Introducing Latitude-Dependent Turbulent Magnetic Diffusivity in a Solar Dynamo Model

Alexandra Berg
alexandra.berg@live.com

under the direction of
Bidya Binay Karak
NORDITA (Nordic Institute for Theoretical Physics)

Research Academy for Young Scientists
July 8, 2015

Abstract

As space technology and communication can be greatly affected by solar activity, it is of importance to understand the phenomenas' underlying cause; solar magnetodynamics. Several models aiming to describe solar magnetodynamics currently exist, of which one is the flux transport dynamo model. This model and its two constituent equations form the basis of SURYA, a Fortran program aimed at modeling the 2D spaciotemporal evolution of the solar magnetic fields. In this study, the toroidal turbulent magnetic diffusivity, a variable present in one of the two flux transport dynamo equations, is modified to be latitudinally dependent. Results indicate that such a program modification yields more accurate and representative magnetic field data, albeit at a considerably greater run time than the original, unmodified SURYA program. The results also highlight the importance of considering toroidal turbulent diffusivity as intrinsically dependent upon latitude.

Contents

1 Introduction

Understanding the sun's magnetodynamics provides vital insight into numerous secondary solar activities, such as coronal mass ejection, solar flares and solar storms [1]. A detailed understanding of these phenomena is not only of theoretical value to the field of astrophysics, but also of high practical value to fields of space technology and communication [1, 2].

Several theoretical models of the sun's magnetodynamics already exist; within one subset of solar models, the kinematic models, is the so-called flux transport dynamo model [3]. This model is the basis of SURYA, a Fortran program aimed at modeling the 2D spaciotemporal evolution of the large-scale magnetic solar fields. Certain approximations and simplifications of the variables involved are made in the SURYA program; one such affected variable is the toroidal turbulent magnetic diffusivity [4].

The aim of this study is firstly to develop the original SURYA program by making this variable, the toroidal turbulent magnetic diffusivity, dependent upon the latitudinal position θ. Secondly, the aim is to investigate subsequent effects on the modeling of the large-scale solar magnetic field with the modification.

2 Theoretical Overview

2.1 Observation

First regularly documented in 1749 by the Zürich Observatory, sunspots are one of the most extensively catalogued solar phenomena, making them one of the primary sources of our knowledge on the sun's magnetodynamics [2].

Sunspot cycle periods span approximately eleven years, during which sunspots first appear at a latitude of forty degrees N/S, and then migrate equatorwards to disappear at the period's end. Sunspots often appear in tandem on the same hemisphere, where members of a pair have opposite polarities; they are essentially coupled areas where magnetic fields enter and exit, respectively. After each sunspot cycle period, the polarity is reversed within the sunspot pairs [1, 2, 5].

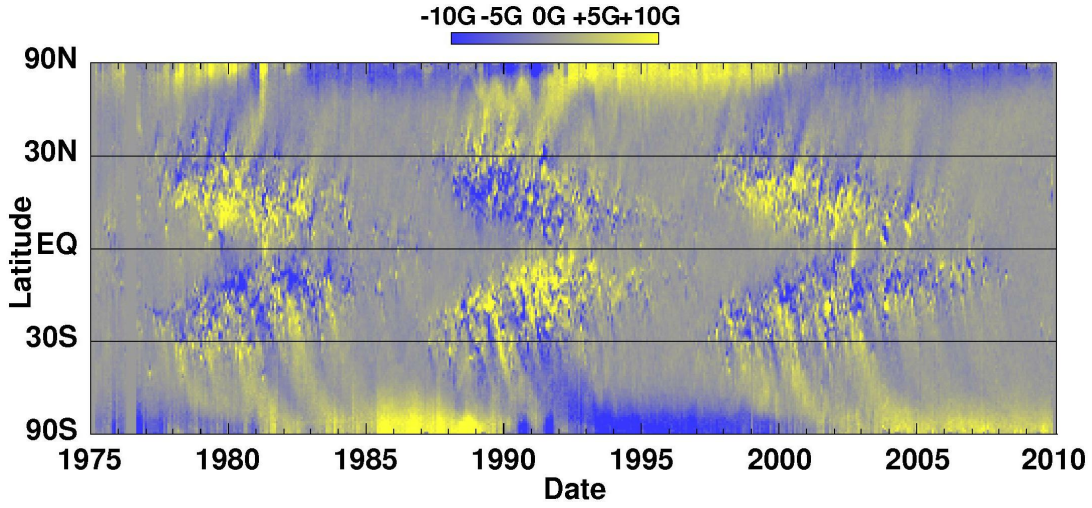


Figure 1: Magnetic fields of the sunspot cycle

When the sun was discovered to be prone to magnetodynamic activity, it was quickly extrapolated that the sunspots and sunspot cycles were direct consequences of a greater cycle. This cycle, the magnetic cycle, spans 22 years during which the involved magnetic fields are continuously generated and plied through a so-called dynamo process [1, 2].

2.2 Hydromagnetic Dynamo Theory

Magnetic fields are initially generated as a result of plasma movement. As quasi-neutral, plasma contains equal amounts of positively and negatively charged par-

ticles per volume unit; its magnetic state is therefor initially $\mathbf{B} = 0$. However, due to small fluctuations in the charge distribution, so-called magnetic seed fields are formed, which in turn give rise to large scale magnetic fields [5]. This process is primarily governed by the two magnetohydrodynamic (MHD) equations known as the induction equation,

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) - \nabla \times \eta(\nabla \times \mathbf{B}) \tag{1}$$

in which $\mathbf{B}$ is the magnetic field, $\mathbf{v}$ is the flow velocity and η is the turbulent magnetic diffusivity, and the motion equation. From the induction equation (see Equation 1) it is clear that if no fluid motion is present ($\mathbf{v} = 0$), then the magnetic field is not actively sustained; it will decay, albeit over great timescales [5].

It is therefore inferable that the solar magnetism, which has lasted over considerable time scales, must be continuously generated. The theory that describes both the initial and the continuous generation, the hydromagnetic dynamo theory, is based on the motion equation and the induction equation. However, depending on to what extent one chooses to consider either equation, the hydromagnetic dynamo theory is divided into two separate areas [5, 6].

The first area entails non-linear dynamos, in which both equations are considered. There is therefore a loop of affect between the magnetic fields' Lorentz forces and the fluid velocities. In the second area however, known as kinematic dynamos, the motion equation is neglected. Plasma flow velocities are considered predetermined and virtually unaffected by the magnetic fields [6, 7].

Non-linear dynamos yield coupled non-linear equations, which are computationally difficult to solve, whilst kinematic dynamos yield simpler, more readily

solvable linear equations [7]. Despite the solar magnetic dynamo process being an intrinsically non-linear phenomenon, only kinematic dynamo models will be considered in this study.

2.3 Kinematic Dynamos and Magnetic Cycle Generation

In keeping to kinematic dynamos only, i.e. assuming that all involved fluid velocities are given, one can now proceed to explain the continuous generation of the magnetic fields and cycle. The total solar magnetic field can be considered to be composed of a poloidal and toroidal magnetic field:

$$\mathbf{B} = \mathbf{B}_t + \mathbf{B}_p$$

were the toroidal field $\mathbf{B}_t = B_\phi \mathbf{e}_\phi$ lies in the azimuthal direction, and the poloidal field $\mathbf{B}_p = B_r \mathbf{e}_r + B_\theta \mathbf{e}_\theta = \nabla \times [A(r,\theta)\mathbf{e}_\phi]$ (where A is the poloidal field vector component) lies in the poleward direction [5].

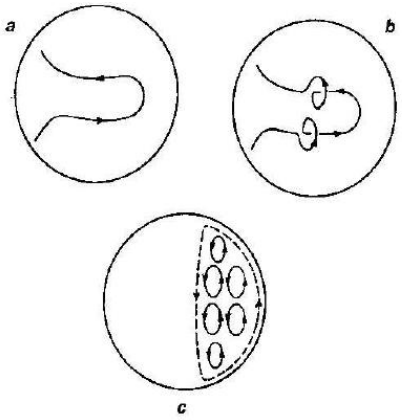


Figure 2: Half a magnetic cycle; the generation of poloidal and toroidal fields

According to Parker's turbulent dynamo theory [5], which delineates the continuous generation of magnetic fields and cycles, the first step of cycle generation

involves the global poloidal fields generated earlier (see Section 2.2). The field lines of these global poloidal fields are essentially dragged by the solar differential rotation, producing toroidal fields that move anti-parallel to each other in either hemisphere (see Subfigure 2a). This process of magnetic field generation powered by the solar differential rotation is known as the omega effect [5].

The second step in continuous magnetic field generation involves the turbulent convective motions of the solar plasma. Due to the Coriolis effect, rising and sinking plasma turbulences start rotating, resulting in helical turbulence.The helical turbulence twists the toroidal field lines, which (due to the toroidal fields' anti-parallel nature and the chirality of the helical turbulence) gives rise to small poloidal fields with same magnetic sense (see Subfigure 2b). This process of magnetic field generation powered by the helical turbulence is known as the alpha effect. The small poloidal fields eventually combine to give rise to global poloidal fields, with opposite magnetic sense to the initial global poloidal field (see Subfigure 2c). Thus, half a magnetic cycle has passed [5].

One can through the steps of half a magnetic cycle explain the sunspot cycle process. During the initial phase of the magnetic cycle, very few small poloidal fields exist. As sunspots are essentially the entries and exits of the small poloidal magnetic fields, there are consequently few sunspots present. However, as helical turbulence gives rise to small poloidal fields, sunspots are formed; the sunspot cycle reaches a maximum. Meanwhile, the solar differential rotation drags the sunspot pairs, producing the characteristic slanted equatorward migration (see Figure 1). As the small poloidal fields combine into global poloidal fields, the sunspot cycle reaches yet another minimum, where after the hemispherical and sunspot polarities are reversed in concordance to the new global poloidal field direction [2, 5].

2.4 Flux transport dynamo model

One kinematic dynamo model is the flux transport dynamo, which models the spaciotemporal evolution of the sun's large scale magnetic fields [3, 7]. More specifically, the flux transport dynamo model is characterized by assumptions pertaining to the meridional plasma flow. Firstly, the model assumes that the equatorward sunspot movement is driven by a meridional flow in the sun. Secondly, it assumes that the magnetic cycle's period is primarily determined by the velocity of the meridional flow [3, 6]. The flux transport dynamo model also accounts for the turbulent magnetic diffusivity, considered specific to the toroidal and poloidal fields respectively [3, 4, 6].

In essence, the flux transport dynamo model is summarized through the following two equations describing the change in the poloidal and toroidal fields' vector components A and B over time.

$$\frac{\partial \mathbf{A}}{\partial t} + \frac{1}{r\sin\theta}(\mathbf{v}\cdot\nabla)(r\sin\theta\mathbf{A}) = \eta_p(\nabla^2 - \frac{1}{r\sin\theta})\mathbf{A} + \alpha\mathbf{B} \tag{2}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \frac{1}{r}[\frac{\partial(r\mathbf{v}_r\mathbf{B})}{\partial r} + \frac{\partial(v_\theta\mathbf{B})}{\partial\theta}] = \eta_t(\nabla^2 - \frac{1}{r\sin\theta})\mathbf{B} + r\sin\theta(\mathbf{B}_p\cdot\nabla)\Omega + \frac{1}{r}\frac{\mathrm{d}\eta_t}{\mathrm{d}r}\frac{\partial(r\mathbf{B})}{\partial r} \tag{3}$$

The terms $\frac{1}{r\sin\theta}(\mathbf{v}\cdot\nabla)(r\sin\theta\mathbf{A})$ and $\frac{1}{r}[\frac{\partial(r\mathbf{v}_r\mathbf{B})}{\partial r} + \frac{\partial(v_\theta\mathbf{B})}{\partial\theta}]$ are the meridional flow transport terms of the respective magnetic fields, $\alpha\mathbf{B}$ and $r\sin\theta(\mathbf{B}_p\cdot\nabla)\Omega$ the source terms for alpha- and omega- effects on the respective magnetic fields, and $\eta_p(\nabla^2 - \frac{1}{r\sin\theta})\mathbf{A}$ and $\eta_t(\nabla^2 - \frac{1}{r\sin\theta})\mathbf{B}$ the terms corresponding to the effects of turbulent magnetic diffusion (η_t and η_p) in either magnetic field. The term $\frac{1}{r}\frac{d\eta_t}{dr}\frac{\partial(r\mathbf{B})}{\partial r}$

represents how toroidal turbulent magnetic diffusivity η_t varies with depth r, the only dimension on which η_t is currently assumed dependent upon [3, 4].

2.5 SURYA Program and Modifications

The SURYA program is a Fortran code aimed at simulating the solar magnetic activity in two dimensions. As it is based entirely on the flux transport dynamo model, it assumes a kinematic solar dynamo process and accounts for meridional circulation (see section 2.4) [4].

In the original SURYA code, the evolution of the toroidal and poloidal fields is defined according to the flux transport dynamo equations; 2 and 3. In the case of the toroidal magnetic fields, the corresponding turbulent magnetic diffusivity η_t is defined as

$$\eta_{t(old)}(r) = \eta_{RZ} + \frac{\eta_{SCZ1}}{2}[1 + \mathrm{erf}(\frac{r - r'_{BCZ}}{d_t})] + \frac{\eta_{SCZ}}{2}[1 + \mathrm{erf}(\frac{r - r_{TCZ}}{d_t})] \qquad (4)$$

were η_{RZ}, η_{SCZ1} and η_{SCZ} are pre-defined parameter values [4] of the turbulent magnetic diffusivity at different areas of the sun. r'_{BCZ} and r_{TCZ} are in turn the radii corresponding to the above mentioned areas, and d_t is the time step of the program [4].

Prior to this study, a currently unpublished investigation by Bidya Binay Karak determined the numerical treatments necessary to make the current toroidal turbulent diffusivity η_t(r) r- and θ-dependent. It was determined that the relation between $\eta_{t(new)}(r, \theta)$ and $\eta_{t(old)}(r)$ is as follows

$$\eta_{t(new)}(r,\theta) = \frac{\eta_{t(old)(r)}}{1+\mathbf{B}^2} \tag{5}$$

were $\mathbf{B}$ is the toroidal magnetic field. It was also determined that in order to account for the new θ-dependency of the toroidal turbulent diffusivity, the equation governing the toroidal field evolution would have the following additional term.

$$\frac{1}{r^2\sin\theta}\frac{\partial\eta_t}{\partial\theta}\frac{\partial(\sin\theta\mathbf{B})}{\partial\theta} \tag{6}$$

Hence, the necessary modifications of the SURYA code needed to make the toroidal turbulent diffusivity r- and θ-dependent (the aim of this study) entail the redefinition of η_t

$$\eta_{t(new)}(r,\theta) = \frac{\eta_{RZ} + \frac{\eta_{SCZ1}}{2}[1+\mathrm{erf}(\frac{r-r'_{BCZ}}{d_t})] + \frac{\eta_{SCZ}}{2}[1+\mathrm{erf}(\frac{r-r_{TCZ}}{d_t})]}{1+\mathbf{B}^2} \tag{7}$$

and the modification of the toroidal field evolution

$$\frac{\partial\mathbf{B}}{\partial t} + \frac{1}{r}[\frac{\partial(r\mathbf{v}_r\mathbf{B})}{\partial r} + \frac{\partial(v_\theta\mathbf{B})}{\partial\theta}] = \tag{8}$$
$$\eta_t(\nabla^2 - \frac{1}{r\sin\theta})\mathbf{B} + r\sin\theta(\mathbf{B}_p\cdot\nabla)\Omega + \frac{1}{r}\frac{\mathrm{d}\eta_t}{\mathrm{d}r}\frac{\partial(r\mathbf{B})}{\partial r} + \frac{1}{r^2\sin\theta}\frac{\partial\eta_t}{\partial\theta}\frac{\partial(\sin\theta\mathbf{B})}{\partial\theta}$$

as shown above. Subsequent, indirect modifications to the SURYA code involve structural changes to accommodate changes in numerical and temporal dependencies of various variables.

3 Method

3.1 Mathematical Procedure

The mathematical procedures conducted in this study revolve around the adaption and integration of the previously determined expressions 7 and 8 for r- and θ-dependent toroidal diffusivity into the program framework of SURYA. Firstly, the new expression 8 for the toroidal magnetic field's evolution is converted to spherical coordinate form. The resulting equation is then solved on the form

$$\frac{\partial \mathbf{B}}{\partial t} = [N_r + N_\theta]\mathbf{B} + Q \tag{9}$$

were N_r and N_θ represent the r- and θ-dependent terms respectively, and Q the omega source term $r \sin\theta(\mathbf{B}_p \cdot \nabla)\Omega$. Secondly, the resulting terms are converted through either previously established difference schemes [2] or, in the case of new terms, difference schemes selected in this study. The resulting equations are organized on the form

$$\frac{-\triangle t}{2} N_r B^m_{i,j} = a(i,j)B^m_{i-1,j} + b(i,j)B^m_{i,j} + c(i,j)B^m_{i+1,j} \tag{10}$$

$$\frac{-\triangle t}{2} N_\theta B^m_{i,j} = d(i,j)B^m_{i-1,j} + e(i,j)B^m_{i,j} + f(i,j)B^m_{i+1,j} \tag{11}$$

were the coefficients $a(i,j)$ through $f(i,j)$ constitute the new SURYA code matrix equation coefficients of the toroidal magnetic field. See appendix A for explicit details.

3.2 Programming Procedure

Direct changes to pre-existing code consist of modifications to the toroidal fields matrix coefficient block, and the program definition of toroidal diffusivity. Structural changes to the SURYA code include remodeling the program to accommodate the direct or indirect time dependencies of the modified matrix coefficients, toroidal diffusivity and diffusivity-dependent functions. See appendix B for explicit details.

3.3 Analytical Procedure

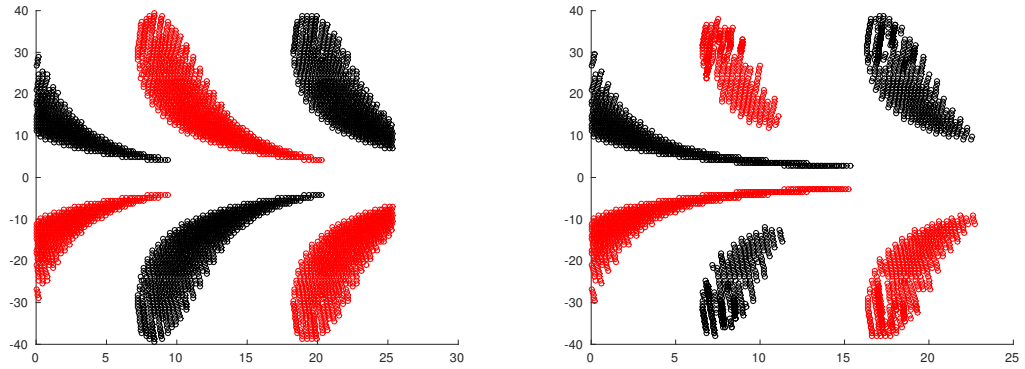
A comparative study of the modified SURYA code in relation to the original code will be conducted, where the two programs are run with the same initial data and identical settings. The comparative procedure with which the results will be analyzed consists of two separate factions.

The first analysis will judge if the results of the modified SURYA code meet the elementary criteria of solar dynamo models, i.e. if the results accurately portray the basic documented behavior of sunspot cycles. The phenomena checked for include an eleven year sunspot cycle period, a tilted equatorward drift of sunspots over time and cyclic polarity reversal.

The second, perhaps less trivial analysis has as purpose to judge the relative quality of the modified SURYA code, in terms of overall performance.

4 Results

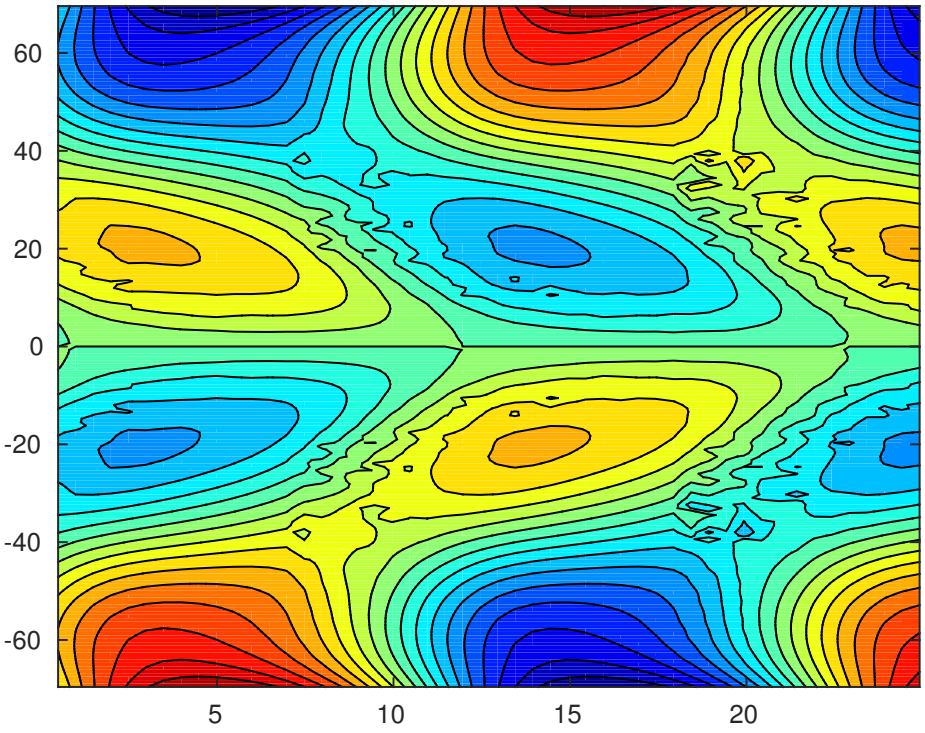
The data shown is from the original and modified SURYA codes regarding the spaciotemporal evolution of the solar magnetic fields and sunspots.



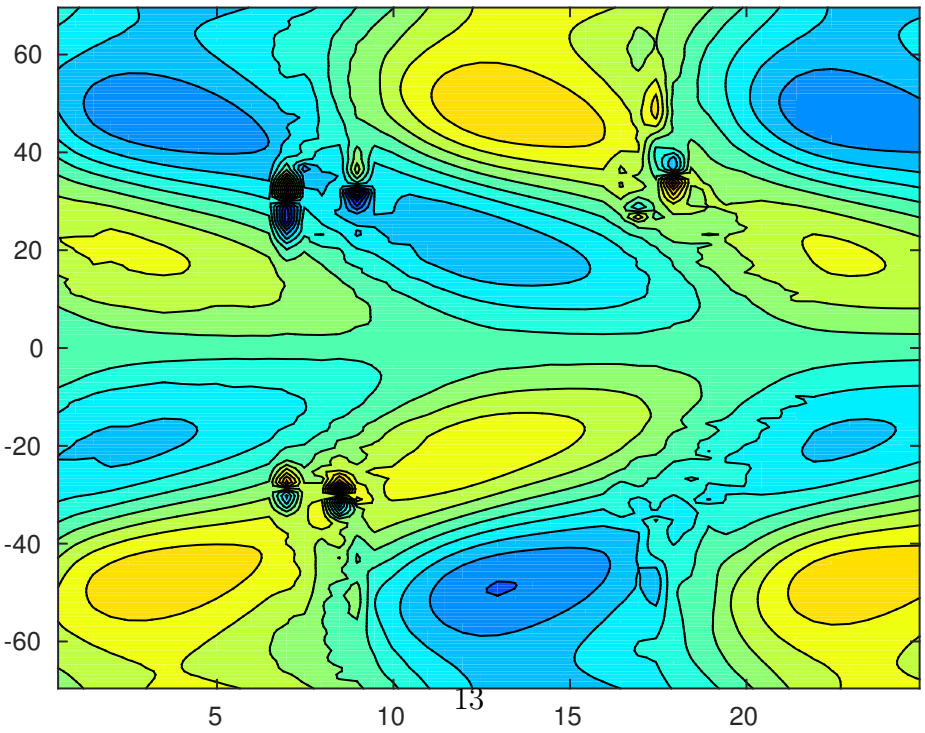
(a) Original SURYA code (b) Modified SURYA code

Figure 3: Theoretical sunspot eruptions; red corresponds to negative polarity

Figure 3 shows the appearance of sunspots, were red and black correspond to different hemispherical polarities, at different latitudes as a function of time in years. Figure 4 shows the magnetic fields, were the color scheme and intensity corresponds to magnetic polarity and relative strength, over the latitudes as a function of time in years.



(a) Original SURYA code



(b) Modified SURYA code

Figure 4: Theoretical radial magnetic fields

5 Discussion

5.1 Result Analysis

In terms of fulfilling the basic criteria imposed on any solar dynamo model, the modified SURYA code overall succeeded. The subfigure 3b seems ambiguous in sunspot period length, as the first period (15 years) seems to compound with the second period (five years) to yield two periods of appropriate lengths. However, the subfigure 4b, which models the magnetic fields that correspond to the sunspot cycles, gives correct period lengths of approximately eleven years. Additionally, both subfigures of the modified SURYA code show clear tilted equatorward migrations of sunspots over time. Finally, hemispherical and sunspot polarity reversal is observed quite clearly in both graphs, with a magnetic cycle spanning 22 years. In conclusion, the modified SURYA code is deemed acceptable as a solar dynamo model.

Further comparison of the original and modified SURYA code yield interesting observations. Firstly, one of the most prominent features of the modified SURYA code is the relative weakness of the magnetic fields, as seen when comparing Subfigures 4a and 4b. This is also coupled with a shift in the central focus of the polar regions, making the total magnetic regions of the each sunspot period (the combined blue and yellow area per 11 year period) approach a more centered, slanted and "sunspot"-like appearance in comparison to the original model. Secondly, there is a very interesting break in symmetry; as seen in Figure 4, the original SURYA code yields highly or completely symmetrical models, whilst the modified SURYA code yields more complex hemispherical representations. Thirdly, as an extension of the previous point, the turbulent behaviors exhibited by Figure 4 dif-

fer markedly. Whilst the original SURYA code yields slight disturbances between each polarity shift, the turbulences at the corresponding points in the modified SURYA code converge into strongly focused, organized secondary magnetic fields. These features, the relatively weaker, more centralized and slanted magnetic fields, the break in symmetry and the appearance of secondary magnetic fields combine to form a more complex and representative picture of solar activity. Thus, in certain aspects, the modified SURYA code is deemed more accurate than the original program.

5.2 Flaws and Considerations

Despite the positive results, their are certain issues to address. The modified SURYA code was found to be computationally intensive, with a run time 10 to 14 times greater than the original code. To what extent this was due to improperly modified code or some fundamental issue pertaining to increased time dependency of several variables is undetermined in this study. However, it is still necessary to discuss the relations between the gains in accuracy and loss of efficiency. Although enhanced representativity is always sought after when producing models, one must weigh the gains in representativity and accuracy with the loss in computational efficiency. In the case of the modified SURYA code, it is unlikely that introducing θ-dependent toroidal turbulent diffusivity improved the program usability: as based on the simpler branch of hydromagnetic dynamo models (the kinematic dynamo models) the purpose of the SURYA code is to provide an environment for quick trials of flux transport dynamo models. However, one must remember that the results of the modifications provided further evidence of the importance of

θ-dependent turbulent diffusivity in kinematic solar dynamo models, which on is a positive result on its own.

5.3 Further Questions

A question worthy of investigation pertains to the above mentioned dilemma of increased time dependency: to what extent is the increased run time of modified SURYA code a theoretical, rather than a practical problem? Furthermore, it is of interest to investigate extended dimensional dependency of other variables, both for the sake of improving the SURYA program and for the sake of investigating the nature of these variables.

6 Acknowledgements

I would like to thank my mentor Bidya Binay Karak, who with great kindness, support and patience guided me through this project. I would also like to thank Alfred Isaac, my travel partner through both the Stockholm traffic and the world of solar physics. Finally, I owe great thanks to Philip Frick, Anna Broms, Serhat Aktay, Sofia Svensson and the entirety of the Ray's program who made this project, and many others, possible in the first place.

References

[1] Hathaway, D.H. *Solar Cycle Prediction.* Nasa.gov (2015).

[2] Hathaway, D.H. *The Solar Cycle* Living Rev. Solar Phys. (2005)

[3] Charbonneau, P. *Flux Transport Dynamos* Scholarpedia, 2(9):3440. (2007)

[4] Choudhuri, A.R. *The User's Guide to the Solar Dynamo Code SURYA*. (2005)

[5] Choudhuri, A.R. *The Physics of Fluids and Plasmas; an Introduction for Astrophysicist* (1998)

[6] Charbonneau, P. *Dynamo Models of the Solar Cycle* Living Rev. Solar Phys. (2010)

[7] Brandenburg, A. *Hydromagnetic Dynamo Theory* Scholarpedia, 2(3):2309 (2007)

[8] Chan, T., Ingemyr, M., Winn, J. N., Holman, M. J., Sanchis-Ojeda, R., Esquerdo, G., Everett, M. *The Transit Light Curve project. XIV. Confirmation of Anomalous Radii for the Exoplanets TrES-4b, HAT-P-3b, and WASP-12b.* The Astrophysical Journal, Volume 141, Issue 6, article id. 179 (2011).

A Mathematical Derivation

As a preliminary step to converting the modified $\frac{\partial \mathbf{B}}{\partial t}$ equation to a form suitable for the SURYA program, the equation is solved on the form

$$\frac{\partial \mathbf{B}}{\partial t} = [N_r + N_\theta]\mathbf{B} + Q \tag{12}$$

where N_r and N_θ correspond to r- and θ-dependent terms respectively, and Q to $r\sin\theta(\mathbf{B}_p \cdot \nabla)\Omega$. First, the original modified equation

$$\begin{aligned}\frac{\partial \mathbf{B}}{\partial t} = -\frac{1}{r}[\frac{\partial(r\mathbf{v}_r\mathbf{B})}{\partial r} + \frac{\partial(v_\theta \mathbf{B})}{\partial \theta}] + \eta_t\left(\nabla^2 - \frac{1}{r\sin\theta}\right)\mathbf{B} + \\ r\sin\theta(\mathbf{B}_p \cdot \nabla)\Omega + \frac{1}{r}\frac{\mathrm{d}\eta_t}{\mathrm{d}r}\frac{\partial(r\mathbf{B})}{\partial r} + \frac{1}{r^2\sin\theta}\frac{\partial\eta_t}{\partial\theta}\frac{\partial(\sin\theta\mathbf{B})}{\partial\theta}\end{aligned} \tag{13}$$

is written in spherical coordinates

$$\begin{aligned}\frac{\partial \mathbf{B}}{\partial t} = -\frac{1}{r}\left[\frac{\partial(r\mathbf{v}_r\mathbf{B})}{\partial r} + \frac{\partial(v_\theta\mathbf{B})}{\partial\theta}\right] + \eta_t[\frac{1}{r^2}\frac{\partial}{\partial r}\{r^2\frac{\partial\mathbf{B}}{\partial r}\} + \\ \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\{\sin\theta\frac{\partial\mathbf{B}}{\partial\theta}\} + \frac{1}{r^2\sin^2\theta}\{\frac{\partial^2\mathbf{B}}{\partial\phi^2}\} - \frac{\mathbf{B}}{r^2\sin^2\theta}] + \\ \frac{1}{r}\frac{\mathrm{d}\eta_t}{\mathrm{d}r}\frac{\partial(r\mathbf{B})}{\partial r} + \frac{1}{r^2\sin\theta}\frac{\partial\eta_t}{\partial\theta}\frac{\partial(\sin\theta\mathbf{B})}{\partial\theta} + Q\end{aligned} \tag{14}$$

and then expanded and restructured

$$\frac{\partial \mathbf{B}}{\partial t} = -\frac{1}{r}[\frac{\partial (r\mathbf{v}_r\mathbf{B})}{\partial r}] + \eta_t[\frac{1}{r^2}\{2r\frac{\partial \mathbf{B}}{\partial r} + r^2\frac{\partial^2 \mathbf{B}}{\partial r^2}\}] - \eta_t[\frac{\mathbf{B}}{2r^2\sin^2\theta}]+$$

$$\frac{1}{r}\frac{\mathrm{d}\eta_t}{\mathrm{d}r}\frac{\partial (r\mathbf{B})}{\partial r} - \frac{1}{r}[\frac{\partial (v_\theta\mathbf{B})}{\partial \theta}] + \eta_t[\frac{1}{r^2\sin\theta}\{\cos\theta\frac{\partial \mathbf{B}}{\partial \theta} + \sin\theta\frac{\partial^2 \mathbf{B}}{\partial \theta^2}\}]- \tag{15}$$

$$\eta_t[\frac{\mathbf{B}}{2r^2\sin^2\theta}] + [\frac{1}{r^2\sin\theta}\frac{\partial \eta_t}{\partial \theta}\{\cos\theta\mathbf{B} + \sin\theta\frac{\partial \mathbf{B}}{\partial \theta}\}] + Q$$

as shown above. Further simplifications yield $\mathrm{N_r}$ and N_θ

$$\begin{aligned}\mathrm{N}_\theta\mathbf{B} = -\frac{1}{r}\frac{\partial (\mathrm{V}_\theta\mathbf{B})}{\partial \theta} + \eta_t\frac{\cot\theta}{r^2}\frac{\partial \mathbf{B}}{\partial \theta} + \eta_t\frac{1}{r^2}\frac{\partial^2 \mathbf{B}}{\partial \theta^2} - \eta_t\frac{\mathbf{B}}{2r^2\sin^2\theta} + \mathbf{B}\frac{\cot\theta}{r^2}\frac{\partial \eta_t}{\partial \theta} \\ +\frac{1}{r^2}\frac{\partial \mathbf{B}}{\partial \theta}\frac{\partial \eta_t}{\partial \theta}\end{aligned} \tag{16}$$

respectively. ($\mathrm{N_r}$ is ommited from the appendix as it is identical to the $\mathrm{N_r}$ term of the original $\frac{\partial \mathbf{B}}{\partial t}$ equation, and hence inconsequential in program modifications.)

B Difference Schemes

As the final step in converting the modified $\frac{\partial \mathbf{B}}{\partial t}$ equation to a form suitable for the SURYA program, the terms of equation 16 derived in appendix A are converted to corresponding difference equations.

The advection term $-\frac{1}{r}\frac{\partial(\mathrm{V}_\theta \mathbf{B})}{\partial \theta}$ is converted through the Lax-Wendroff difference scheme [2], and the r-dependent diffusion terms through the Crank Nicholsen method [2]. The θ-dependant diffusion terms are also treated via the Crank Nicholsen method due to their strucural equivalency to the r-dependent diffusion terms.The resulting equation

$$N_\theta B = \frac{\eta_t}{r^2}[\frac{B^m_{j+1} + B^m_{j-1} - 2B^m_j}{\triangle\theta^2}] + \frac{\eta_t \cot\theta}{r^2}[\frac{B^m_{j+1} - B^m_{j-1}}{2 \triangle \theta}] - \eta_t \frac{B_j}{2r^2 \sin^2\theta} -$$

$$-\frac{1}{\triangle\theta} v_\theta(i, j-\frac{1}{2}) \sin(\theta + \frac{\triangle\theta}{2})[\frac{B^m_{j+1}}{2} + \frac{B^m_j}{2} - \frac{\triangle t}{4 \triangle \theta} + \tag{17}$$

$$(v_\theta(i, j+1) \sin(\theta + \triangle\theta) B^m_{j+1}) - v_\theta(i, j) \sin(\theta) B^m_j] + \frac{1}{r^2}\frac{\partial \eta_t}{\partial \theta}[\frac{B^m_{j+1} - B^m_{j-1}}{2 \triangle \theta}] +$$

$$\frac{\partial \eta_t}{\partial \theta}\frac{\cot\theta}{r^2}$$

is then solved and written on the form

$$-\frac{2}{\triangle t} N_\theta B = a(i,j) B^m_{j-1,i} + b(i,j) B^m_{j,i} + (i,j) B^m_{j+1,i} \tag{18}$$

were $a(i,j)B^m_{j-1,i}$, $b(i,j)B^m_{j,i}$ and $c(i,j)B^m_{j+1,i}$ are the SURYA code matrix coefficients for the N_θ term of the modified $\frac{\partial \mathbf{B}}{\partial t}$ equation. Upon comparison with the original $\frac{\partial \mathbf{B}}{\partial t}$ equation's N_θ term [2],

$$N_\theta B = \frac{\eta_t}{r^2}[\frac{B^m_{j+1} + B^m_{j-1} - 2B^m_j}{\triangle\theta^2}] + \frac{\eta_t \cot\theta}{r^2}[\frac{B^m_{j+1} - B^m_{j-1}}{2 \triangle \theta}] - \eta_t \frac{B_j}{2r^2 \sin^2\theta} -$$

$$-\frac{1}{\triangle\theta} v_\theta(i, j-\frac{1}{2}) \sin(\theta + \frac{\triangle\theta}{2})[\frac{B^m_{j+1}}{2} + \frac{B^m_j}{2} - \frac{\triangle t}{4 \triangle \theta} + \tag{19}$$

$$(v_\theta(i, j+1) \sin(\theta + \triangle\theta) B^m_{j+1}) - v_\theta(i,j) \sin(\theta) B^m_j]$$

we can deduce that the modifications to the SURYA code's original matrix coefficients only consitute of adding new terms to the pre-existing ones. The new terms are extrapolated to be as follows;

$$-\frac{\partial \eta_t}{\partial \theta} \frac{1}{r^2 2 \triangle \theta} \tag{20}$$

for the $a(i,j)B^m_{j-1,i}$ matrix coefficient,

$$\frac{\partial \eta_t}{\partial \theta} \frac{\cot\theta}{r^2} \tag{21}$$

for the $b(i,j)B^m_{j,i}$ matrix coefficient, and

$$\frac{\partial \eta_t}{\partial \theta} \frac{1}{r^2 2 \triangle \theta} \tag{22}$$

for the $c(i,j)B^m_{j+1,i}$ matrix coefficient. Thus, we have not only converted the modi-

fied equations to their corresponding difference schemes, but also extrapolated the exact terms needed for the SURYA code modification.

C Modified SURYA Code

Due to the extensive nature of the original SURYA code, only directly modified subsets (PART 0, II and V) of the code are included in this appendix. The complete code, along with a guide, is available upon request from the original creator Choudhuri [2].Guide lines specific to the modifications are available in the code listed below in the form of comments beginning with (AB98), but these assume that the reader has some prior experience with program modifications.

```
C          PART 0
implicit real*8(a-h,o-z)
parameter (nmax=257,lmax=96)
common u(nmax,nmax),t,it
common/lmx/l
common /plsincos/pl(nmax,lmax),sn(nmax)
double precision a(nmax,nmax),b(nmax,nmax),c(nmax,nmax),
& d(nmax,nmax),fun(nmax),uint(nmax),
&  e(nmax,nmax),f(nmax,nmax),vp(2*nmax,2*nmax),vq(2*nmax,2*nmax)
&  ,a1(nmax),b1(nmax),c1(nmax),r(nmax),phi(nmax,nmax),
&  phib(nmax,nmax),oldu(nmax,nmax),ss1(nmax),uub(nmax),
&  uu(nmax),al(nmax,nmax),dom(nmax,nmax),ub(nmax,nmax),ra(nmax)
double precision ab(nmax,nmax),bb(nmax,nmax),cb(nmax,nmax),
&  db(nmax,nmax),eb(nmax,nmax),fb(nmax,nmax),eta(nmax,nmax),
&  dror(nmax,nmax),drot(nmax,nmax),vp1(2*nmax,2*nmax),
&  vq1(2*nmax,2*nmax),psi(2*nmax,2*nmax),etab(nmax,nmax),
```

```
&   ss1_p(nmax),etab2(2*nmax,2*nmax),dvp(2*nmax,2*nmax),
&   vpb(2*nmax,2*nmax),deta(2*nmax,2*nmax),uin(nmax,nmax),
&   detaq(2*nmax,2*nmax)
external ss,erf
C         (AB98, 3)
C         NOTE 1:          Added detaq(2*nmax, 2*nmax)
n=nmax-1
pi=4.0d0*atan(1.0d0)

PART II.  This is the part where the alpha coefficient,
C         diffusivity, differential rotation and meridional circulation
C         are specified. Level II Users wishing to make changes in
C         this part should read Sect. 4 of the 'Guide'.


pm=6.96d0
pb=0.55d0*pm
pw=2.5*pm
qm=pi
dp=(pm-pb)/float(n)
dq=-qm/float(n)
ita=int((0.7d0-0.55d0)*pm/dp)+1
fac=1.0d8/(3600.*24*365)

C         Here begin the do loops to calculate profiles of alpha,
C         diffusivity and differential rotation at all the grid
```

```
C        points. The differential rotation is written in the file
C        'diffrot.dat'.

open(25,file='omega.dat',status='unknown',access='append')
do i = 1, nmax
ra(i)=pb+float(i-1)/float(n)*(pm-pb)
p=pb+float(i-1)/float(n)*(pm-pb)
do j = 1, nmax
q=qm-float(j-1)/float(n)*qm
co = dcos(q)

C ALPHA PROFILE
al(i,j) = 0.25d0*co*(1.00d0+
&        erf((p-0.95d0*pm)/(0.030d0*pm)))*(1.00d0-erf((p
&        -pm)/(0.030d0*pm)))

C DIFFUSIVITY PROFILES
eta(i,j) = 0.000220d0 + (et0/2.00d0)*(1.0d0 +
&        erf((p-0.7d0*pm)/(0.030d0*pm)))
!            etab(i,j) = 0.00022d0 + (et1/2.00d0)*(1.0d0 +
!     &          erf((p-0.725d0*pm)/(0.030d0*pm)))+ (et0/2.0d0)
!     &          *(1.0d0+erf((p-0.975d0*pm)/(0.030d0*pm)))

C SOLAR DIFFERENTIAL ROTATION PROFILE
```

```
dom(i,j) = 271.9d0 + 0.50d0*(1.000d0 +
&       erf((p-0.7d0*pm)/(0.030d0*pm)))*(289.5d0 -
&       39.4d0*co*co - 42.2d0*co*co*co*co -
&       271.9d0)

write(25,27)q,p,dom(i,j)
27      format(3(f13.5,1x))

end do
end do

close(25)
C       The do loops are closed. We also need the diffusivity at
C       mid-points in the grid for some calculations. This is
C       obtained and stored now.

!       do i=1,2*n+1
!            p=pb+float(i-1)*(pm-pb)/float(2*n)
!       do j=1,2*n+1
!            etab2(i,j) = 0.00022d0 + (et1/2.00d0)*(1.0d0 +
!     &         erf((p-0.725d0*pm)/(0.030d0*pm)))+ (et0/2.0d0)
!     &         *(1.0d0+erf((p-0.975d0*pm)/(0.030d0*pm)))
!       end do
!       end do
```

```
C       Derivatives of differential rotation are obtained and stored
C       now for future use.

do i=2,n
do j=2,n

dror(i,j)=(dom(i+1,j)-dom(i-1,j))/(2.0d0*dp)
drot(i,j)=(dom(i,j+1)-dom(i,j-1))/(2.0d0*dq)

end do
end do

C MERIDIONAL CIRCULATION. Note that this is calculated both at the
C   grid-points and mid-points. First the stream function psi(i,j)is
C   calculated. It is written in the file 'psi.dat'.
C

beta1=1.5d0
beta2=1.3d0

pp=0.635d0*pm
del=2.0000001d0
```

```
gm=3.1d0
c           gm=3.47d0
p0=(pm-pb)/3.50d0
open(82,file='psi.dat',status='unknown',access='append')

do  i=1,2*n+1
p=pb+float(i-1)/float(2*n)*(pm-pb)
do  j=1,2*n+1

q=qm-float(j-1)/float(2*n)*qm

C
if(q.le.pi/2.0d0)  then
exq0=dexp(-beta1*q**del)
exqm=dexp(beta2*(q-pi/2.0d0))
gau=dexp(-1.05*((p-p0)/gm)**2)

psi(i,j)=(p-pp)*dsin(pi*(p-pp)/(pm-pp))*
&     (1.00d0-exq0)*(1.00d0-exqm)*gau
end  if

C
if(q.gt.pi/2.0d0)  then
exq0=dexp(-beta1*(pi-q)**del)
```

```
exqm=dexp(beta2*(pi/2.0d0-q))
gau=dexp(-1.05*((p-p0)/gm)**2)
psi(i,j)=-(p-pp)*dsin(pi*(p-pp)/(pm-pp))*
& (1.00d0-exq0)*(1.00d0-exqm)*gau
end if

C
if(p.le.pp)then
psi(i,j)=0.0d0
end if

write(82,34)q,p,psi(i,j)
34          format(3(f13.5,1x))
end do
end do

close(82)

C       Now components of velocity are calculated at grid-points and
C       mid-points from the stream function psi(i,j).
C

open(87,file='v_theta.dat',status='unknown')
```

```
do  i=2,2*n
p=pb+float(i-1)*(pm-pb)/float(2*n)
do  j=2,2*n

q=qm-float(j-1)*qm/float(2*n)
vp1(i,j)=2.4573d0*v0*(psi(i,j+1)-psi(i,j-1))/(p**2*dsin(q)*dq
&              *(pm/p-0.95d0)**1.5)
vq1(i,j)=-2.4573d0*v0*(psi(i+1,j)-psi(i-1,j))/(p*dsin(q)*dp
&              *(pm/p-0.95d0)**1.5)
write(87,34)q,p,vq1(i,j)

vp(i,j)=vp1(i,j)*dt/(2.d0*p*dp)
vq(i,j)=vq1(i,j)*dt/(2.d0*p*dsin(q)*dq)
end do

end do

close(87)

C

C       PART V. Now we come to the central part of the
C       programme where the time advancement
C       takes place. 'k' is the counter to keep tab on
C       the time.  In each step of the do loop 'do k=1,kend',
```

```
C        the magnetic fields are advanced through one time step 'dt'.

kend = tmax/dt

t=0.0d0

do k=1,kend



C        (AB98,1)
C        NOTE 1: Added extra terms from modified dB/dt to ab(i,j),
C        bb(i,j), cb(i,j);
C        NOTE 3: Moved matrix coefficient block from PART IV to V to
C        accomodate time dependency of added terms

do i=2,n

p=pb+float(i-1)/float(n)*(pm-pb)
do j=2,n
q=qm-float(j-1)/float(n)*qm

C        (AB98, 4)
C        NOTE 1: Changed expression for etab(i,j) and etab2(i,j)
C        (both are expressions for eta toroidal)and moved them
```

```
C            into time loop to accomate for the time dependency of
C                 new eta expression

etab(i,j) = (0.00022d0 + (et1/2.00d0)*(1.0d0 +
&         erf((p-0.725d0*pm)/(0.030d0*pm)))+ (et0/2.0d0)
&         *(1.0d0+erf((p-0.975d0*pm)/(0.030d0*pm))))/
&         (1.0d0+(ub(i,j))**2)

end do

end do



do i=1,2*n+1
p=pb+float(i-1)*(pm-pb)/float(2*n)
do j=1,2*n+1

etab2(i,j) = (0.00022d0 + (et1/2.00d0)*(1.0d0 +
&         erf((p-0.725d0*pm)/(0.030d0*pm)))+ (et0/2.0d0)
&         *(1.0d0+erf((p-0.975d0*pm)/(0.030d0*pm))))/
&              (1.0d0+(ub(i,j))**2)
end do
end do
```

```
do i=2,2*n
p=pb+float(i-1)*(pm-pb)/float(2*n)
do j=2,2*n

q=qm-float(j-1)*qm/float(2*n)

C       (AB98, 2)
C       NOTE 2: As the extra terms of the modified matrix coefficient
C       include a previously undefined function d(eta)/d(theta
C       a new function detaq was defined for the purpose.
C       The definition was put within the time V loop to
C       accomodate time dependency.

deta(i,j)=(etab2(i+1,j)-etab2(i-1,j))/(dp)
detaq(i,j)=(etab2(i,j+1)-etab2(i,j-1))/(dq)
vpb(i,j)=(vp1(i,j)-deta(i,j))*dt/(2.d0*p*dp)

end do
end do


do i=2,n
p=pb+float(i-1)/float(n)*(pm-pb)
```

```
do  j=2,n

q=qm-float(j-1)/float(n)*qm

C         (AB98,1)
C         NOTE 1: Added extra terms from modified dB/dt to ab(i,j),
C                 bb(i,j),cb(i,j);
C         NOTE 3: Moved matrix coefficient block from PART IV to V to
C         accomodate time dependence of added terms

ab(i,j)=-(etab(i,j)*dt/(2.0d0*(p*dq)**2)-etab(i,j)*dt/
& (4.0d0*dtan(q)*p*p*dq)+
& vq(2*i-1,2*j-2)*dsin(q-dq/
& 2.0d0)*(1.0d0+vq(2*i-1,2*j-3)*dsin(q-dq))/2.0d0
&-detaq(2*i-1,2*j-1)*(dt/2.0d0)/(2.0d0*dq*p**2))
bb(i,j)=-(-etab(i,j)*dt/((p*dq)**2)-
& etab(i,j)*dt/(4.0d0*(p*dsin(q))**2)-(vq(
& 2*i-1,2*j)*dsin(q+dq/2.0d0)*(1.0d0+vq(2*i-1,2*j-1)*dsin(q))
&-vq(2*i-1,2*j-2)*dsin(
& q-dq/2.0d0)*(1.0d0-vq(2*i-1,2*j-1)*dsin(q)))/2.0d0+
& ((1/dtan(q))*detaq(2*i-1,2*j-1)*(dt/2.0d0))/(p**2))
cb(i,j)=-(etab(i,j)*dt/(2.0d0*(p*dq)**2)+
& etab(i,j)*dt/(4.0d0*dtan(q)*p*p*
& dq)-vq(2*i-1,2*j)*dsin(q+dq/2.0d0)*(1.0d0-vq(2*i-1,2*j+1)*dsin(q+
```

```
& dq))/2.0d0+(detaq(2*i-1,2*j-1)*(dt/2.0d0))/(2.0d0*dq*p**2))
db(i,j)=-(etab(i,j)*dt/(2.0d0*dp**2)-etab(i,j)*dt/(2.d0*p*dp)+
& vpb(2*i-1,2*j-1)*(p-dp/2.0d0)*(
& 1.0d0 +vpb(2*i-2,2*j-1)*(p-dp))/(2.0d0))
eb(i,j)=-(-etab(i,j)*dt/(dp**2)-
& 0.0d0*etab(i,j)*dt/(p*dp)-etab(i,j)*dt/(4.0d0*
& (p*dsin(q))**2)-dvp(2*i-1,2*j-1)*dt/2.0d0-vpb(2*i-1,2*j-1)*(dp+
& vpb(2*i,2*j-1)*p*(p+dp/2.0d0)+(p-dp/2.0d0)*p*
& vpb(2*i-2,2*j-1))/2.0d0)
fb(i,j)=-(etab(i,j)*dt/(2.0d0*dp**2)+
& etab(i,j)*dt/(2.d0*p*dp)-vpb(2*i-1,2*j-1)*
& (p+dp/2.0d0)*(1.0d0 -vpb(2*i,2*j-1)*(p+dp))/2.0d0)

end do
end do



C PART V-A. CALCULATING TIME ADVANCED MAGNETIC FIELDS AT INTERIOR GRID
C POINTS. Sect. 5 of the 'Guide' explains how magnetic fields are
C advanced by solving tridiagonal matrices with the subroutine 'tridag'

do i=2,n
do j=2,n
phi(i,j)=-d(i,j)*u(i-1,j)+(-e(i,j)+1.0d0)*u(i,j)
```

```
&          -f(i,j)*u(i+1,j)+al0*al(i,j)*ub(i,j)*dt/2.0d0

end do
end do

do i=2,n
p=pb+float(i-1)/float(n)*(pm-pb)
do j=2,n

q=qm-float(j-1)/float(n)*qm
br = (u(i,j+1)*dsin(q+dq)-u(i,j-1)*dsin(q-dq))*dt/(4.0d0*dq)
bt=(u(i-1,j)*(p-dp)-u(i+1,j)*(p+dp))*dsin(q)*dt/(4.0d0*p*dp)
phib(i,j)=-db(i,j)*ub(i-1,j)+(-eb(i,j)+1.0d0)*ub(i,j)
&          -fb(i,j)*ub(i+1,j)+br*dror(i,j)+bt*drot(i,j)

end do
end do



do i=2,n
do j=2,n

a1(j-1)=a(i,j)
b1(j-1)=b(i,j)+1.0d0
```

```
c1(j-1)=c(i,j)
r(j-1)=phi(i,j)

end do

r(1)=phi(i,2)-a1(1)*u(i,1)
r(n-1)=phi(i,n)-c1(n-1)*u(i,n+1)
call tridag(a1,b1,c1,r,uu,n-1)

do j=2,n
u(i,j) = uu(j-1)

phi(i,j)=-phi(i,j)+2.0d0*uu(j-1)

end do
end do



!              do i=2,n
!                do j=2,n
!                  p=pb+float(i-1)/float(n)*(pm-pb)
!                  q=qm-float(j-1)/float(n)*qm
!                  br = (u(i,j+1)*dsin(q+dq)-u(i,j-1)*
!                                  dsin(q-dq))*dt/(4.0d0*dq)
```

```
!                bt=(u(i-1,j)*(p-dp)-u(i+1,j)*(p+dp))*
!                                   dsin(q)*dt/(4.0d0*p*dp)
!                 phib(i,j)=-db(i,j)*ub(i-1,j)+(-eb(i,j)+1.0d0)*ub(i,j)
!     &             -fb(i,j)*ub(i+1,j)+br*dror(i,j)+bt*drot(i,j)
!               end do
!             end do



do i=2,n
do j=2,n

a1(j-1)=ab(i,j)
b1(j-1)=bb(i,j)+1.0d0
c1(j-1)=cb(i,j)
r(j-1)=phib(i,j)

end do

r(1)=phib(i,2)-a1(1)*ub(i,1)
r(n-1)=phib(i,n)-c1(n-1)*ub(i,n+1)

call tridag(a1,b1,c1,r,uub,n-1)
do j=2,n
```

```
ub(i,j) = uub(j-1)
phib(i,j)=-phib(i,j)+2.0d0*uub(j-1)
phi(i,j)=phi(i,j)+al0*al(i,j)*ub(i,j)*dt/2.0d0

end do
end do



do i=2,n
p=pb+float(i-1)/float(n)*(pm-pb)
do j=2,n

q=qm-float(j-1)/float(n)*qm
br = (u(i,j+1)*dsin(q+dq)-u(i,j-1)*dsin(q-dq))*dt/(4.0d0*dq)
bt=(u(i-1,j)*(p-dp)-u(i+1,j)*(p+dp))*dsin(q)*dt/(4.0d0*p*dp)
phib(i,j)=phib(i,j)
&          +br*dror(i,j)+bt*drot(i,j)

end do
end do



do j=2,n
do i=2,n
```

```
a1(i-1)=d(i,j)
b1(i-1)=e(i,j)+1.0d0
c1(i-1)=f(i,j)
r(i-1)=phi(i,j)

end do

r(1)=phi(2,j)-a1(1)*u(1,j)
r(n-1)=phi(n,j)-c1(n-1)*u(n+1,j)
call tridag(a1,b1,c1,r,uu,n-1)
do i=2,n
u(i,j)=uu(i-1)

end do
end do

C

do j=2,n
do i=2,n

a1(i-1)=db(i,j)
b1(i-1)=eb(i,j)+1.0d0
c1(i-1)=fb(i,j)
```

```
r(i-1)=phib(i,j)

end do

r(1)=phib(2,j)-a1(1)*ub(1,j)
r(n-1)=phib(n,j)-c1(n-1)*ub(n+1,j)
call tridag(a1,b1,c1,r,uub,n-1)
do i=2,n
ub(i,j)=uub(i-1)

end do
end do
```