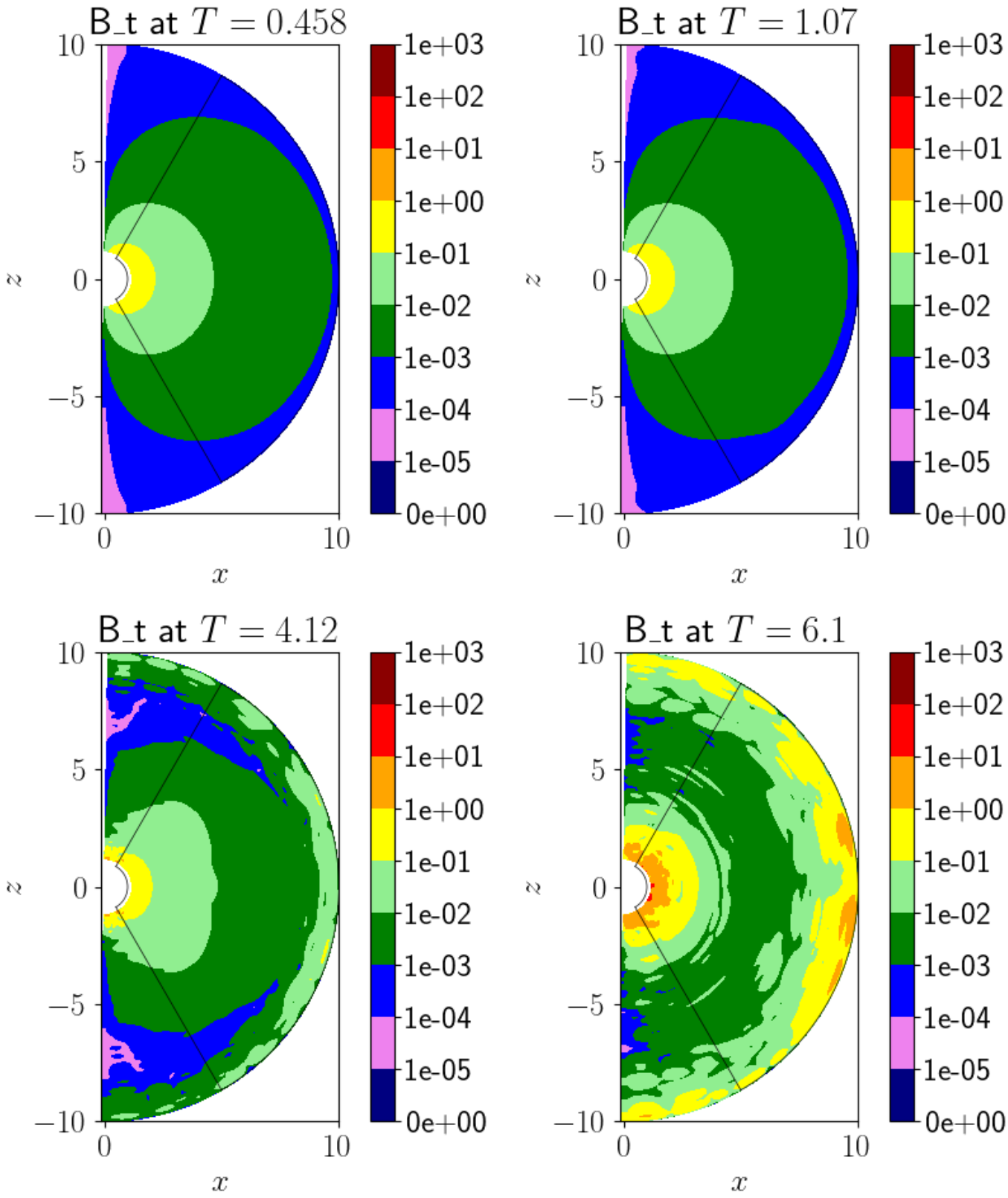

Figure 29.10: Contour plot of B_θ at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

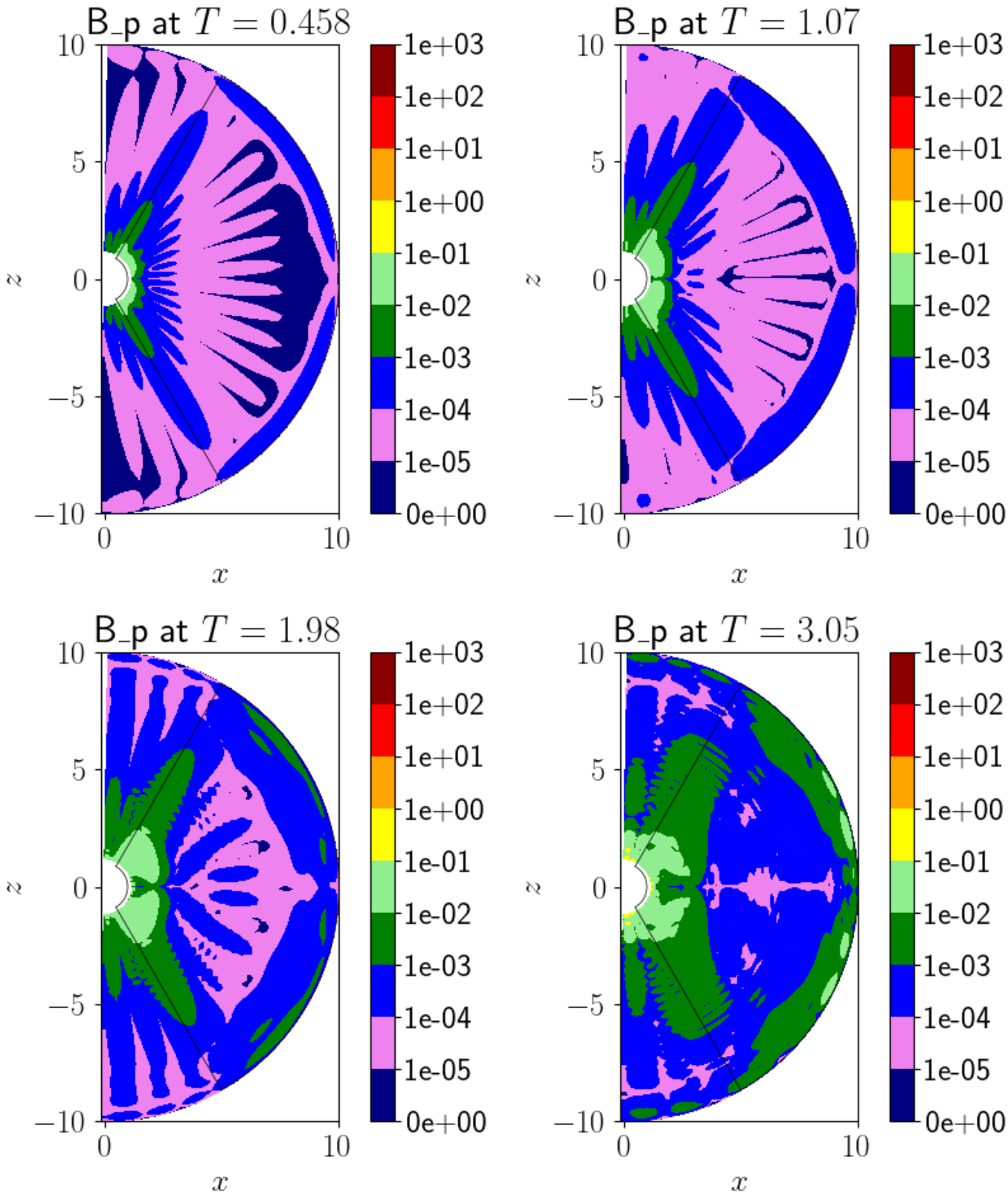

Figure 29.11: Contour plot of B_ϕ at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

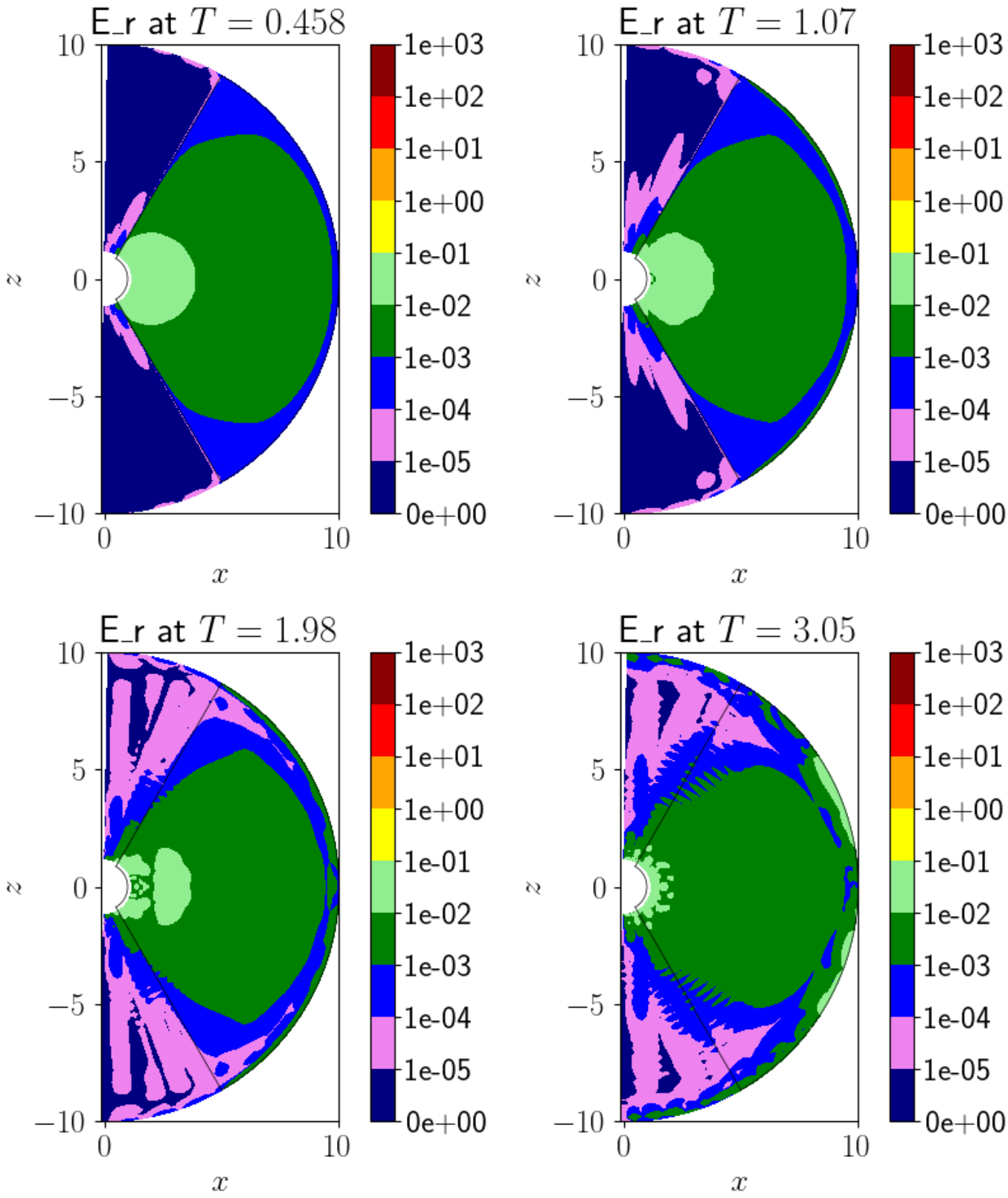

Figure 29.12: Contour plot of E_r at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

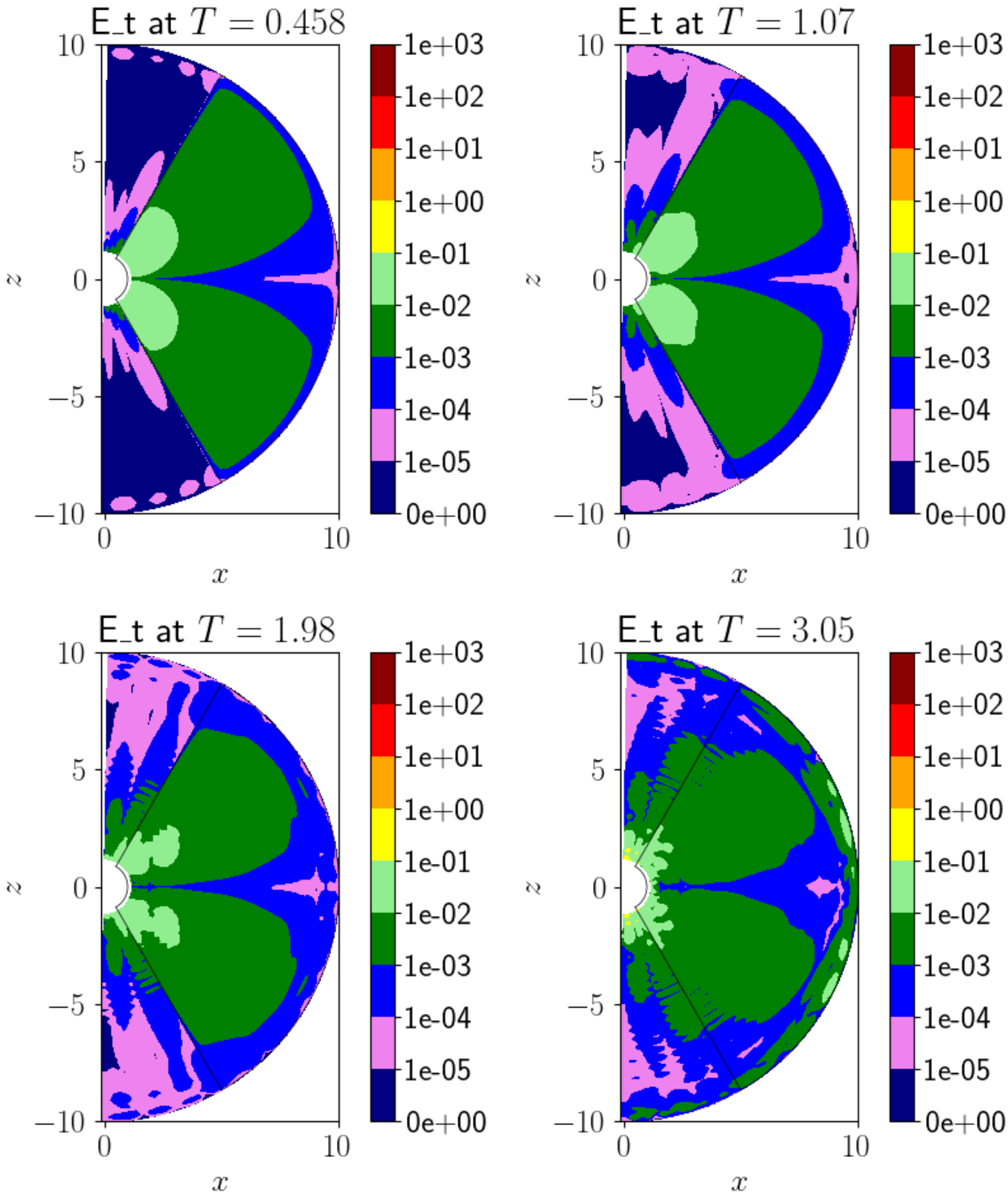

Figure 29.13: Contour plot of E_θ at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

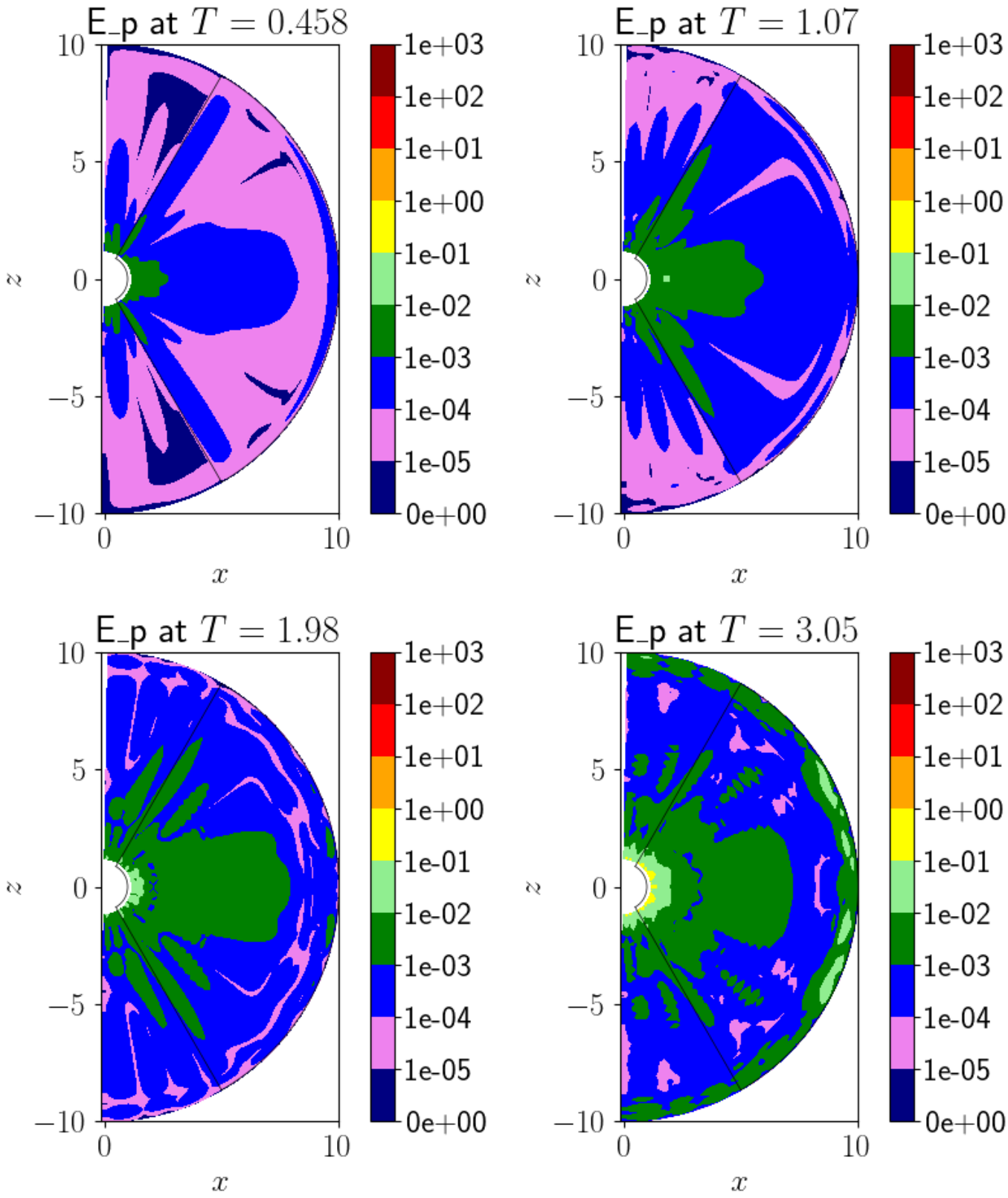

Figure 29.14: Contour plot of E_ϕ at various timepoints for the model with a twist symmetric about the equatorial line. The black box represents the twisted region.

Figure 29.15 shows the total energy for both configurations. That of the twist in the northern hemisphere deviates little from the nonrotating model, differing by just 3% after 1350 timesteps ($T \approx 4.12$). That of the symmetric twist deviates significantly from the nonrotating model, and underestimates the rotating model by an amount that grows with time.

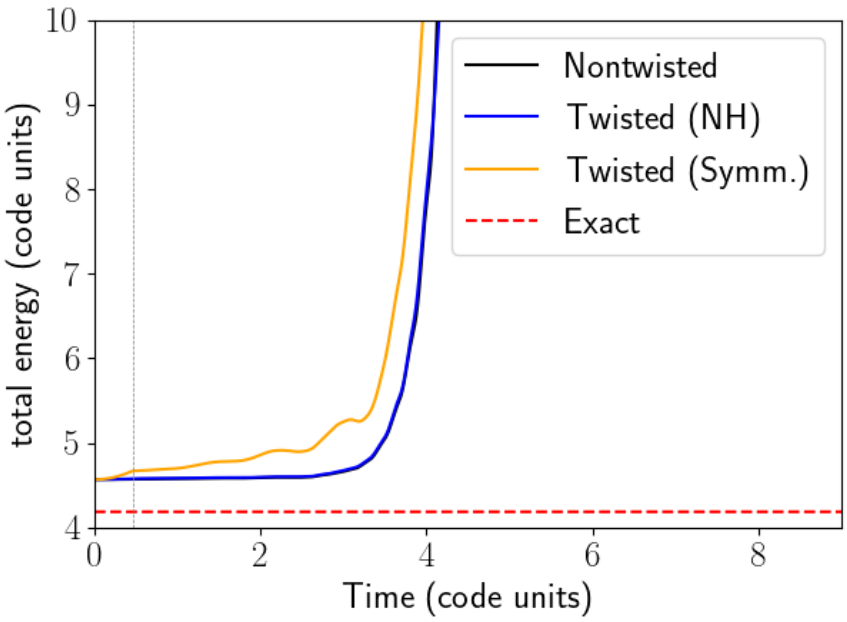

Figure 29.15: Total energy as a function of time for the twisted models. "NH" denotes the twist localised within the northern hemisphere; "Symm." denotes the twist symmetric about the equatorial line.

Figure 29.16 shows the volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$. Both twisted models have a significantly higher value than the nonrotating model, but still appreciably close to zero until around 4 light-crossing times.

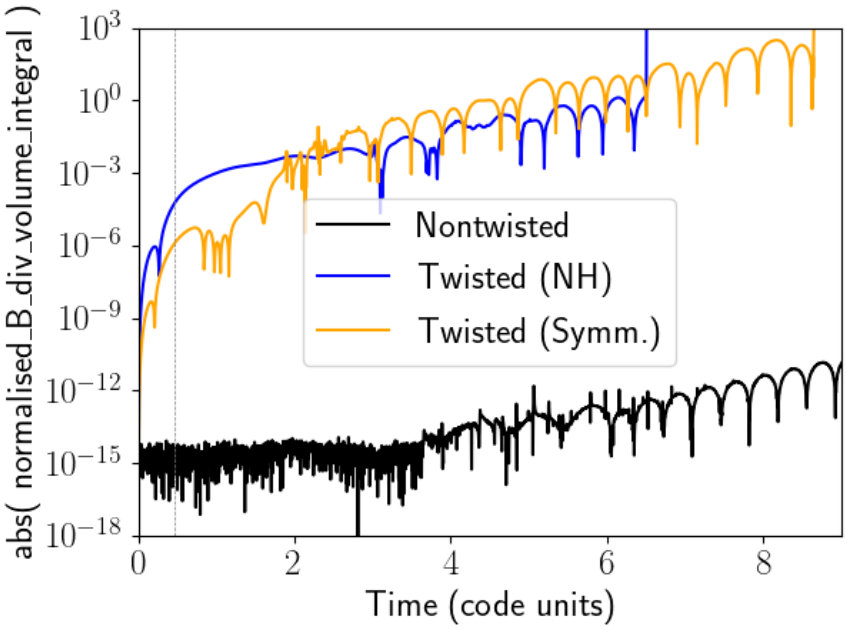

Figure 29.16: Normalised volume integral of $\boldsymbol{\nabla} \cdot \mathbf{B}$ across the domain as a function of time for the twisted models. "NH" denotes the twist localised within the northern hemisphere; "Symm." denotes the twist symmetric about the equatorial line.

30 Summary

30.1 Introduction

Neutron stars are the remnants of stars who began their life at $M_{\text{ZAMS}} \approx 10 - 25\, M_\odot$[1] (e.g. Heger et al., 2003, Figure 4). They cool extremely rapidly due to neutrino emission by the Urca process (Gamow & Schoenberg, 1941). The resulting low luminosity, coupled with their small size, helps explain why three decades passed between their theorisation Baade & Zwicky (1934a,b) and eventual detection as radio pulsars in late 1967 (Hewish et al., 1968). Neutron stars are detectable due to emission mechanisms in their **magnetospheres**, their surrounding regions which are populated by charged particles. Isolated neutron stars are divided into two main classes:

1. **Pulsars** emit beams of radio waves and X-rays from their poles that sweep across the sky as the star rotates, and are detected from Earth as regular pulses of radiation.

2. **Magnetars** do not emit strong beams, but their extreme internal magnetic fields drive transient X-ray and gamma-ray emission from the magnetosphere including violent bursts.

The aim of this thesis is to perform numerical simulations in order to better understand the structure of neutron star magnetospheres.

We will approximate a pulsar by a magnetic dipole rotating at a constant rate, following the **Goldreich-Julian model** (Goldreich & Julian, 1969) that has become standard (Chapter 2). Magnetars rotate so slowly (Chapter 6) that they can be considered stationary relative to the motion of any twisted field lines in their magnetospheres. Further, we tested the stability of our evolution code for both stationary (Chapter 27) and rotating (Chapter 28) dipoles, finding that the stationary configuration remains stable for longer. Then, we may expect the best numerical results if we approximate the magnetar as stationary as opposed to rotating at its slow rate.

However, the Goldreich-Julian model appears inconsistent with models of magnetars where their strong magnetic fields occur by amplification due to convection (§7.1). Further, Contopoulos et al. (2024) have recently explored alternative equilibrium configurations for pulsar magnetospheres, suggesting that a simple dipole is not the only possible stable configuration.

Neutron stars are often assumed to obey **force-free electrodynamics (FFE)** (§20.2), where all non-magnetic forces are negligible and so charged particles are restricted to flowing along magnetic field lines. The magnetic field lines of a non-rotating dipole are "closed" in that they originate at one pole and terminate at the other. For a neutron star, which has finite size, this means that the field lines are characterised by **footpoints** on the surface. If the neutron star rotates, these footpoints and hence the magnetic field corotate with it. However, there exists a radial distance known as the **light cylinder** beyond which a co-rotating magnetic field line would be moving faster than light. Since in FFE the field lines carry charged particles, the force-free approximation breaks down and field lines extending beyond the light cylinder must be "open" (extend to infinite radii). This is illustrated in Figure 7.1. In our code, we make the hard restriction that FFE applies at all points in the magnetosphere that are within the light cylinder, and does not apply beyond it (Proposition 22.4).

Duncan & Thompson (1992) suggested that magnetar bursts are caused by an ultra-strong internal magnetic field, which is capable of plastically deforming the crust. This shifts the footpoints of the external magnetic field, creating twists in the magnetosphere. If the twists are strong enough, they may form a current sheet before returning to a stable configuration by undergoing magnetic reconnection and ejecting magnetic energy from the system in the form of a burst. These bursts may even be powerful enough to explain the four ultra-powerful **giant flares** that have been observed since 1979 (Chapter 7).

[1] "ZAMS" means "zero age main sequence", the mass of the star when it entered the main sequence and ignited hydrogen fusion.

30.2 Mathematical and numerical methods

This project aims to characterise the response of a neutron star magnetosphere to some initial disturbance, or **twist**, which we impart. We do this by writing a `C++` code to evolve Maxwell's equations under FFE. The evolutionary equations are described in Chapter 20, with their final forms given in Proposition 20.5.

The code evolves the magnetic field $\mathbf{B}$ and electric field $\mathbf{E}$ directly, as opposed to other abstracted quantities such as streamfunctions. This decision was made at an early stage, primarily to aid visualisation of the results, but evolving the streamfunction brings advantages such as guaranteeing that the magnetic field remains divergence-free (§9.4.1) and straightforward plotting of magnetic field lines as contours of streamfunction (§20.6).

We model the configuration in axisymmetric spherical polar coordinates, where the fields are expected to be independent of the azimuthal coordinate ϕ. We model the fields using a **pseudospectral method**; that is, the angular components (θ, ϕ) are handled by expanding into a set of basis functions, leaving the radial component (r) to be modelled separately by any method. In the full 3D case, the angular component is expanded as a series of **vector spherical harmonics** (Chapters 18 and 19). In axisymmetry, the ϕ-dependence drops out and the θ-dependence can be modelled by **Legendre polynomials** (Chapter 15), with their derivatives modelled by **associated Legendre functions** (Chapter 16). Spectral methods allow spatial derivatives, particularly divergence and curl, to be calculated exactly (Propositions 19.4 and 19.6). They can also guarantee divergenceless fields to remain so (Corollary 19.5), but this comes at the cost of taking an additional radial derivative and we find that the resulting numerical error outweighs the advantages of maintaining $\boldsymbol{\nabla} \cdot \mathbf{B} = 0$ (§24.3).

The radial dependence is handled by a finite-differencing scheme. We experimented with a differentiated Chebyshev series expansion (Chapter 14) but found it less accurate (§24.1 and §24.2). One of the reasons for the finite-differencing scheme proving more accurate is that we generalised some of the more widely known finite-difference expressions to consider an arbitrary number of neighbouring gridpoints in Chapter 12.

The simulation requires multidimensional numerical integration over the coordinates: spectral methods require 2D integrals for the full 3D consideration (Proposition 19.2) and 1D integrals in axisymmetry, Eqs. (19.50) to (19.52). We also require volume integrals to calculate the total energy of the system (§22.3) and to characterise $\boldsymbol{\nabla} \cdot \mathbf{B}$ across the domain (§24.5). To that end, we develop multidimensional numerical integration methods for spherical coordinates by generalising the trapezium rule in §10.5.

Time-integration is, of course, one-dimensional. We use a third-order Adams-Bashforth method (§10.3), which considers the field values at the previous two timesteps. This has a stabilising effect against numerical noise and contributes to the code's maintained accuracy over long timescales.

However, there exists an uncharacterised numerical instability within the code. This causes all three of our final simulations to blow up after roughly 7 to 9 light-crossing times. The instability is probably due to the radial treatment since code stability seems to be highly sensitive to the chosen number of radial gridpoints (§26.1), and by comparison seems to offer diminishing returns as more angular gridpoints are used (§26.2). We could not diagnose the instability but our choice of parameters minimised its effect.

The evolutionary code is numerically efficient: our final simulations (Chapters 27, 28 and 29) were performed on a conventional laptop on the order of five minutes.

30.3 Results

30.3.1 Stationary dipole

The stationary dipole (Chapter 27) should persist indefinitely, providing a measure of the long-term stability of the numerical scheme. To quantify this, we (1) calculate the value of a conserved quantity (the total energy contained within the fields) at each timestep and monitor its evolution (§22.3); (2) ensure that minimal numerical divergence of the magnetic field develops by calculating a dimensionless volume integral of $\boldsymbol{\nabla}\cdot\mathbf{B}$ and monitoring it over time (§24.5).

We find that, with our optimised number of coordinates found in Chapter 26, the model indeed remains stable for around $2-3$ light-crossing times, or $650-1000$ timesteps with our choice of ΔT. Figure 27.6 shows that the total energy remains relatively stable for this time, before rapidly increasing to high values. The length of time for which the system remains stable appears sensitive to the truncation index of the VSH decomposition of the fields $\ell_{\max}$, with the known exact result $\ell_{\max}=1$ surviving until around $T=5$.

The volume-integrated $\boldsymbol{\nabla}\cdot\mathbf{B}$ tells a similar story (Figure 27.7), although the sudden increase is replaced by a steady rise. Note that the absolute value is plotted; the calculated value oscillates about zero, demonstrating that the code is attempting to retain zero divergence. We conclude that artificial divergence is never a limiting factor in our simulations, with the effect of some undetermined numerical instability appearing first.

Deviations in the field values appear first around the inner and outer boundaries (Figures 27.3 to 27.5). We based our boundary conditions around those implemented by Pétri (2012) and Parfrey (2012), but further consideration may be required.

30.3.2 Pulsar model (rotating dipole)

We model a pulsar as a magnetic dipole with a constant angular velocity. The rotating dipole is simulated by imparting a nonzero electric field to the stationary dipole (§22.2). The rotation rate is ramped up linearly over around half a light-crossing time (§23.1), which was arbitrarily chosen but appeared to have little influence on the result. This ensures a smooth transition from the known stable stationary dipole to the rotating case. The ramping process appears to work smoothly, with the field values remaining close to their theoretical values until around $T=2$ (Figure 28.1). However, there appears to be a sudden jump toward the end of the ramp phase, observed as a rise in both the standard deviation of the VSH decomposition of $\mathbf{E}$ (Figure 28.2) and the total electric energy (Figure 28.9). The energy never recovers from this jump and continues to rise, so that the field structures are significantly altered by around $T=2$ (Figures 28.3 to 28.8).

Unlike for the stationary dipole, our choice of $\ell_{\max}$ is more difficult to make. The evolutionary equations guarantee that nonzero $\mathbf{E}$ will affect $\mathbf{B}$, so even after a few timesteps an analytic expression for the fields is difficult to find. In principle, infinitely many VSH series coefficients would then be required to describe them. In an ideal world, the magnitude of these coefficients would decrease as ℓ increases so that we could arbitrarily choose any "high enough" cutoff $\ell_{\max}$ beyond which additional terms contribute negligibly. However, the coefficients are calculated by numerical integrations which carry nonzero error, so increasing $\ell_{\max}$ allows more opportunities for error to contribute. The problem is exacerbated if the coefficients drop off relatively slowly with ℓ. If we could perform the integrals for the VSH coefficients analytically, it would be straightforward to compare the calculated and expected values in order to decide a cutoff. We choose $\ell_{\max}=20$ for all runs as a compromise. See Example 14.7 for a discussion of the same issue applied to Chebyshev series.

The rotating simulation demonstrates that our reinforcement of the force-free conditions at each timestep (§20.2) works well and contributes to the long-term stability of the simulation. As shown in Figure 28.11, it is only after $T=2$ when the percentage of gridpoints violating the force-free conditions rises significantly, after the jump toward the end of the rampup. Despite the percentage of gridpoints uncontrollably increasing with time after this point, there are short-term decreases which show that the code is actively working to maintain FFE. Further, Figure 28.12 suggests that individual violating gridpoints do not persist to the following timesteps, only that on average there are more as the simulation progresses. Interestingly, the violating gridpoints appear to be concentrated in arcs of constant radius, and perhaps are concentrated around the equator and $\theta=\frac{\pi}{4},\frac{3\pi 4}{4}$ as well as the boundaries. There is no clustering near the line defining the edge of our force-free domain, nor

is there significant discontinuity in the field structure around this line (Figures 28.3 to 28.8), indicating that splitting the domain into distinct force-free and relaxed regions does not cause further instabilities.

Figure 30.1 plots B_r for the stationary model and for the pulsar (constantly rotating) model at three light-crossing times. The rotational electric field has noticeably influenced the magnetic field by this point, which should not be surprising since we model the fields by coupled differential equations (Proposition 20.5). The large-scale structure of B_r is still preserved at this time, but the field appears to be developing a more intricate structure, especially in the boundaries of the $10^{-2} - 10^{-1}$ contour. This may be physical, or a result of the Gibbs phenomenon.

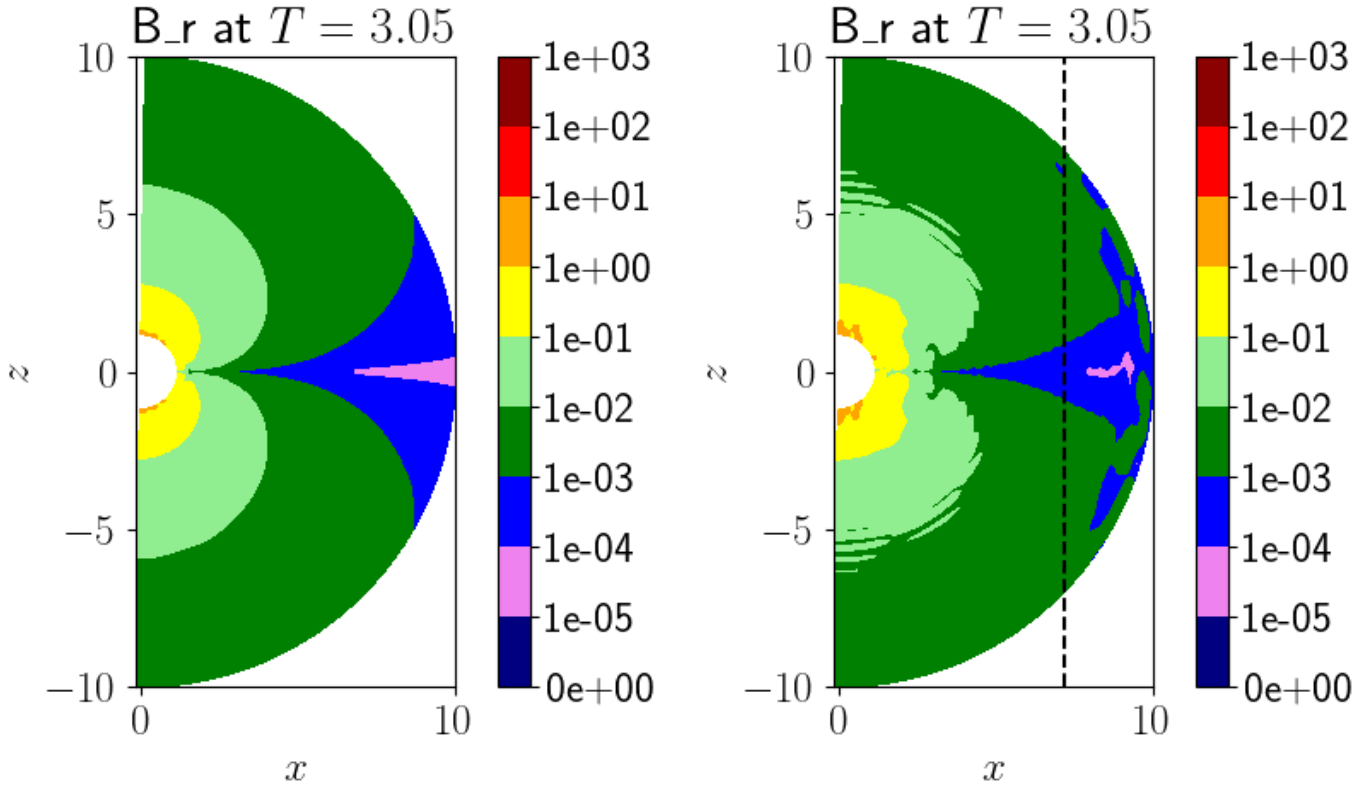

Figure 30.1: Contour plots of B_r at $T = 3.05$ for the stationary model (*Left*) and for the pulsar model (*Right*). The dashed black line represents the light cylinder.

30.3.3 Magnetar model (constant magnetospheric twist)

The twisted magnetosphere is simulated by defining a region within which the rotation rate is allowed to grow higher than that of the magnetosphere itself. Similarly to the global rotation rate, this twist is ramped up linearly to ensure a smooth transition from the initial configuration. We are also careful to ramp up smoothly over the coordinates, minimising jump discontinuities in the rotation rate between the twisted and non-twisted regions that may cause jump discontinuities in the field values there (Chapter 25). We do not find a perfect smoothing function, but are able to produce a noticeable and fully controllable smoothing effect (§25.3).

Two twist configurations are tested: a shear contained entirely within the northern hemisphere to avoid any cancellation of its effect by symmetry across the equator $\theta = \frac{\pi}{2}$, and then a symmetric twist once this was confirmed, enabling comparisons with the literature.

In both cases, after the electric field within the twisted region has been raised by the ramping process, the code immediately acts to restore equilibrium by reducing $|E_r|$ and $|E_\theta|$ (Figure 29.1). The simulation overshoots and by this time ($T \approx 4 - 5$) the effects of the numerical instability have begun to dominate the evolution.

There is definite interaction between the twisted and non-twisted regions (Figures 29.3 to 29.14), with the localised amplified electric field spreading throughout the domain over time. In particular, we may expect the influence of the northern-hemisphere twist to be felt in the southern hemisphere after around π light-crossing times if the signal travels at the speed of light, and Figures 29.6 and 29.7 appear roughly in line with this. These heatmaps of field values also show that there is little uncontrolled growth near the boundaries of the twisted region - the most significant uncontrolled growth is still at the inner and outer boundaries of the domain. Our spatial smoothing function thus behaves sufficiently well.

Figure 30.2 plots B_r for both twisted models at three light-crossing times. In both, the field is noticeably affected within the twisted region and by this time the effect has started to spread to the non-twisted region - for example, not the distortion of the outer boundary of the $10^{-2} - 10^{-1}$ contour near the north pole. Both twists appear to have had a similar influence on the region $\theta \in [0, 30^\circ]$ (the sector between the north pole and the twisted region) by this time.

The model with a symmetric twist survives for longer than that with the northern hemisphere only. We suggest that the system becomes a close approximation of the rotating dipole if the twist is symmetric, a configuration that our scheme can model with reasonable stability.

We should not be surprised that the code's effect is a stabilising one: the Adams-Bashforth time integration method and force-free condition enforcement in particular have already been highlighted as working to maintain stability. Modelling rotating as an electric field yields an additional contribution to the total energy of the system (§22.3), so if stability is obtained by minimising total energy, reducing $|E_r|$ and $|E_\theta|$ is a logical action to take. It is interesting that the code did not take the same action when ramped up from stationary to rotating without twists; we suggest that this is because the rotating dipole is inherently stable and that the rampup mechanism is sufficiently smooth not to introduce additional instabilities.

Figure 29.2, however, suggests that the twisted configuration is still somewhat stable because the standard deviation of the VSH decomposition of the fields (a measure of the numerical noise within the system) remains roughly constant from the end of the twist rampup to around $T = 4$. Perhaps this is because the twist is relatively low in magnitude.

Indeed, in neutron star magnetospheres we expect low-magnitude twists to form regularly and persist for longer timescales, so our code is a viable candidate for modelling these common events. Bursting events require twists of high enough magnitude for current sheets to form, but this process is only possible in the presence of resistivity. Since the code assumes zero resistivity everywhere, bursts cannot yet be modelled.

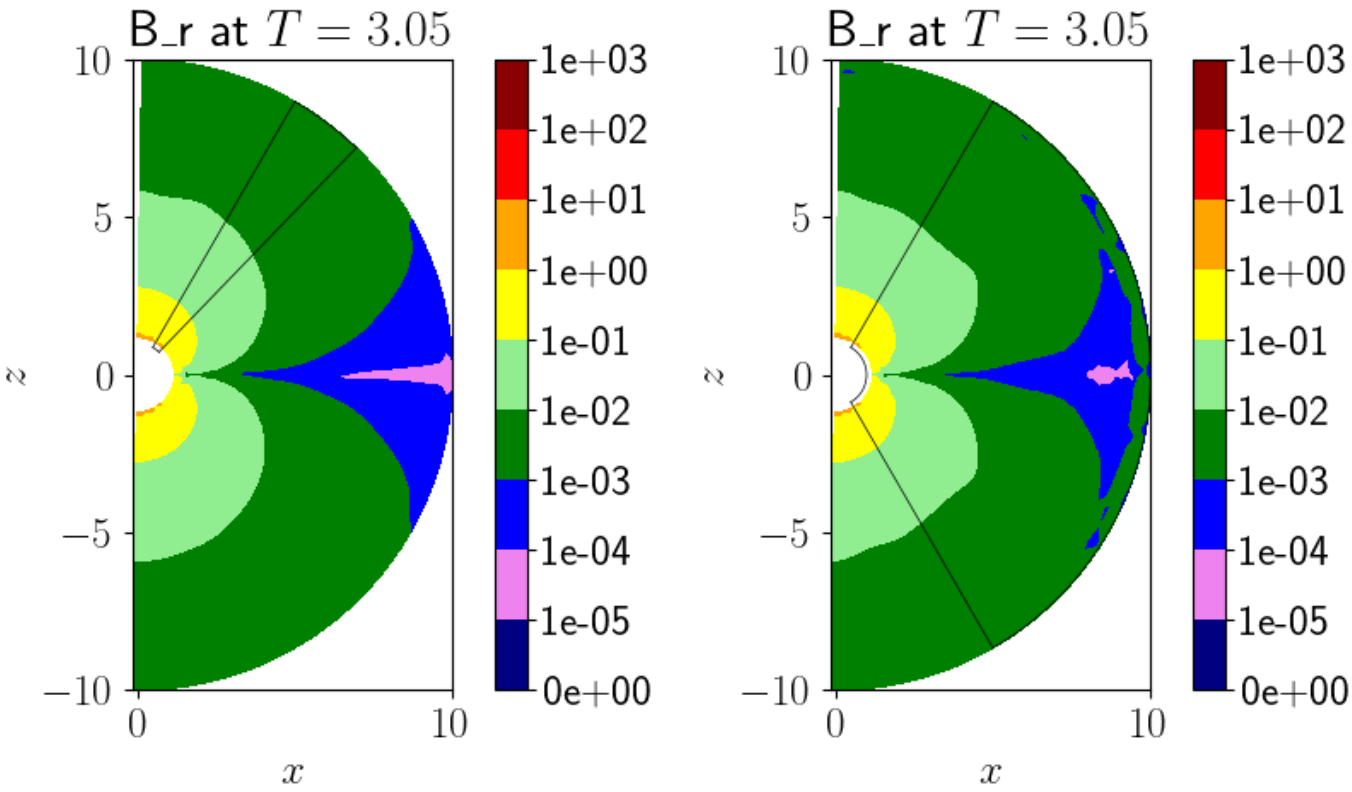

Figure 30.2: Contour plots of B_r at $T = 3.05$ for the two magentar (twisted) models. *Left:* Localised twist in northern hemisphere. *Right:* Symmetric twist about equatorial line. The black box represents the twisted region.

30.4 Further discussion

We attributed the instability of the code to a possible numerical instability in the radial direction, but another possibility is that high-frequency is that high-multipole features begin to form at later times (Figure 30.1). This may be due to the **Gibbs phenomenon**, a consequence of approximating functions by Chebyshev or Legendre series where, instead of converging to the true value as we increase the truncation index $n_{\max}$ or $\ell_{\max}$, the approximation oscillates increasingly rapidly about it (Boyd, 1996; Zhang, 2022). A classic example is the approximation of a square wave (e.g. Arfken & Weber, 2005, §14.5). Discontinuities or short intervals over which the function changes rapidly, like current sheets, are particularly susceptible to Gibbs' phenomenon. The effect may countered by **spectral filtering** (e.g. Gottlieb & Shu, 1997), but if our aim is to characterise these sharp regions, filtering may not be desirable.

Recently, Contopoulos et al. (2024) presented another possible explanation: there may exist more than one stable configuration for the magnetic field of a rotating neutron star. In particular, a solution may be possible based on a dipole whose field lines do not extend to infinity but are contained within the light cylinder and hence are all closed. This is interesting because our numerical scheme clearly attempts to return to an equilibrium solution, but perhaps this equilibrium is not the one in which real neutron stars exist.

The difficulties in code stability during development meant that we ran out of time to implement the relevant mechanisms to simulate a reconnection event. However, the response mechanism of our code, by attempting to restore the configuration to equilibrium, still has useful applications in modelling the more-frequent and lower-energy persistent twists that we expect in neutron stars and in the Solar magnetosphere.

30.5 Suggestions for the future

The code assumes zero resistivity everywhere, and this precludes the formation of current sheets and hence the onset of magnetic reconnection. Then, numerical resistivity must be implemented in order to model giant flare formation.

Despite its popularity, FFE cannot give a complete description of neutron star magnetospheres because a component of $\mathbf{E}$ parallel to $\mathbf{B}$ is required in order to drive particle acceleration and emission. The disadvantage of relaxing $\mathbf{E} \cdot \mathbf{B} = 0$ is that the current $\mathbf{J}$ is no longer uniquely determined; nevertheless, Kalapotharakos et al. (2012) explored pulsar magnetospheres without the force-free condition.

Although nonzero magnetic field divergence never appears to be a limiting factor (Figures 27.7, 28.10 and 29.16), the code does not feature a divergence-cleaning mechanism. The figures show that the numerical divergence grows over time in simulations and is considerably higher for rotating or twisted configurations than for a stationary dipole, so this may require consideration in more detailed simulations.

Our simulations appear to break down first near the inner and outer boundaries. We have implemented the same boundary conditions as Pétri (2012) and Parfrey (2012), with a special consideration given for the non-rotating case (§20.4 and §20.5). We also add a sponge layer near the outer boundary (Parfrey, 2012, §3.9). Further consideration of these may improve code stability.

The domain uses linear spacing in both the radial and polar coordinates (§23.2). However, we discuss in Chapter 27 that alternative mappings may prevent clustering of gridpoints around the inner boundary, and might be employed in order to shift accuracy to regions that require it. In principle, the code may be modified to suit arbitrary coordinate mappings. The numerical integration schemes would still apply, but the finite area and volume elements would need to be calculated carefully (e.g. Proposition 10.3). However, our finite differencing scheme assumes a constant grid spacing. Perhaps separate expressions for variable spacings can be derived by similar methods to those used in Chapter 12, but this was not attempted. Instead, we recommend handling the radial dependence with a Chebyshev series which may be differentiated term-by-term (Chapter 14). The code retains this functionality (e.g. §24.2).

Part V

Appendices

31 Running the code

31.1 Installing C++ on a personal computer

These are brief notes on how to install C++ on a Windows computer, outlining the process followed in the video youtube.com/watch?v=0HD0pqVtsmw
They are accurate as of 15 February 2022.

1. Go to mingw-w64.org and download MSYS2 (sometimes stylised as Msys2), or alternatively go directly to msys2.org The file is 91.2 MB in size.

2. Run the file and click Next until it begins. Leave "Run MSYS2 64bit now" checked, and click Finish.

3. Now we will install pacman. Open a terminal, e.g. by clicking Start, typing cmd and hitting the Enter key. Type pacman -Syu and then y twice to confirm. (The latter mentions that an error has occurred, but this is expected.) This will close the terminal.

4. Open MSYS2 MSYS. Update with pacman -Su and y. When finished, close the terminal.

5. Now open MSYS2 MinGW 64-bit (or MSYS2 MinGW x64). Search for the gcc package with pacman -Ss gcc. Look for the option that says mingw64/mingw-w64-x86_64-gcc. The line will also have (C,C++,OpenMP) on it. Highlight and copy this text: mingw64...-gcc. Paste it as follows:

   ```
   pacman -S mingw-w64-x86_64-gcc
   y
   ```

   Check version numbers with

   ```
   gcc --version
   g++ --version
   ```

   This gives a way to confirm that gcc and g++ have been installed correctly.

6. Now we will install the debugger gdb. Type

   ```
   pacman -Ss gdb
   ```

   and then

   ```
   pacman -S mingw-w64-x86_64-gdb
   y
   gdb --version
   ```

   Close the terminal.

7. Now we will set the path environment variable. This tells the computer where to look when interpreting the command gcc etc. To see why it's needed, go to cmd.exe and type gcc --version. It will return 'gcc' is not recognized as an internal or external command, operatable program or batch file. Browse to the location of gcc, g++ and gdb:

   ```
   C:\msys64\mingw64\bin
   ```

Go to

```
Start -> Edit the system environment variables
-> Advanced -> Environment variables -> (System variables part of the box) -> Path.
```

With this highlighted, click

```
Edit... -> New
```

and paste the link. Press the `Enter` key, then `OK` three times to close those windows. Now open `cmd.exe` and `try gcc --version` again.

You can now compile and run `C++` codes from `cmd.exe`. A command-line emulator is recommended, which allows you to use Linux-style commands like `ls`, `pwd` etc. A good example is `cmder`.

31.2 Installing Python on a personal computer

These are brief notes on how to install `Python 3` and the IDE `Anaconda` on a Windows computer, accurate as of 15 February 2022.

1. Go to anaconda.com/download and download `Anaconda Individual Edition`. The file is 510 MB in size.

2. Run the executable file `Anaconda3-2021.11-Windows-x86_64`. Click through the text boxes that occur, checking the following boxes when they appear:

   ```
   Install for: Just me (recommended)
   Register Anaconda3 as my default Python 3.9
   ```

3. Once the installation is complete, open the `Start` menu and search for `Spyder (anaconda3)`. This will open the `Anaconda` program, which contains a script editor plus a dedicated console for running the code.

31.3 Running the code on a personal computer

The evolutionary code consists of three `C++` files:

```
Initial_conditions_xx.h
Header_Time_Evolution_xx.h
Time_Evolution_xx.cpp
```

That is, two header files and a single code file, only the latter of which is compiled and ran. The subscripts `_xx` refer to version numbers; we maintain an archive of all previous versions of each file for version control. Files contain brief descriptions of the changes made since the previous version in the multi-line comment at the top; the user need not be concerned with these changes unless they are tracking issues to previous versions. The user can assume that they have the most up-to-date versions of each file, but may need to update the `#include` lines at the top of `Time_Evolution_xx.cpp` if different versions of the header files are available.

To run the code, open a terminal and navigate to the parent folder containing the three `C++` codes. For example, on the author's computer, it would be

```
cd OneDrive\PhD\Codes\20230810 Time evolution with updated equations
```

Then, the `Time_Evolution_xx.cpp` file must be compiled by

```
g++ Time_Evolution_xx.cpp -o Time_Evolution --std=c++11
```

where the added `--std=c++11` is required because some of the functions and object types used by the code were only developed as early as `C++11`. This produces an executable file, which is ran by

```
Time_Evolution
```

or, depending on the user's computer system,

```
./Time_Evolution
```

If the code is running successfully, the user will see output to the console similar to that shown in Figure 31.1 and described below.

First, all user-chosen values are output as described in §31.7 and §31.8, along with the timestamp at which the execution began. Before evolution begins, a message appears reminding the user that the execution can be safely stopped at any time by the normal method, pressing `CTRL-C` or similar. Output files are appended as the code runs, so data will still be available if the code is stopped early.

Then, the evolution begins and properties of the neutron star are output at certain timesteps according to the choice of `int cout_freq_T` in §31.8. Here, the user chooses a single gridpoint by indices `cout_i` and `cout_j`, and tracks the evolution of this gridpoint with time. If one particular region is known to be troublesome, e.g. the outer boundary, a point within this region can be chosen to give immediate feedback when results begin to diverge from what is expected. Alternatively, if the choice of gridpoint is not important, one can simply watch for when values begin to diverge rapidly compared to previous timesteps, or when `nan` values appear. Finally, a timestamp is given at the end of the evolution and the final execution time is output.

The user may change the values output to the screen at each timestep within the functions

```
void output_headers_to_screen()
void output_to_screen()
```

contained within `Header_Time_Evolution_xx.cpp`. The parameter `% E pts chngd` gives the percentage of values of the electric field that were updated in order to satisfy the force-free conditions. The parameter `% done` is calculated by scaling the current timestep to the final timestep, and the parameter `Est. time left` is calculated by scaling the elapsed time so far, to the number of timesteps remaining. It should be interpreted only as a rough estimate, e.g. when deciding whether to commit to a run at a certain resolution

All values output to the terminal are also saved to the `_Log.txt` file to give a permanent record. If console output is not available, e.g. if the code is ran on a computing cluster, the log file can be opened to gauge progress at any time. It is advisable to first create a copy of the log file and open that instead, to avoid issues where the code tries to write to an open file.

All numerical evolution is performed with the above `C++` codes. The result is a set of `CSV` files, which can be analysed by `Python` codes for future analysis and graph plotting, and a `txt` file which acts as a log for referencing previous runs.

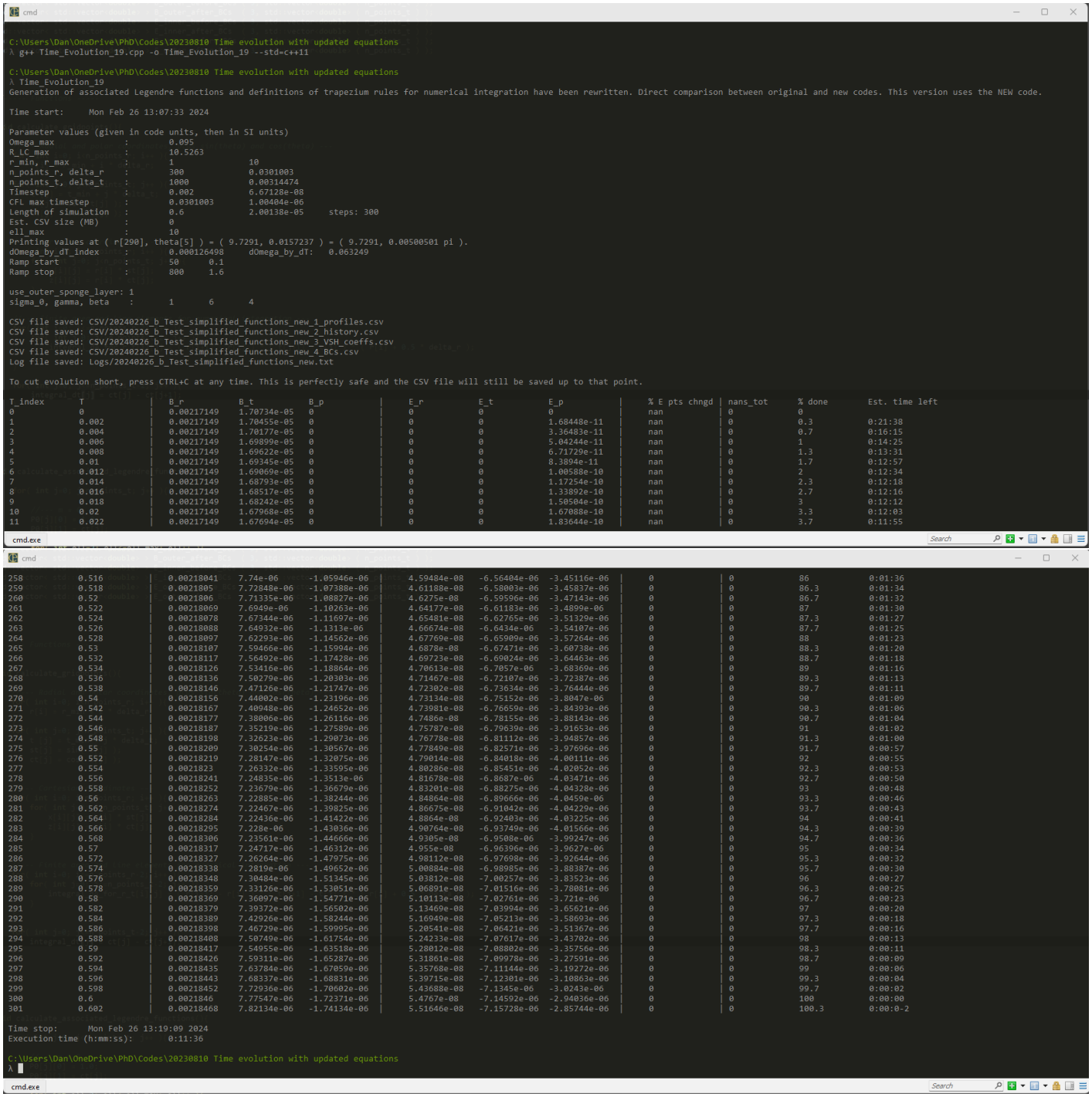

Figure 31.1: Output to the console after successful running of the code. Output is given line-by-line in real time; these screenshots were taken after it had finished.

31.4 Running the code on the Ada computing cluster

Ada is the computing cluster at the University of East Anglia. For more intense runs, or for a sequence of consecutive runs, it is preferable to use this cluster as opposed to a personal computer. Instructions on how to access the cluster off-campus from a personal computer are below, accurate as of 12 December 2023. There are two preliminary steps:

1. Register your UEA account with ADA: contact IT Services for instructions on how to do this.
2. Download the UEA VPN: go to vpn.uea.ac.uk and log in with your UEA account. Navigate to Download VPN (Windows) or similar. This will download the programme Big Edge Client, with filename BIGIPEdgeClient.exe. Run this file.

Then, the process is as follows.

1. Run the programme BIG-IP Edge Client.
2. Change the server to vpn.uea.ac.uk.[1]
3. Click Connect.
4. Open a web browser and go to adaood01.uea.ac.uk.
5. Log in with your UEA account. This brings you to the Ada On Demand console. Use the open-on-demand as normal; start a terminal or open a virtual desktop.
6. Open a virtual desktop by navigating to

   ```
   Interactive Apps -> Desktop (XFCE)
   ```

   Choose the parameters that suit your needs for that session and click Launch. In particular, be realistic with the Number of hours setting; if you finish with the desktop early, computing power will still be reserved until chosen duration has passed, lessening the resources available to other users.
7. The virtual desktop features a command-line terminal xterm, but it doesn't recognise the ADA job commands sbatch, squeue etc. Instead, go back to Ada On Demand and open Clusters -> _Ada Shell Access. This opens a terminal which does recognise the commands.
8. On the virtual desktop, navigate to the folder from which you will perform the evolutions. For the author, this is

   ```
   /gfps/home/abc12def/Time_evolution_codes
   ```

   Ensure that the folder contains the subfolders CSV and Logs as described in §31.5, along with the subfolder Output_and_error_files and the most up-to-date versions of the following files:

   ```
   Time_Evolution_xx.cpp
   Header_Time_Evolution_xx.h
   Header_Initial_Conditions_xx.h
   sbatch_xx.sh
   ```

9. Open Header_Time_Evolution_xx.cpp, navigate to void count_nans() and change std::isnan to isnan and std::isinf to isinf. Otherwise, the code will not compile on Ada.

[1] The default vpn2020.uea.ac.uk will not work; it will get stuck in a loop alternating between Downloading server settings... and Waiting to connect to server....

10. Open the shell script file `sbatch_xx.sh`. This is a minimally-changed script provided by IT services during training sessions for `Ada`. Update the relevant variables as they are described. In particular, it is good practice to update

    ```
    #SBATCH --job-name=yyyymmdd_a
    #SBATCH -o Output_and_error-files/yyyymmdd_a_%j.out
    #SBATCH -e Output_and_error-files/yyyymmdd_a_%j.err
    ```

    with a unique name for the run.[2]

11. Open `Header_Initial_Conditions_xx.h` and set the parameters desired for the run.

12. Go to the terminal opened in step 7, `cd` to the target folder (here `Time_evolution_codes`) and compile the `C++` code in the usual way:[3]

    ```
    g++ Time_Evolution_xx.cpp -o Time_Evolution --std==c++11
    ```

13. Run the shell script with

    ```
    sbatch sbatch_xx.sh
    ```

    A message will appear, confirming that the job has been submitted:

    ```
    Submitted batch job 12345678
    ```

    Here, `12345678` is a job ID which `Ada` assigns and feeds back to you. You will receive an email when the job starts, which will not be immediately after submission because every job must be scheduled, and when the job completes. To check the progress of a job, type

    ```
    squeue -u abc12def
    ```

    To cancel a job, type

    ```
    scancel 12345678
    ```

31.4.1 Running `Python` on the virtual desktop

For simple codes, the following process may be used.

1. Open the code, which in this example is named `mycode.py`, in a text editor such as `emacs` and perform the edits required. If the code can be ran as-is, this step may be skipped.

2. Open a console such as `xterm` or `_Ada Shell Access` and type

    ```
    python &
    ```

    The `&` signifies that you still wish to use `bash` commands like `ls`, `cd` etc.

3. `cd` to the directory containing the code and run it with

    ```
    python mycode.py
    ```

[2] In this example, the convention `yyyymmdd_a` is chosen to enumerate jobs by the date started and then a lower-case Latin letter `a`, `b` etc. Note that you must physically click the `Save` icon to save any files you edit on the virtual desktop; keyboard shortcuts such as `CTRL-S` may not be recognised.

[3] Sadly, this will not be recognised if included in the shell script, so must be entered manually to the console.

The above does not work for codes that use modules like `matplotlib`, the most widely used graph-plotting module. To get around this, we must open an interactive `Jupyter` session as follows.

1. Go to `Ada On Demand` and open

   `Interactive Apps -> JupyterLab (Beta)`

2. Open the `Python` code that you want to run.

3. Right-click somewhere within the opened file and select `Create console for editor`. You can now run code snippets by highlighting them and either selecting `Run -> Run Code` or using the keyboard shortcut `Shift-Enter`. To run the entire code, press `CTRL-A` and `Shift-Enter`.

Note that `LaTeX` formatting does not work. You must replace the usual lines, e.g.

```
plt.rcParams.update({ "font.size":18 })
plt.rc( "text", usetex=True )
```

by the line

```
plt.rcParams.update( plt.rcParamsDefault )
```

Otherwise, you will get the following error message:

```
RuntimeError: Failed to process string with text because latex could not be found.
```

A further complication is that the virtual desktop has no installed software to view `PNG` images. You will have to make do with the inline preview image that `Python` or `Jupyter` shows when running the code, or email the file to yourself to view on a different computer.

31.4.2 Running the code directly on a university computer

The user may wish to log in to a university computer and run the code through a terminal without submitting to `Ada`. The process is as follows.

1. Ensure that the target folder is up-to-date as described in step 8 above, minus the shell script `sbatch_xx.sh`.

2. Launch the programme `PuTTY` and sign in with your UEA account `abc12def` and password. You should already be in the `ADA` directory,

   `/gpfs/home/abc12def`

3. Launch the programme `WinSCP`. Sign in to `Ada` and enter your password.

4. Send updated versions of the following files to your UEA documents:

   ```
   Time_Evolution_xx.cpp
   Header_Time_Evolution_xx.h
   Header_Initial_Conditions_xx.h
   ```

   Using `WinSCP`, move them from e.g. `Downloads` to the target folder from which you will perform the computations. For the author, this is

   `/gfps/home/abc12def/Time_evolution_codes`

   Ensure that the subfolders `CSV` and `Logs` exist within this target folder, as described in §31.5.

5. Go to `PuTTY` and type

```
interactive
```

Your location output will now change from

```
[abc12def@login01 ~/Time_evolution_codes]$
```

to

```
[abc12def@c0010 ~/Time_evolution_codes]$
```

or similar.

6. Go to PuTTY and compile and run the C++ code in the normal way:

```
cd Time_evolution_codes
g++ Time_Evolution_xx.cpp -o Time_Evolution_xx --std=c++11
./Time_Evolution_xx
```

31.5 File structure

It is vital to use the same file structure for which the codes were written. Suppose that we have some parent folder, which may be saved anywhere on the user's computer. Within this parent folder should be the four codes above, plus the following subfolders:

1. Archive: Previous versions of the code files.
2. CSV: CSV files containing values of the electric and magnetic fields as a function of position and time, for further analysis. These CSV files are the main output of the C++ code.
3. Figures: Graphs plotted from the data using the accompanying Python codes. To avoid clutter, they are divided into subfolders by date of code execution in YYYYMMDD format (which makes alphabetical sorting equivalent to chronological sorting). This was a convention used by the author but is not necessary to follow.
4. Logs: Text files maintaining copies of all results output to the console during running of each code. The files contain the values of all changeable parameters, plus a brief comment about why the run is being performed. This allows the user to trace their choice of parameters through previous evolutions, and also provides a more permanent history of code execution than CSV files that will require frequent deletion to save space.
5. Python analysis codes: Generalised codes written in Python which can read any of the CSV files and produce graphs to visualise the data, allowing for publication of results or comparison between successive runs.
6. Test codes: Extra codes described throughout this Part, which test the numerical methods as they appear in the main evolution code by reading the header files directly, as opposed to copying code into a standalone file for testing. This prevents issues of code divergence when updates are made, giving confidence that we are always testing the actual code that will be ran during evolution and not a previous version of it.

Of these, only CSV and Logs are critical to the running of Time_Evolution_xx.cpp.

31.6 Output files

The code outputs results to up to seven files. The filenames are all prefixed by the user's choice of

```
std::string output_filename
```

which we denote by `XXX` below. The `CSV` files are saved within the folder `CSV`, while the log file is saved within the folder `Logs`.

1. `XXX_1_time_values.csv`: The values of time, angular velocity and rotation period of the magnetosphere, light-cylinder radius, and total angle rotated, at each timestep. This is read by analysis `Python` codes, so is always created.

2. `XXX_2_gridpoints.csv`: The values of the radial coordinate in code units and the polar coordinate in radians as a function of the coordinate indices. This is read by analysis `Python` codes, so is always created.

3. `XXX_4_profiles.csv`: A series of snapshots of the system at regular intervals, with samples of spatially varying quantities such as the magnetic field values. These may be used to plot, for example, the heatmaps of the field components in Figure 27.3 etc by the Python code `Plot_Profiles.py`. Depending on the chosen temporal and spatial output frequency, controlled by

   ```
   csv_profiles_write_freq_T, csv_profiles_write_freq_r, csv_profiles_write_freq_t
   ```

   this file can become very large in size. It will likely be the largest of the files created. A rough estimate of the final size of this file is made by the function

   ```
   double estimate_csv_profiles_filesize();
   ```

   and output to the terminal and log file upon code execution. This file is optional.

4. `XXX_5_history.csv`: Values of quantities that are constant throughout the domain, such as the total energy, output as a function of time. These may be plotted by the Python code `Plot_History_Multiple.py` to produce graphs such as Figure 27.6. This file is optional.

5. `XXX_6_VSH_coeffs.csv`: The values of the numericaly-calculated VSH coefficients of the fields and their radial derivatives, at a sample of radial gridpoints, as a function of time. This may be useful for diagnosing the accuracy of the VSH decomposition or the radial derivatives. This file is optional.

6. `XXX_7_BCs.csv`: The values of the field components at both the inner boundary and the outer boundary, as a function of θ, both before and after the application of boundary conditions. This may be useful for diagnosing the consideration of the boundary conditions by the code. This file is optional.

7. `XXX.txt`: A log file which records all parameters chosen in `Header_Initial_Conditions_xx.h`, a brief comment about the purpose of the run, set by the user as

   ```
   std::string log_file_comment
   ```

   and the complete list of values that were output to the terminal during code execution. This facilitates the maintaining of a full record of runs that is descriptive and repeatable, so is always created.

31.7 Notation convention

The notation convention within the C++ codes is as follows:

1. The three spherical polar coordinates r, θ, ϕ are denoted `r`, `t` and `p`. We choose single characters to guarantee alignment between similar lines of code, improving readability and error catching.

2. The radial gridpoints are contained within a list `r`, with `n_r` values ranging from `r_min` to `r_max` and enumerated by `i`, such that `r[0]` is `r_min` and `r[n_r-1]` is `r_max`.

3. The polar-angle gridpoints are contained within a list `t`, with `n_t` values ranging from `t_min` to `t_max` and enumerated by `j`, such that `t[0]` is `t_min` and `t[n_t-1]` is `t_max`.

4. Time in code units is denoted by `T`. The timesteps are enumerated by `T_index`. Note that time values are not stored in an array; instead, we increase `T` at each timestep until `T_index_max`.

5. The associated Legendre function index ℓ is denoted by `ell`, ranging from `0` to `ell_max`.

6. The Chebyshev polynomial index n is denoted by `n`, ranging from `0` to `n_max`.

7. The rotational angular velocity of the neutron star is denoted by `Omega`, capitalised to distinguish its mathematical symbol Ω in this text from ω.

We do not anticipate situations where ranges other than $\theta \in [0, \pi]$ will be necessary, and cannot guarantee that calculations or integrals will be accurate when restricting θ to smaller ranges than this, so we relegate the assignments `double t_min = 0` and `double t_max = pi` to `Header_Time_Evolution_xx.h` to avoid tempting the user to change them.

31.8 User-defined variables in `Header_Initial_Conditions_xx.h`

All of the free parameters available to the user are contained within `Header_Initial_Conditions_xx.h`. Below are brief descriptions of selected parameters, updated 25 February 2025.

1. `double B_r_function( double r, double t )`, `double B_t_function( double r, double t )` and `double B_p_function( double r, double t )`: Functions encoding mathematical expressions for the chosen initial magnetic field, with a separate function for each spherical polar coordinate.

2. `double delta_T`: The length of each timestep in code units.

3. `double P_SI_final`: The final rotation period of the neutron star once rotation has fully ramped up, in SI units (seconds).

4. `int T_index_rotation_ramp_start` and `int T_index_rotation_ramp_stop`: The values of `T_index` at which we begin and end rotation ramp-up, so that `T_index_rotation_ramp_start` is the first value of `T_index` with nonzero `Omega[T_index]` and `T_index_rotation_ramp_stop` is the first value of `T_index` at which `Omega[T_index]=Omega_final`.

5. `int T_index_twist_ramp_start` and `int T_index_twist_ramp_stop`: The values of `T_index` at which we begin and end ramp-up of rotation in the twisted region.

6. `int twist_final_Omega`: The final rotation rate of the twisted region, on top of the global rotation rate.

7. `int twist_r_min` etc: The radial coordinate and angular coordinate at which the twisted region begins, ends, and reaches 99% of its maximum value from below (`twist_r_min_99pct`) and from above (`twist_r_max_99pct`). These parameters control the size, location and smoothing over the coordinates of the twisted region (Chapter 25).

8. `std::string output_folder`: The output folder into which result files are saved. This must contain the subfolders `CSV` and `Logs`. If not, or if there is a typo in the chosen path, output files will not be saved. The code will still run without issue, so this may be difficult to diagnose.

9. `std::string output_filename`: Desired prefix for output files.

10. `std::string log_file_comment`: A brief description of why this particular run is being performed, to remind the user of the significance of this run when browsing through old log files.

11. `int cout_i` and `int cout_j`: The coordinates of an arbitrarily chosen gridpoint at which parameters such as the field values are output to the terminal, the log file and `XXX_5_history.csv` at each timestep. This can provide a real-time indicator of the general conditions of the system, or monitoring of the conditions in a particularly troublesome region.

12. `int cout_freq_T`: The number of timesteps between subsequent outputs to the terminal.

13. `int csv_profiles_write_freq_r` etc: The number of gridpoints or timesteps between outputs to the `CSV` files. The endpoints `i=0`, `i=n_points_r-1`, `j=0` and `j=n_points_t-1`, `T_index=0` and `T_index=T_index_max` are always included. These enable smaller output filesizes, especially if the runs require many gridpoints for computational accuracy.

14. `int csv_profiles_write_i_min` etc: Only output to the `CSV` files a particular region or window of time. Useful if high-resolution results are required when diagnosing issues occurring in known regions or times.

15. `bool use_outer_sponge_layer`, `double r_sponge` etc: Option and controlling parameters for the sponge layer near the outer boundary; see Eq. (3.40) of Parfrey (2012).

31.9 Process in `Time_Evolution_xx.cpp`

Here, we outline the order of processes carried out at each timestep during evolution, as they appear in `Time_Evolution_xx.cpp`. The functions themselves are all contained within `Header_Time_Evolution_xx.cpp`, and we relegate detailed descriptions of these functions to the verification tests performed throughout the rest of this document.

First, we perform a few prerequisite steps:

1. Include header files, featuring some standard `C++ headers` and the files `Initial_conditions_xx.h` and `Header_Time_Evolution_xx.h` from which parameter values and function definitions are read.

2. Create all the `CSV` and `_Log` files so that they can appended do whilst the code runs. Output chosen values of the parameters to the screen and the log.

3. Determine the gridpoints and pre-calculate the Legendre polynomials, associated Legendre functions, Chebyshev polynomials, integration finite length/area elements and sponge layer friction terms (Parfrey, 2012, §3.9). We find that pre-calculation drastically reduces code runtime.

4. Calculate the maximum timestep under the CFL condition and output it along with the user-defined `Delta_T`. Provide a warning message if the CFL condition is violated but do not fail the code.

5. Fail the code if the chosen parameters will cause issues such as list overflow.

Following this, we begin numerical evolution. The process at each timestep is as follows.

1. Increment the value of time.
2. If we are within the rotation ramp-up phase so that `T_index` is greater-than-or-equal-to `T_index_rotation_ramp_start` but less-than-or-equal-to `T_index_rotation_ramp_stop`, increase its angular velocity `Omega` and update `star_rotation_angle`, the total angle in radians through which the star has rotated during the simulation.
3. Do the same for twisted region.
4. Apply the boundary conditions (§20.4 and §20.5) and pre-calculate the values of $B^2, E^2, \mathbf{E}\cdot\mathbf{B}$ across the domain.
5. Enforce the force-free conditions (§20.2) and pre-calculate $B^2, E^2, \mathbf{E}\cdot\mathbf{B}$ again, since they will have changed in general.
6. Expand $B_r, B_\phi, E_r, E_\theta, E_\phi$ as VSH series.
7. Calculate the radial derivatives of the fields by taking finite-difference expressions of their VSH coefficients. This step may be substituted for a Chebyshev decomposition and subsequent determination of its radial derivative.
8. Either (a) Expand B_θ as a VSH series and calculate its radial derivative by a finite-difference expression, the same as above, or (b) calculate the second radial derivative of the B_r VSH coefficients and use this to calculate the B_θ VSH coefficients and their radial derivatives. Following tests in §24.3, we choose method (a).
9. Calculate the standard deviation of the VSH decomposition relative to the numerical field values at each gridpoint, in order to quantify the accuracy with which this calculation is being performed. Calculate the radial derivatives of the field components themselves, having already calculated the radial derivatives of the VSH coefficients. These are not used by the code but are saved to the output values to allow for sanity-check calculations.
10. Calculate the spatial derivatives $\boldsymbol{\nabla}\cdot\mathbf{B}, \boldsymbol{\nabla}\cdot\mathbf{E}, \boldsymbol{\nabla}\times\mathbf{B}, \boldsymbol{\nabla}\times\mathbf{E}$ and hence the current density $\mathbf{J}$ and the time-derivatives of the electric and magnetic fields from Eq. (20.5).
11. Calculate the total energy of the domain (§22.3) and the dimensionless volume integral of $\boldsymbol{\nabla}\cdot\mathbf{B}$ (§24.5) as measures of code performance, and determine at how many gridpoints the field values have become `nan` or infinite.
12. Output to the `CSV` and log files and to the terminal.
13. Increment the time.

Bibliography

Alexeyev E., Alexeyeva L., Krivosheina I., Volchenko V., 1988, Phys. Lett. B, 205, 209

Aly J. J., 1984, ApJ, 283, 349

Aly J. J., 1991, ApJL, 375, L61

Antonioli P., et al., 2004, New J. Phys., 6, 114

Aptekar R. L., Frederiks D. D., Golenetskii S. V., Il'inskii V. N., Mazets E. P., Pal'shin V. D., Butterworth P. S., Cline T. L., 2001, ApJS, 137, 227

Arfken G. B., Weber H. J., 2005, Mathematical Methods for Physicists, 6 edn. Elsevier Academic Press, Cambridge, MA, US, https://ui.adsabs.harvard.edu/abs/2005mmp..book.....A

Atkinson K. E., 1989, An Introduction to Numerical Analysis. John Wiley & Sons, NY, US, http://www.worldcat.org/isbn/0471500232

Atteia J. L., et al., 1987, ApJL, 320, L105

Axelrad V., 1998, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17, 149

Baade W., 1942, ApJ, 96, 188

Baade W., Zwicky F., 1934a, Proceedings of the National Academy of Science, 20, 259

Baade W., Zwicky F., 1934b, Physical Review, 46, 76

Backus G., 1986, Reviews of Geophysics, 24, 75

Badenes C., Maoz D., Draine B. T., 2010, MNRAS, 407, 1301

Barrera R. G., Estévez G. A., Giraldo J., 1985, European Journal of Physics, 6, 287

Bateman H., 1953, Higher Transcendental Functions. Bateman Manuscript Project Vol. II, McGraw-Hill, New York City, NY, US, https://authors.library.caltech.edu/records/cnd32-h9x80

Beloborodov A. M., Thompson C., 2007, ApJ, 657, 967

Bionta R. M., et al., 1987, Phys. Rev. Lett., 58, 1494

Bolton J. G., Stanley G. J., 1949, Australian Journal of Scientific Research A Physical Sciences, 2, 139

Boyd J. P., 1996, in Ilin A. V., Ridgway Scott L., eds, n Proceedings of the Third International Conference on Spectral and High Order Methods. Houston Journal of Mathematics, pp 267–276, https://public.websites.umich.edu/~jpboyd/op89icoepsf.pdf

Braithwaite J., 2009, MNRAS, 397, 763

Bratton C. B., et al., 1988, Phys. Rev. D, 37, 3361

Burke D., 2006, Toroidal coord.jpg, https://en.wikipedia.org/wiki/Toroidal_and_poloidal_coordinates#/media/File:Toroidal_coord.png

Burns E., et al., 2021, ApJL, 907, L28

Carroll B. W., Ostlie D. A., 2014, An Introduction to Modern Astrophysics: Pearson New International Edition, 2 edn. Pearson Education Ltd, Essex, GB

Chandrasekhar S., 1956, Proceedings of the National Academy of Science, 42, 1

Cheng B., Epstein R. I., Guyer R. A., Young A. C., 1996, Nature, 382, 518

Clifford N., Ransom S. M., 2019, Long-Term Timing of Pulsars in NGC 6440: An Updated Mass Limit of Millisecond Pulsar J1748-2021B (BS Thesis). University of Virginia, Charlottesville, VA, US, doi:10.18130/v3-bfjj-w888, https://doi.org/10.18130/v3-bfjj-w888

Cline T. L., et al., 1982, ApJL, 255, L45

Cline T., Frederiks D. D., Golenetskii S., Hurley K., Kouveliotou C., Mazets E., van Paradijs J., 2000, ApJ, 531, 407

Coelho J. G., Malheiro M., 2014, PASJ, 66, 14

Comella J. M., Craft H. D., Lovelace R. V. E., Sutton J. M., Tyler G. L., 1969, Nature, 221, 453

Contopoulos I., Kazanas D., Fendt C., 1999, ApJ, 511, 351

Contopoulos I., Dimitropoulos I., Ntotsikas D., Gourgouliatos K. N., 2024, Universe, 10

Courant R., Friedrichs K., Lewy H., 1967, IBM Journal of Research and Development, 11, 215

Daugherty J. K., Harding A. K., 1983, ApJ, 273, 761

Davidson P. A., 2001, An Introduction to Magnetohydrodynamics. Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, GB, https://ui.adsabs.harvard.edu/abs/2001inma.book.....D

Deutsch A. J., 1955, Annales d'Astrophysique, 18, 1

Donati J.-F., Landstreet J., 2009, ARAA, 47, 333

Dong S.-h., Lemus R., 2002, Applied Mathematics Letters, 15, 541

Doroshenko V., Santangelo A., Tsygankov S. S., Ji L., 2021, A&A, 647, A165

Duncan J. C., 1921, Proceedings of the National Academy of Sciences, 7, 179

Duncan R. C., 2000, in Kippen R. M., Mallozzi R. S., Fishman G. J., eds, American Institute of Physics Conference Series Vol. 526, Gamma-ray Bursts, 5th Huntsville Symposium. AIP, pp 830–841 (arXiv:astro-ph/0002442), doi:10.1063/1.1361651

Duncan R. C., Thompson C., 1992, ApJL, 392, L9

Fehlberg E., 1966, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 46, 1

Feroci M., Frontera F., Costa E., Amati L., Tavani M., Rapisarda M., Orlandini M., 1999, ApJL, 515, L9

Gamow G., Schoenberg M., 1941, Physical Review, 59, 539

Gelfand J. D., Gaensler B. M., 2007, ApJ, 667, 1111

Glampedakis K., Lander S. K., Andersson N., 2014, MNRAS, 437, 2

Gold T., 1968, Nature, 218, 731

Goldreich P., Julian W. H., 1969, ApJ, 157, 869

Goldreich P., Reisenegger A., 1992, ApJ, 395, 250

Golenetskii S. V., Ilinskii V. N., Mazets E. P., 1984, Nature, 307, 41

Gottlieb D., Shu C.-W., 1997, SIAM Review, 39, 644

Göğüş E., Woods P. M., Kouveliotou C., van Paradijs J., Briggs M. S., Duncan R. C., Thompson C., 1999, ApJL, 526, L93

Göğüş E., Woods P. M., Kouveliotou C., van Paradijs J., Briggs M. S., Duncan R. C., Thompson C., 2000, ApJL, 532, L121

Göğüş E., et al., 2010a, ApJ, 718, 331

Göğüş E., Woods P. M., Kouveliotou C., Kaneko Y., Gaensler B. M., Chatterjee S., 2010b, ApJ, 722, 899

Grad H., Rubin H., 1958, in Martens J. H., et al., eds, Second United Nations International Conference on the Peaceful Uses of Atomic Energy Vol. 31, Theoretical and Experimental Aspects of Controlled Nuclear Fusion. pp 190–197, https://digitallibrary.un.org/record/3808553?

Griffiths D. J., 2017, Introduction to Electrodynamics, 4 edn. Cambridge University Press, Cambridge, GB, https://ui.adsabs.harvard.edu/abs/2017inel.book.....G

Gruzinov A., 1999, arXiv e-prints, pp astro–ph/9902288

Hairer E., Nørsett S. P., Wanner G., 1993, Solving ordinary differential equations I: Nonstiff problems, 2 edn. Springer Series in Computational Mathematics, Springer Verlag, Berlin, DE, doi:10.1007/978-3-540-78862-1

Halpern J. P., Gotthelf E. V., 2010, ApJ, 725, 1384

Haverkort J. W., 2009b, Axisymmetric Ideal MHD equilibria with Toroidal Flow, http://homepage.tudelft.nl/20x40/documents/toroidalflow.pdf

Haverkort J. W., 2009a, Axisymmetric Ideal MHD Tokamak Equilibria, http://homepage.tudelft.nl/20x40/documents/Equilibria.pdf

Heger A., Fryer C. L., Woosley S. E., Langer N., Hartmann D. H., 2003, ApJ, 591, 288

Hewish A., Bell S. J., Pilkington J. D. H., Scott P. F., Collins R. A., 1968, Nature, 217, 709

Hirata K., et al., 1987, Phys. Rev. Lett., 58, 1490

Hirata K. S., et al., 1988, Phys. Rev. D, 38, 448

Hoffman J. D., 1992, Numerical Methods for Engineers and Scientists. McGraw-Hill Co., Singapore, SG

Hulleman F., van Kerkwijk M. H., Kulkarni S. R., 2000, Nature, 408, 689

Hulleman F., Tennant A. F., van Kerkwijk M. H., Kulkarni S. R., Kouveliotou C., Patel S. K., 2001, ApJL, 563, L49

Hurley K., et al., 1999, Nature, 397, 41

Hurley K., et al., 2005, Nature, 434, 1098

Ibrahim A. I., Swank J. H., Parke W., 2003, ApJL, 584, L17

Ibrahim A. I., et al., 2004, ApJL, 609, L21

Israel G. L., et al., 2003, ApJL, 589, L93

Jackson J. D., 1999, Classical Electrodynamics, 3 edn. John Wiley & Sons, Hoboken, NJ, US, https://ui.adsabs.harvard.edu/abs/1998clel.book.....J

Jiang J.-L., Tang S.-P., Wang Y.-Z., Fan Y.-Z., Wei D.-M., 2020, ApJ, 892, 55

Jones M. H., Lambourne R. J. A., Serjeant S., 2015, An Introduction to Galaxies and Cosmology, 2 edn. The Open University, Cambridge University Press, Cambridge, GB, https://ui.adsabs.harvard.edu/abs/2015igc..book.....J

Judson H. F., 2003, New York Times, 20 October

Kalapotharakos C., Kazanas D., Harding A., Contopoulos I., 2012, ApJ, 749, 2

Kargaltsev O., et al., 2012, ApJ, 748, 26

Kaspi V. M., Beloborodov A. M., 2017, ARAA, 55, 261

Kennea J. A., et al., 2013, ApJL, 770, L24

Kou F. F., Tong H., 2015, MNRAS, 450, 1990

Kouveliotou C., et al., 1994, Nature, 368, 125

Kouveliotou C., et al., 1998, Nature, 393, 235

Kouveliotou C., et al., 1999, ApJL, 510, L115

Kouveliotou C., Duncan R. C., Thompson C., 2003, Scientific American, 288, 34

Kozlova A. V., et al., 2016, MNRAS, 460, 2008

Kulkarni S. R., Frail D. A., 1993, Nature, 365, 33

Kulkarni S. R., Frail D. A., Kassim N. E., Murakami T., Vasisht G., 1994, Nature, 368, 129

Kumar H. S., Safi-Harb S., 2008, ApJL, 678, L43

Kunkel W., et al., 1987, IAU Circ., 4316, 1

Lamb R. C., Fox D. W., Macomb D. J., Prince T. A., 2002, ApJL, 574, L29

Lampland C. O., 1921, Publ. Astron. Soc. Pac., 33, 79

Lander S. K., 2016, ApJL, 824, L21

Lander S. K., 2021, MNRAS: Letters, 507, L36

Lander S. K., Jones D. I., 2009, MNRAS, 395, 2162

Lander S. K., Jones D. I., 2011a, MNRAS, 412, 1394

Lander S. K., Jones D. I., 2011b, MNRAS, 412, 1730

Levin L., et al., 2012, MNRAS, 422, 2489

Linzer N. B., Scholberg K., 2019, Phys. Rev. D, 100, 103005

Livingstone M. A., Scholz P., Kaspi V. M., Ng C. Y., Gavriil F. P., 2011, ApJL, 743, L38

Longair M. S., 2011, High Energy Astrophysics, 3 edn. Cambridge University Press, Cambridge, GB, doi:10.1017/CBO9780511778346, https://doi.org/10.1017/CBO9780511778346

Low B. C., Lou Y. Q., 1990, ApJ, 352, 343

Lower M. E., Shannon R. M., Johnston S., Bailes M., 2020, ApJL, 896, L37

Lynden-Bell D., Boily C., 1994, MNRAS, 267, 146

Lyne A. G., Pritchard R. S., Graham Smith F., 1993, MNRAS, 265, 1003

Lyutikov M., Temim T., Komissarov S., Slane P., Sironi L., Comisso L., 2019, MNRAS, 489, 2403

MacDonald D., Thorne K. S., 1982, MNRAS, 198, 345

Makarenko E. I., Igoshev A. P., Kholtygin A. F., 2021, MNRAS, 504, 5813

Manchester R. N., Hobbs G. B., Teoh A., Hobbs M., 2005, AJ, 129, 1993

Marsden D., White N. E., 2001, ApJL, 551, L155

Martin B. R., 2009, Nuclear and Particle Physics: An Introduction, 2 edn. John Wiley & Sons, West Sussex, GB

Mayall N. U., 1939, Leaflet of the Astronomical Society of the Pacific, 3, 145

Mayall N. U., Oort J. H., 1942, Publ. Astron. Soc. Pac., 54, 95

Mazets E. P., Golentskii S. V., Ilinskii V. N., Aptekar R. L., Guryan I. A., 1979, Nature, 282, 587

Mazets E. P., Cline T. L., Aptekar' R. L., Butterworth P. S., Frederiks D. D., Golenetskii S. V., Il'Inskii V. N., Pal'Shin V. D., 1999, Astronomy Letters, 25, 635

Mereghetti S., 2008, A&A Review, 15, 225

Mereghetti S., Esposito P., Tiengo A., Götz D., Israel G. L., De Luca A., 2012, A&A, 546, A30

Mereghetti S., et al., 2024, Nature, 629, 58

Michel F. C., 1973, ApJL, 180, L133

Mikić Z., Linker J. A., 1994, ApJ, 430, 898

Molkov S., Hurley K., Sunyaev R., Shtykovsky P., Revnivtsev M., Kouveliotou C., 2005, A&A, 433, L13

Muno M. P., et al., 2006, ApJL, 636, L41

Murakami T., Tanaka Y., Kulkarni S. R., Ogasaka Y., Sonobe T., Ogawara Y., Aoki T., Yoshida A., 1994, Nature, 368, 127

Nauenberg M., 1972, ApJ, 175, 417

Olausen S. A., Kaspi V. M., 2014, ApJS, 212, 6

Oppenheimer J. R., Volkoff G. M., 1939, Physical Review, 55, 374

Pacini F., 1967, Nature, 216, 567

Pacini F., 1968, Nature, 219, 145

Paczyński B., 1992, Acta Astron., 42, 145

Parfrey K., 2012, PhD thesis, Columbia University, New York City, NY, US, doi:10.7916/D8Q81M5C

Parfrey K., 2019, in KITP Conference: Connecting Micro and Macro Scales: Acceleration, Reconnection, and Dissipation in Astrophysical Plasmas. p. 21

Parfrey K., Tchekhovskoy A., 2017, ApJL, 851, L34

Parfrey K., Beloborodov A. M., Hui L., 2012, MNRAS, 423, 1416

Parfrey K., Beloborodov A. M., Hui L., 2013, ApJ, 774, 92

Park S., Hughes J. P., Slane P. O., Burrows D. N., Lee J.-J., Mori K., 2012, ApJ, 748, 117

Park S., Bhalerao J., Kargaltsev O., Slane P. O., 2020, ApJ, 894, 17

Pétri J., 2012, MNRAS, 424, 605

Rezzolla L., Most E. R., Weih L. R., 2018, ApJL, 852, L25

Romani R. W., Kandel D., Filippenko A. V., Brink T. G., Zheng W., 2022, ApJL, 934, L17

Rosswog S., Brüggen M., 2007, Introduction to High-Energy Astrophysics. Cambridge University Press, Cambridge, GB, https://ui.adsabs.harvard.edu/abs/2011ihea.book.....R

Ryan S. G., Norton A. J., 2010, Stellar Evolution and Nucleosynthesis. The Open University, Cambridge University Press, Cambridge, GB, https://www.cambridge.org/gb/universitypress/subjects/physics/astrophysics/stellar-evolution-and-nucleosynthesis?format=HB&isbn=9780521196093

Sakamoto T., et al., 2011, Advances in Space Research, 47, 1346

Salaris M., Cassisi S., 2005, Evolution of Stars and Stellar Populations. John Wiley & Sons, West Sussex, GB, https://ui.adsabs.harvard.edu/abs/2005essp.book.....S

Scharlemann E. T., Wagoner R. V., 1973, ApJ, 182, 951

Shafranov V. D., 1966, Reviews of Plasma Physics, 2, 103

Shibata K., Magara T., 2011, Living Reviews in Solar Physics, 8, 6

Shklovsky I. S., 1958, Reviews of Modern Physics, 30, 1047

Shklovsky J., 1964, Nature, 201, 588

Spitkovsky A., 2006, ApJL, 648, L51

Sturrock P. A., 1991, ApJ, 380, 655

Sukhbold T., Ertl T., Woosley S. E., Brown J. M., Janka H.-T., 2016, ApJ, 821, 38

Tam C. R., Kaspi V. M., Gaensler B. M., Gotthelf E. V., 2006, ApJ, 652, 548

Thompson C., Duncan R. C., 1993, ApJ, 408, 194

Thompson C., Duncan R. C., 1995, MNRAS, 275, 255

Thompson C., Duncan R. C., 1996, ApJ, 473, 322

Thompson C., Duncan R. C., Woods P. M., Kouveliotou C., Finger M. H., van Paradijs J., 2000, ApJ, 543, 340

Thompson C., Lyutikov M., Kulkarni S. R., 2002, ApJ, 574, 332

Thorne K. S., MacDonald D., 1982, MNRAS, 198, 339

Timokhin A. N., 2006, MNRAS, 368, 1055

Tolman R. C., 1939, Physical Review, 55, 364

Turolla R., Zane S., Watts A. L., 2015, Reports on Progress in Physics, 78, 116901

Uzdensky D. A., 2011, Space Sci. Rev., 160, 45

Vrba F. J., et al., 1996, ApJ, 468, 225

Wachter S., et al., 2004, ApJ, 615, 887

Wachter S., Ramirez-Ruiz E., Dwarkadas V. V., Kouveliotou C., Granot J., Patel S. K., Figer D., 2008, Nature, 453, 626

Wang Z., Chakrabarty D., 2002, ApJL, 579, L33

Wiegelmann T., Sakurai T., 2021, Living Reviews in Solar Physics, 18, 1

Willis D. M., Young L. R., 1987, Geophysical Journal, 89, 1011

Wolfson R., 1995, ApJ, 443, 810

Woltjer L., 1958, BAIN, 14, 39

Woltjer L., 1964, ApJ, 140, 1309

Woods P. M., et al., 1999, ApJL, 524, L55

Zhang S., 2022, Battling Gibbs Phenomenon: On Finite Element Approximations of Discontinuous Solutions of PDEs (arXiv:1907.03429), https://arxiv.org/abs/1907.03429

Zhou P., Chen Y., Li X.-D., Safi-Harb S., Mendez M., Terada Y., Sun W., Ge M.-Y., 2014, ApJL, 781, L16

van der Horst A. J., et al., 2010, ApJL, 711, L1