Solar Phys (2012) 281:621–637
DOI 10.1007/s11207-012-0074-x

A Unified and Very Fast Way for Computing the Global Potential and Linear Force-Free Fields

Chaowei Jiang · Xueshang Feng

Received: 27 March 2012 / Accepted: 6 July 2012 / Published online: 26 July 2012
© Springer Science+Business Media B.V. 2012

Abstract We present a fast solver for computing potential and linear force-free fields (LFFF) above the full solar disk with a synoptic magnetic map as input. The global potential field and the LFFF are dealt with in a unified way by solving a three-dimensional Helmholtz equation in a spherical shell and a two-dimensional Poisson equation on the solar surface. The solver is based on a combination of the spectral method and the finite-difference scheme. In the longitudinal direction the equation is transformed into the Fourier spectral space, and the resulting two-dimensional equations in the r–θ plane for the Fourier coefficients are solved by finite differencing. The solver shows an extremely fast computing speed, *e.g.*, the computation for a magnetogram with a resolution of $180(\theta) \times 360(\phi)$ is completed in less than 2 s. Even on a high-resolution 600×1200 grid, the solution can be obtained within only about one minute on a single CPU. The solver can potentially be applied directly to the original resolution of observed magnetograms from SDO/HMI for routinely analyzing daily full-disk data.

Keywords Fast solver · Linear force-free fields · Potential fields · Solar corona

1. Introduction

Magnetic field extrapolation is an important tool to study the three-dimensional (3D) solar coronal magnetic field, which is difficult to directly measure (Sakurai, 1989; Aly, 1989; Amari *et al.*, 1997; McClymont, Jiao, and Mikic, 1997; Aschwanden, 2005; Wiegelmann, 2008). Three models being used most popularly for field extrapolation are the potential-field model, the linear force-free field (LFFF) model, and the nonlinear force-free field (NLFFF) model. On a large scale involving global structures, coronal fields are usually extrapolated from the line-of-sight (LoS) magnetogram using the potential-field model with

C. Jiang (✉) · X. Feng
SIGMA Weather Group, State Key Laboratory for Space Weather, Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing 100190, China
e-mail: cwjiang@spaceweather.ac.cn

some kind of upper boundary condition to emulate the effect of the solar wind expansion, *e.g.*, the potential-field source-surface (PFSS) model (Altschuler and Newkirk, 1969; Schatten, Wilcox, and Ness, 1969; Hoeksema, 1984). By these models and by using synoptic magnetic field data, the extrapolated global fields can be computed and they are used to study the general structures of the corona and the heliosphere (*e.g.*, the location, shape, and size of coronal holes, coronal streamers, and heliospheric current sheet, and their evolution). With some empirical relations developed in the Wang–Sheeley–Arge (WSA) model, the PFSS field can also be used to predict the solar wind speed and the interplanetary magnetic field (IMF) at the Earth (Wang and Sheeley, 1990; Arge and Pizzo, 2000). On a local scale with fine structures, when one's interest is focused on the active regions, LFFF and more general NLFFF extrapolation which needs vector magnetograms are more suitable to characterize the strongly non-potential regions with significant currents. The three models are governed by the same force-free equation

$$\nabla \times \mathbf{B} = \alpha \mathbf{B}, \qquad \nabla \cdot \mathbf{B} = 0, \tag{1}$$

where $\mathbf{B}$ is the magnetic field vector, α is a scalar function called force-free parameter. The potential model is represented by $\alpha = 0$, which means that no electric current is present in the field volume, and the equation can be simplified as $\mathbf{B} = \nabla\psi$ with $\nabla^2\psi = 0$. The LFFF is denoted by $\alpha =$ constant and NLFFF by generally variable α but invariable along each field line.

The currently available methods for global potential field are mainly based on expansion of spherical harmonics or a full 3D finite-difference scheme. The spherical harmonic expansion (SPHE) method can be easily implemented and the field value is expressed in an analytic form at an arbitrary location above the solar surface. The full 3D finite-difference method is much more complicated by numerically discretizing the Laplace equation $\nabla^2\psi = 0$, which results in a large sparse linear equation system for all the unknowns $\psi_{i,j,k}$ needed to be solved. What makes its implementation complicated is that such linear equations cannot be solved directly and an iterative approach is generally necessary. For relatively low-resolution data like a 180×360 magnetogram, these methods perform good and fast with results coming out in minutes or less than an hour on a single CPU. However, it could be very unsatisfactory if one wants to apply these methods to very high-resolution data matching the observations in an operational basis, for example a 1080×3600 synoptic magnetogram from SOHO/MDI and a 4096×4096 full-disk magnetogram from SDO/HMI. In our experience, the SPHE method implemented by the PFSS package[1] in the SolarSoftWare (SSW) library takes about several minutes for a resolution of 180×360, but requires more than two hours when the resolution is doubled (see Section 4), which means the computing time is scaled up by 40 times and it could be even unbearable for much larger magnetograms. Very recently, Tóth, van der Holst, and Huang (2011) developed a full 3D finite-difference iterative potential solver (FDIPS)[2] which uses a bi-conjugate gradient-stabilized (BiCGSTAB) iterative method (Van der Vorst, 1992) to solve the resulting linear system with an incomplete lower-upper decomposition (ILU) preconditioner. As reported their solver needs about an hour when applied to a resolution of 180×360, while to perform on much higher resolutions within a time of hours, it can only rely on massively parallel computation.

[1] http://www.lmsal.com/~derosa/pfsspack.

[2] http://csem.engin.umich.edu/fdips/.

The present work is devoted to the development of a new fast and unified solver for computing the global potential and LFFF fields by a combination of the Fourier series expansion and the finite-difference (FD) schemes. It will be shown that our mixed spectral finite-difference (MSFD) solver is extremely fast by computing the solution to a problem of 180×360 data points within about a second and even a computation of 600×1200 is finished in only nearly a minute on a single CPU. This solver is also capable of solving a LFFF field on the same magnetogram with arbitrarily given force-free parameter α without costing any additional computational resources. A global LFFF model is criticized and rarely considered for the global corona due to the facts that the magnetic energy is unbounded (Seehafer, 1978) and generally the large-scale force-free parameter α (also known as the current helicity) has opposite signs between the hemispheres (Pevtsov, Canfield, and Metcalf, 1995; Bao and Zhang, 1998). However, a global LFFF solver can still be used for many other applications, for example, initializing the global NLFFF or magnetohydrodynamics codes (Jiang and Feng, 2012; Jiang, Feng, and Xiang, 2012; Feng *et al.*, 2012). Recently, Hu and Dasgupta (2008) developed a non-force-free field model that can represent the local and global coronal fields. In their non-force-free model based on the principle of minimum dissipation rate, the magnetic field is expressed as the superposition of one potential field and two LFFFs. To match the numerical solution with the measured magnetogram, the parameters α of the LFFFs are determined by a trial and error procedure which needs many times of LFFF computations. If the model is used for the global corona, a fast global LFFF solver is highly desirable and our solver can be a good candidate to meet the need. In view of its very fast speed for dealing with high-resolution data, our global LFFF solver can also be applied directly to the extrapolation of local LFFF for active regions, and the computation can benefit because there is no need to consider the lateral boundaries, which cause problems in the codes designed for local extrapolation.

The remainder of the paper is organized as follows. In Section 2 we review the LFFF equation in spherical coordinates and its boundary conditions. In Section 3 the numerical algorithm of the MSFD solver is described. The results are validated and discussed in Section 4 and finally we draw conclusions in Section 5.

2. The LFFF Equation in Spherical Coordinates

In spherical geometry, the linear force-free equation can be written equivalently as

$$\mathbf{B} = \nabla \times (\nabla \times \psi \mathbf{e}_{\mathrm{r}}) + \alpha \nabla \times \psi \mathbf{e}_{\mathrm{r}}, \tag{2}$$

where $\mathbf{e}_{\mathrm{r}}$ is the unit vector in the radial direction and the scalar function $\psi(r, \theta, \phi)$ satisfies the Helmholtz equation (Seehafer, 1978; Durrant, 1989)

$$\left(\nabla^2 + \alpha^2\right)\psi = 0. \tag{3}$$

Expressed in components, Equation (2) becomes

$$\begin{aligned} B_r &= -\left[\frac{1}{r^2 \sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\psi}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2\psi}{\partial\phi^2}\right], \\ B_\theta &= \frac{1}{r}\frac{\partial^2\psi}{\partial\theta\partial r} + \frac{\alpha}{r\sin\theta}\frac{\partial\psi}{\partial\phi}, \\ B_\phi &= \frac{1}{r\sin\theta}\frac{\partial^2\psi}{\partial\phi\partial r} - \frac{\alpha}{r}\frac{\partial\psi}{\partial\theta}. \end{aligned} \tag{4}$$

At the solar surface (photosphere) $r = R$ with the radial component of the magnetic field $M(\theta, \phi)$ given,[3] we have

$$\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\psi}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2\psi}{\partial\phi^2} = -R^2 M(\theta, \phi), \tag{5}$$

which is a two-dimensional (2D) Poisson equation on the spherical surface. By solving this equation we obtain the value of $\psi_R \equiv \psi(R, \theta, \phi)$ on the solar surface which serves as the bottom boundary condition for the Helmholtz equation (3). Besides, a second boundary surface is needed to be set for numerical discretization of the Helmholtz equation. For this purpose we simply specify the upper boundary condition at a source surface R_S similar to the PFSS model as

$$\frac{\partial\psi}{\partial r} = 0 \quad \text{at } R_S = 2.5R. \tag{6}$$

Then the Helmholtz equation is well defined in the spherical shell ($R \le r \le R_S$) by the Dirichlet boundary condition at the inner surface and the Neumann boundary condition at the outer surface. Note that defining the outer boundary condition in the above way makes the LFFF solution with $\alpha = 0$ the same as the PFSS model, because Equation (6) ensures $B_\theta = B_\phi = 0$ according to Equation (4) with zero α. Durrant (1989) suggested another way of defining the outer boundary condition by minimizing the tangential field on the source surface. Setting an upper boundary for the global LFFF is also required theoretically since a magnetic field being force-free with nonzero constant α everywhere outside the sun cannot have a finite energy content and may even not have a unique solution (Chiu and Hilton, 1977; Seehafer, 1978; Gary, 1989).

For solely solving the potential field, it is preferred in consideration of numerical accuracy to solve the Laplace equation $\nabla^2\psi = 0$ (with boundary conditions $\partial\psi/\partial r|_R = M$ and $\psi|_{R_S} = 0$) to compute $\mathbf{B}$ by $\mathbf{B} = \nabla\psi$ rather than computing the LFFF with $\alpha = 0$, because if ψ is obtained with the same accuracy, deriving the magnetic vector from $\mathbf{B} = \nabla\psi$ is more accurate than using Equation (2), which needs a second-order mixed difference. In the code we have implemented both ways for computing the potential field. Anyway, computation of the LFFF and potential field can be unified by using the same 3D Helmholtz solver.

3. Numerical Method

The computing kernel we need is a 2D sphere Poisson solver for Equation (5) and a 3D Helmholtz solver for Equation (3) or the Laplace equation ($\alpha = 0$). Here we developed the solvers in a similar way as that proposed by Lai, Lin, and Wang (2002). The basic ideal is combining the spectral and the finite-difference methods. According to the periodic condition in the longitudinal direction, function ψ can be expressed by a Fourier series of ϕ and then the Fourier coefficients are solved by the finite-difference method in the (r, θ) plane. In this way we can avoid the 3D-difference discretization, which results in a large linear system that cannot be solved by any direct method. Specifically function ψ can be approximated by the discrete Fourier series

$$\psi(r, \theta, \phi) = \sum_{n=-N/2}^{N/2-1} \psi_n(r, \theta) e^{in\phi}, \tag{7}$$

[3]Generally observations do not give the radial component M but the line-of-sight component of the field (B_ℓ), and M is converted from B_ℓ assuming that the field is radial at the photosphere.

where $\psi_n(r,\theta)$ is the *complex* Fourier coefficient given by

$$\psi_n(r,\theta) = \frac{1}{N}\sum_{k=0}^{N-1}\psi(r,\theta,\phi_k)\mathrm{e}^{-\mathrm{i}n\phi_k}, \tag{8}$$

and $\phi_k = 2\pi k/N$ and N is the number of grid points in the ϕ-direction. Substituting the Fourier series (7) into Equation (3) gives

$$\frac{\partial^2\psi_n}{\partial r^2} + \frac{2}{r}\frac{\partial\psi_n}{\partial r} + \frac{1}{r^2}\frac{\partial^2\psi_n}{\partial\theta^2} + \frac{\cot\theta}{r^2}\frac{\partial\psi_n}{\partial\theta} + \left(\alpha^2 - \frac{n^2}{r^2\sin^2\theta}\right)\psi_n = 0. \tag{9}$$

Substituting the Fourier series (7) into the boundary equation (5) gives

$$\frac{\mathrm{d}^2\psi_n(R,\theta)}{\mathrm{d}\theta^2} + \cot\theta\frac{\mathrm{d}\psi_n(R,\theta)}{\mathrm{d}\theta} - \frac{n^2}{\sin^2\theta} = -R^2M_n(\theta), \tag{10}$$

where $M_n(\theta)$ is the Fourier coefficient of the magnetogram $M(\theta,\phi)$. By first solving the preceding ordinary differential equation (10) and then the 2D partial differential equation (PDE), Equation (9), using a finite-difference method, we obtain the Fourier coefficients ψ_n and thus the solution ψ. The Fourier transform can be routinely implemented by the discrete Fourier transform (DFT) functions in Intel's Math Kernel Library (MKL) and the finite-difference algorithm is described as follows.

A uniform grid is used and to avoid the pole problem (*i.e.*, the denominator $\sin\theta = 0$ when $\theta = 0$ or π), the grid is shifted by one-half mesh width in the θ-direction as

$$r_i = R + i\Delta r, \qquad \theta_j = (j - 1/2)\Delta\theta, \tag{11}$$

for $i = 0,\ldots,L+1$, $j = 0,\ldots,M+1$ with the grid size $\Delta r = (R_\mathrm{S} - R)/(L+1)$ and $\Delta\theta = \pi/M$. The grid boundaries at $j = 0$ and $j = M+1$ serve as ghost points and the real boundary conditions are specified at $i = 0$ (solar surface) and $i = L+1$ (source surface). Denoting the discrete values as $u_{i,j} = \psi_n(r_i,\theta_j)$ and applying the centered difference to Equation (10), we have

$$\frac{u_{0,j+1} - 2u_{0,j} + u_{0,j-1}}{(\Delta\theta)^2} + \cot\theta_j\frac{u_{0,j+1} - u_{0,j-1}}{2\Delta\theta} - \frac{n^2}{\sin^2\theta_j}u_{0,j} = -R^2M_n(\theta_j), \tag{12}$$

for $j = 1,\ldots,M$. This is a tridiagonal linear system for $u_{0,j}$, which can be solved by Gaussian elimination with $\mathrm{O}(M)$ arithmetic operations. The boundary condition can be deduced from the periodic condition $\psi(r,\theta,\phi) = \psi(r,-\theta,\phi+\pi)$, which says

$$\psi(r,-\theta,\phi+\pi) = \sum_{n=-N/2}^{N/2-1}\psi_n(r,-\theta)\mathrm{e}^{\mathrm{i}n(\phi+\pi)} = \sum_{n=-N/2}^{N/2-1}(-1)^n\psi_n(r,-\theta)\mathrm{e}^{\mathrm{i}n\phi}. \tag{13}$$

Comparing the coefficients of Equation (7) and Equation (13) gives

$$\psi_n(r,-\theta) = (-1)^n\psi_n(r,\theta). \tag{14}$$

Similarly, $\psi_n(r,\pi+\theta) = (-1)^n\psi_n(r,\pi-\theta)$ is also satisfied. Then the boundary condition for Equation (12) is

$$u_{0,0} = (-1)^n u_{0,1}, \qquad u_{0,M+1} = (-1)^n u_{0,M}. \tag{15}$$

It should be noted that when $n = 0$ the solution to Equation (12) is non-unique upon a constant, and we choose $u_{0,1} = 0$ to determine the solution.

In the same way, applying the centered difference to Equation (9) gives

$$r_i^2\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta r)^2}+2r_i\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta r}+\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta\theta)^2}$$
$$+\cot\theta_j\frac{u_{i,j+1}-u_{i,j-1}}{2\Delta\theta}+\left(\alpha^2r_i^2-\frac{n^2}{\sin^2\theta_j}\right)u_{i,j}=0 \tag{16}$$

for the internal points $1\le i\le L, 1\le j\le M$, and the ghost point values can be given by

$$u_{i,0}=(-1)^n u_{i,1},\qquad u_{i,M+1}=(-1)^n u_{i,M}. \tag{17}$$

On the solar surface, for solving the LFFF we have the boundary condition on $u_{0,j}$ obtained from Equation (12), or for only the potential field we use one-sided difference of second-order for the Neumann boundary condition $\frac{\partial\psi}{\partial r}=M$,

$$\frac{-3u_{0,j}+4u_{1,j}-u_{2,j}}{2\Delta r}=M_n(\theta_j), \tag{18}$$

which can maintain the same accuracy with the centered difference. On the outer surface for LFFF we have

$$\frac{-3u_{L+1,j}+4u_{L,j}-u_{L-1,j}}{2\Delta r}=0, \tag{19}$$

or for the potential field $u_{L+1,j}=0$.

The above discrete equations (16) along with the boundary conditions lead to a linear system for unknowns $u_{i,j}$ with a $L\times L$ block-tridiagonal coefficient matrix. This large system can generally be handled by the iterative methods, such as the Krylov-space solver (*e.g.*, the BiCGSTAB). Fortunately, a direct solver exists for this special type of linear systems by the generalized cyclic reduction method (Swarztrauber, 1974). An open-source Fortran package *Fishpack*[4] (Adams, Swarztrauber, and Sweet, 1980) has been designed for directly solving similar linear systems (from the finite difference of 2D separable elliptic PDEs) using cyclic reduction. The subroutine named `cblktri.f` in *Fishpack* for complex block-tridiagonal linear systems is utilized for our purpose. Accordingly, Equation (16) is written as

$$a_1(i)u_{i+1,j}+a_2(i)u_{i-1,j}+\big[b_1(i)+b_2(j)\big]u_{i,j}$$
$$+c_1(j)u_{i,j+1}+c_2(j)u_{i,j-1}=d(i,j) \tag{20}$$

with the coefficients

$$a_1(i)=\frac{r_i^2-r_i\Delta r}{(\Delta r)^2},\qquad a_2(i)=\frac{r_i^2+r_i\Delta r}{(\Delta r)^2},\qquad b_1(i)=\alpha^2r_i^2-\frac{2r_i^2}{(\Delta r)^2},$$
$$b_2(j)=\frac{-2}{(\Delta\theta)^2}-\frac{n^2}{\sin^2\theta_j},\qquad c_1(j)=\frac{1+\frac{1}{2}\cot\theta_j\Delta\theta}{(\Delta\theta)^2},\qquad c_2(j)=\frac{1-\frac{1}{2}\cot\theta_j\Delta\theta}{(\Delta\theta)^2}, \tag{21}$$

and these coefficients are provided as input arguments to `cblktri.f` to find the solution. Note that neighboring to the boundary the coefficients should be adjusted properly according to the boundary conditions (see Equations (17)–(19)). The number of operations needed by the subroutine is $O(ML\log_2 L)$. It should be noted here that a direct solver is very suitable for the present 2D linear system of which the matrix size is not too large and a parallel computation is not necessary. For a very large system such as that resulting from the scheme of full-3D difference of equations, iterative schemes may be more efficient and robust especially on parallel machines.

[4] http://www.netlib.org/fishpack.

Table 1 Benchmark tests of the solvers. The resolution is given by $L = M = N/2$, and $x = r\sin\theta\cos\phi$, $y = r\sin\theta\sin\phi$, $z = r\cos\theta$. L_∞ error is the maximum error between the numerical solution and the exact solution, and L_∞ order is the convergence order of the L_∞ error in terms of the mesh size.

Solver	Test solution	Resolution N	L_∞ error	L_∞ order
2D Poisson solver	$\psi = \sin\theta\cos\phi$	32	9.355E-04	
		64	2.256E-04	2.05
		128	5.642E-05	2.00
	$\psi = \cos\theta\sin^2\theta\cos(2\phi - 3)$	32	4.106E-03	
		64	1.031E-03	1.99
		128	2.573E-04	2.00
3D Helmholtz solver	$\psi = \sin(x + y + z), \alpha = \sqrt{3}$	32	2.316E-02	
		64	6.168E-03	1.91
		128	1.566E-03	1.98
	$\psi = \sin x \sin y \sin z, \alpha = \sqrt{3}$	32	5.496E-03	
		64	1.348E-03	2.03
		128	3.412E-04	1.98

In Table 1 we give results of several benchmark tests for the 2D Poisson and the 3D Helmholtz solvers using analytic solutions. For the 3D tests, we use the Dirichlet boundary condition at the inner surface and the Neumann condition at the outer surface. The convergence rates of the results show that the designed order of accuracy is well achieved.

After obtaining ψ, the final vector $\mathbf{B}$ can be derived according to Equation (2) or $\mathbf{B} = \nabla\psi$ for only the potential field. Either full-3D difference or the same procedure of Fourier transform in ϕ and finite difference in (r, θ) can be used to get $\mathbf{B}$ and we will compare the results of using these two schemes in Section 4.

A non-uniform radial grid is also preferred such as $\Delta r \sim r$ or $\Delta r \sim r^2$ for saving computational resources, but then the centered difference cannot be applied directly due to the unequal grid space. To extend the above solver to a non-uniform radial grid, we transform Equation (9) from a given non-uniform radial grid r to a new uniform radial grid ξ using the relations

$$\begin{aligned} &r = r(\xi), \qquad \frac{\partial u}{\partial r} = \xi_r \frac{\partial u}{\partial \xi}, \\ &\frac{\partial^2 u}{\partial r^2} = \frac{\partial}{\partial r}\left(\xi_r \frac{\partial u}{\partial \xi}\right) = \xi_{rr}\frac{\partial u}{\partial \xi} + \frac{\partial^2 u}{\partial \xi^2}\xi_r^2, \end{aligned} \tag{22}$$

with ξ as the new radial coordinate (here u represents an arbitrary function). For example one can simply set $\xi = i$ in a discrete sense, thus $r_i = r(i) = r(\xi)$. To derive ξ_r and ξ_{rr}, we have

$$\xi_r = \frac{1}{r_\xi}, \tag{23}$$

$$\xi_{rr} = \frac{\mathrm{d}}{\mathrm{d}r}\left(\frac{1}{r_\xi}\right) = -(r_{\xi\xi}\xi_r)/(r_\xi)^2 = -(r_{\xi\xi})/(r_\xi)^3, \tag{24}$$

where

$$r_\xi = \frac{r_{i+1} - r_{i-1}}{2}, \qquad r_{\xi\xi} = r_{i+1} - 2r_i + r_{i-1}. \tag{25}$$

Then the centered-difference method is applied to the transformed equations. If necessary, the same procedure can also deal with a non-uniform longitude or latitude grid (*e.g.*, the uniform-spaced sine-latitude mesh with which the synoptic maps are usually provided).

Finally, in order that Equation (5) has a solution on the sphere, the right-hand side function $M(\theta, \phi)$ must satisfy the compatibility condition

$$\oint M(\theta, \phi)\, \mathrm{d}s = 0, \tag{26}$$

which is also the divergence-free constraint for the magnetogram. However, in measured data this cannot be guaranteed, thus usually a deviation of ΔM is subtracted from the map to fulfill the flux-balance condition

$$\oint (M - \Delta M)\, \mathrm{d}s = 0, \tag{27}$$

and we choose $\Delta M = k|M|$ to retain the global structure in the map (*e.g.*, the magnetic neutral lines at $M = 0$). This gives the coefficient k as $k = \frac{\oint M\, \mathrm{d}s}{\oint |M|\, \mathrm{d}s}$.

4. Results

To validate the MSFD method, we compare our solution for the potential with that computed by the PFSS package in SSW. We take the extrapolation of Carrington rotation (CR) 2029 as an example. The synoptic map for B_r is obtained from SOHO/MDI with an original resolution of 1080×3600 on the $\sin\theta$–ϕ plane. We remap and rebin the original data onto the θ–ϕ plane. Both solutions are computed on the same grid with a size of $54(r) \times 180(\theta) \times 360(\phi)$, which is uniform in the θ–ϕ plane and non-uniform in the radial direction ($\Delta r = r\Delta\theta$). In the MSFD method, as noted in Section 2, we compute the field by solving the Laplace equation and then using two schemes to get $\mathbf{B}$ from $\mathbf{B} = \nabla\psi = (\frac{\partial\psi}{\partial r}, \frac{1}{r}\frac{\partial\psi}{\partial\theta}, \frac{1}{r\sin\theta}\frac{\partial\psi}{\partial\phi})$, *i.e.*,

i) the scheme of combined spectral/finite-difference method as described above (referred to as the MSFD-M solution in the following) and
ii) a full-3D finite-difference method (referred to as the MSFD-FD solution).

In fact for components B_r and B_θ, both the MSFD-M and the MSFD-FD schemes give exactly the same solution and the only difference between them is the longitudinal component B_ϕ. To compute the longitudinal derivative $\frac{\partial\psi}{\partial\phi}$, the MSFD-FD scheme performs centered difference on the grid point values $\psi_{i,j,k}$, while the MSFD-M scheme differentiates directly the discrete Fourier series (Equation (7)) term-by-term, which gives

$$\frac{\partial\psi}{\partial\phi} = \sum_{n=-N/2}^{N/2-1} \mathrm{i}n\psi_n(r, \theta)\mathrm{e}^{\mathrm{i}n\phi}. \tag{28}$$

In the SSW/PFSS package the solution is obtained by SPHE with the highest order of $L_{\max} = 180$.

Figure 1 compares the 3D field lines of the solutions, which are highly consistent with each other (for the MSFD method only the MSFD-M solution is shown, since the MSFD-FD solution is almost the same). Only by a careful inspection can very small difference be found between them, and this difference is due to the numerical discretization errors. In Figure 2 we compare quantitatively the field values by plotting the contours of the two solutions on the same figure. The left column is the solutions sliced at a height of $r_{10} = 1.19R$ and the

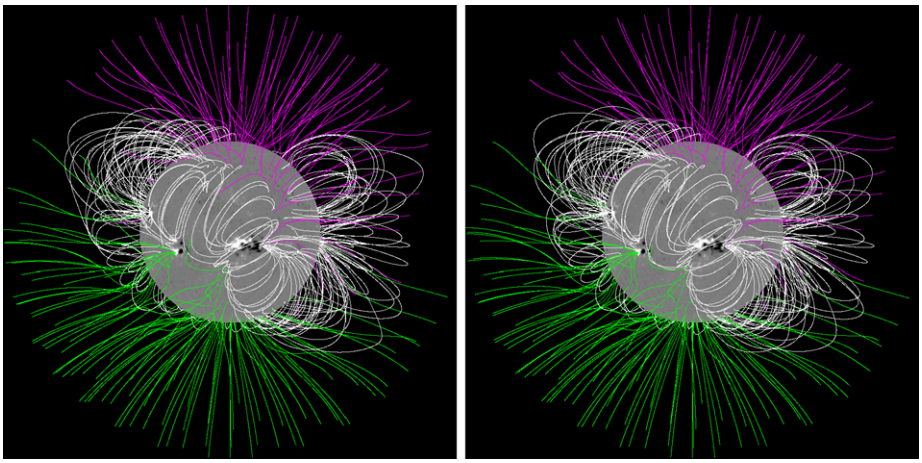

Figure 1 Comparison of the 3D field configurations between our solution (left) and that by SSW/PFSS (right). Closed field lines are rendered with white color and open field lines are rendered with pink and green according their polarities.

right column for $r_{30} = 1.68R$. Again the differences are rather small since the contours of different solutions almost overlap each other.

In Figure 3 we plot profiles of the field values along the radial line of $(\theta, \phi) = (20°, 50°)$. This figure shows obvious difference between the solutions only near the solar surface $r = R$. To show how the deviation of the entire solutions behaves along the radial direction, the average relative error between our solutions $\mathbf{B}$ and the SSW/PFSS solution $\mathbf{B}_{\rm ssw}$ is calculated as

$$E_{\rm n}(r_i) = \frac{\sum_{j,k} |\mathbf{B}(r_i, \theta_j, \phi_k) - \mathbf{B}_{\rm ssw}(r_i, \theta_j, \phi_k)|}{\sum_{j,k} |\mathbf{B}_{\rm ssw}(r_i, \theta_j, \phi_k)|}, \tag{29}$$

which are plotted in the bottom-right panel in Figure 3 (the black line denotes the MSFD-FD solution and red for the MSFD-M solution). As can be seen, the error $E_{\rm n}$ at the solar surface is rather large with a value of ≈ 0.5. It drops very quickly to nearly 2 % and the solutions match quite well above $r = 1.1R$. This near-surface error is caused by different numerical errors in the methods, which become significant especially near the bottom boundary.

Figure 4 shows the computed magnetic maps on the solar surface by different methods. For instance, the derived field B_r in our method matches exactly the given boundary data $M(\theta, \phi)$ because the discrete Fourier series is precise at the grid nodes, while it cannot be ensured in the SPHE method due to the truncation terms unless the expansion order $L_{\rm max}$ goes to ∞. Because of the truncation errors, the simulated synoptic map by the SPHE method usually shows ringing patterns surrounding strong magnetic features (Tran, 2009; Tóth, van der Holst, and Huang, 2011). Such aliasing effect is more obvious for the case with lower $L_{\rm max}$ as shown by the result of $L_{\rm max} = 90$ in Figure 4. On the other hand the value of $L_{\rm max}$ should not exceed an upper limit determined by the size of the data (Tran, 2009), in order not to over-sample. Therefore a reasonable upper limit is $360°/(2\Delta)$ where Δ is the size of a pixel in the map (*e.g.*, $\Delta = 1°$ in the present case). That is why we use $L_{\rm max} = 180$ for the SSW/PFSS solution. As can be seen in Figure 4, the MSFD-M solution for B_ϕ also shows obvious aliasing in the ϕ direction, especially across the strong magnetic features, while the solution for B_r and B_θ does not show such artifacts. The oscillations in

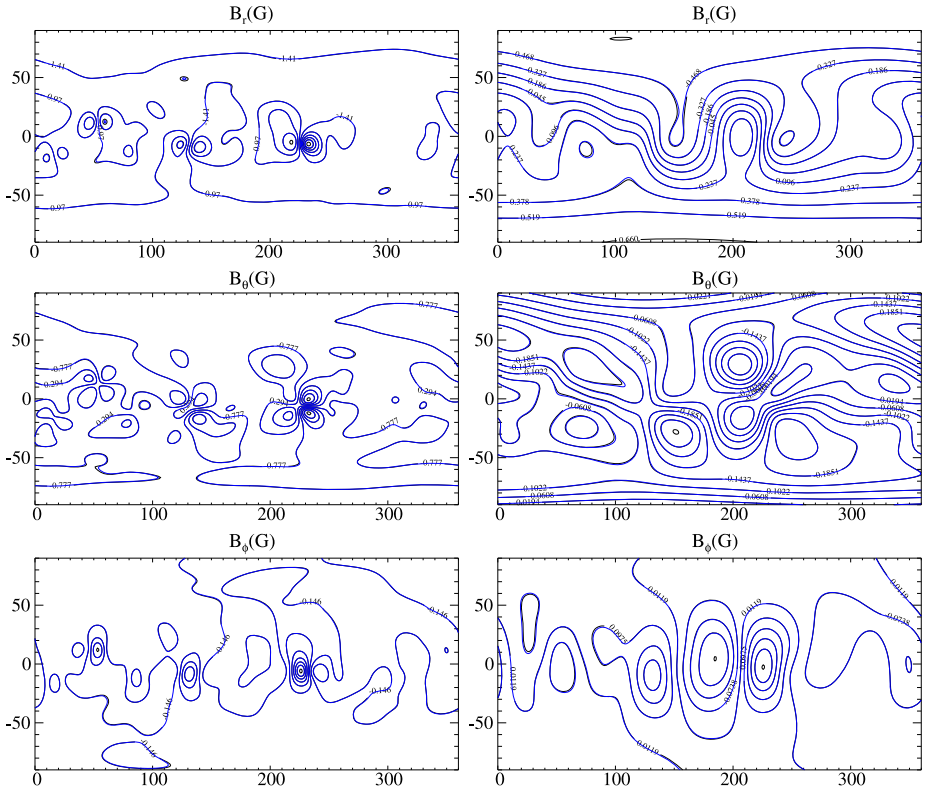

Figure 2 Contours of the field values in our solution (black lines) and in the SSW/PFSS solution (blue lines) at heights of $r_{10} = 1.19R$ (left column) and $r_{30} = 1.68R$ (right column).

the MSFD-M solution is due to the fact that the DFT-based differentiation (Equation (28)) is not accurate for functions with sharp gradients or discontinuities because of the Gibbs phenomenon (*e.g.*, see Canuto *et al.*, 2006).

We further compare the surface solutions by plotting the calculated magnetic field components as a function of longitude ϕ along a given latitude ($\theta = 104.5°$, which is chosen to show the magnetic field across some active regions with strong field gradient) in Figure 5. The SSW/PFSS solution (blue curves in the figure) exhibits high-frequency oscillations of an amplitude of ≈ 10 G (gauss) overlying the MSFD-FD solution (black curves). The oscillations of B_ϕ seen in the MSFD-M solution (the red curves in the third panel of the figure) are much larger with an amplitude exceeding 20 G and exhibit obvious overshoot across the active regions. The reason for these oscillations can be well understood by inspecting the last panel of Figure 5, in which the grid-node values (black dots) and the discrete Fourier series (the red curve) of ψ are plotted. It is the differentiation of high-frequency mode terms that gives the oscillations in the longitudinal derivatives. On the other hand, centered difference on the grid-node solution can avoid such oscillations. By comparison, we conclude that the MSFD-FD scheme gives the most accurate and reasonable solution that does not exhibit those artifacts. It is also expected that the artifacts manifest only near the surface where the fields are very rough. As shown by Figure 6, in which the same field profiles as in Figure 5

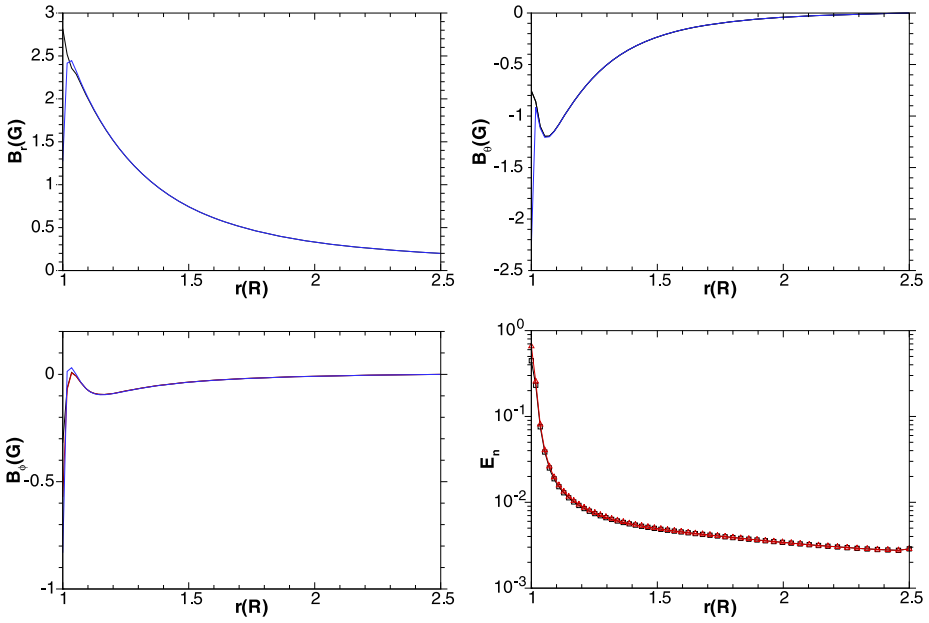

Figure 3 Profiles of field values B_r (top left), B_θ (top right), and B_ϕ (bottom left) in units of gauss (G) at $\theta = 20^\circ$ and $\phi = 50^\circ$ as a function of r. The bottom right panel shows the profile of relative error E_n as a function of r. The blue lines denote the SSW/PFSS solution, the black lines for the MSFD-FD solution, and red lines for MSFD-M solution.

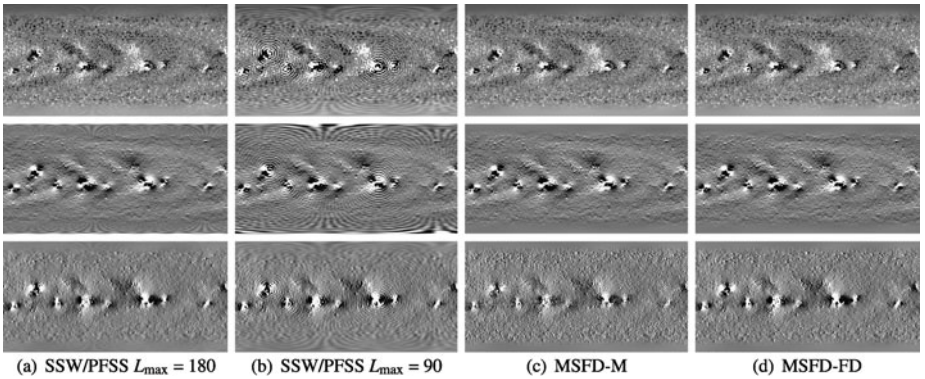

Figure 4 Magnetic maps on the solar surface computed by different methods: (a) the SSW/PFSS software with the expansion order $L_{\max} = 180$; (b) SSW/PFSS with $L_{\max} = 90$; (c) the MSFD-M scheme; (d) the MSFD-FD scheme. The top row is for the radial field component B_r, the second row for B_θ, and the bottom row for B_ϕ. The saturation values for these maps is ± 20 G.

but at the radius of $r_2 \approx 1.035R$ (*i.e.*, two grid points above the photosphere) are plotted, the oscillations almost disappear.

Another point we can notice in Figure 6 is that the SSW/PFSS solution is somewhat smoother than the MSFD solutions, particularly within the active regions. This is probably due to the finite differencing in our method, which can reproduce local sharp gradients (*e.g.*,

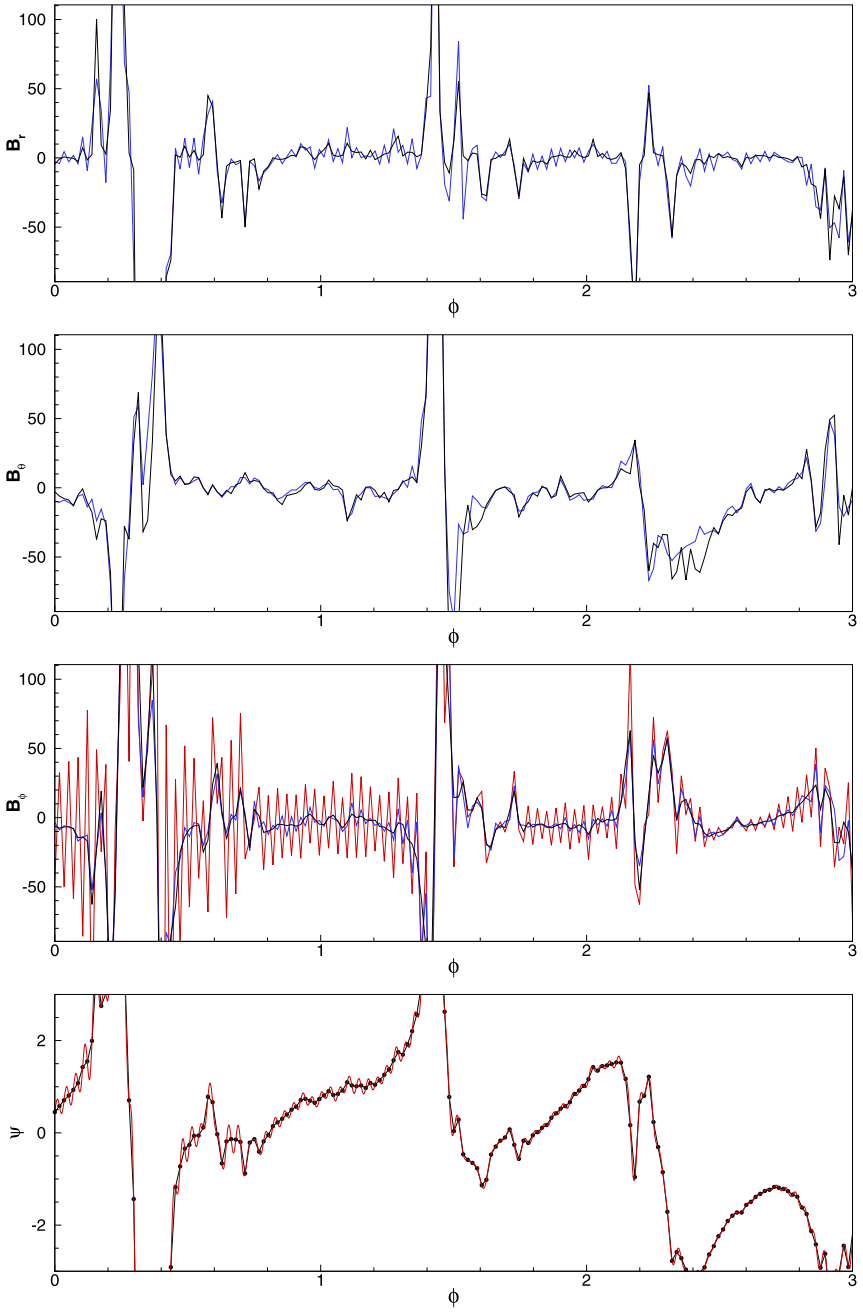

Figure 5 Examples of the solution on the solar surface ($r_0 = 1R$) at a given latitude ($\theta = 104.5°$) as a function of longitude ϕ. In the first three panels, the blue curves represent the SSW/PFSS solution, the red represents the MSFD-M solution, and the black the MSFD-FD solution. The range of the ordinates is limited to ± 100 G to emphasize the numerical artifacts in the solutions. Note that the MSFD-M and MSFD-FD solutions for B_r and B_θ are exactly the same and thus the red lines overlap the black lines in the first two panels. The last panel gives the grid-node solution of ψ (black dots) and its discrete Fourier series (the red curve).

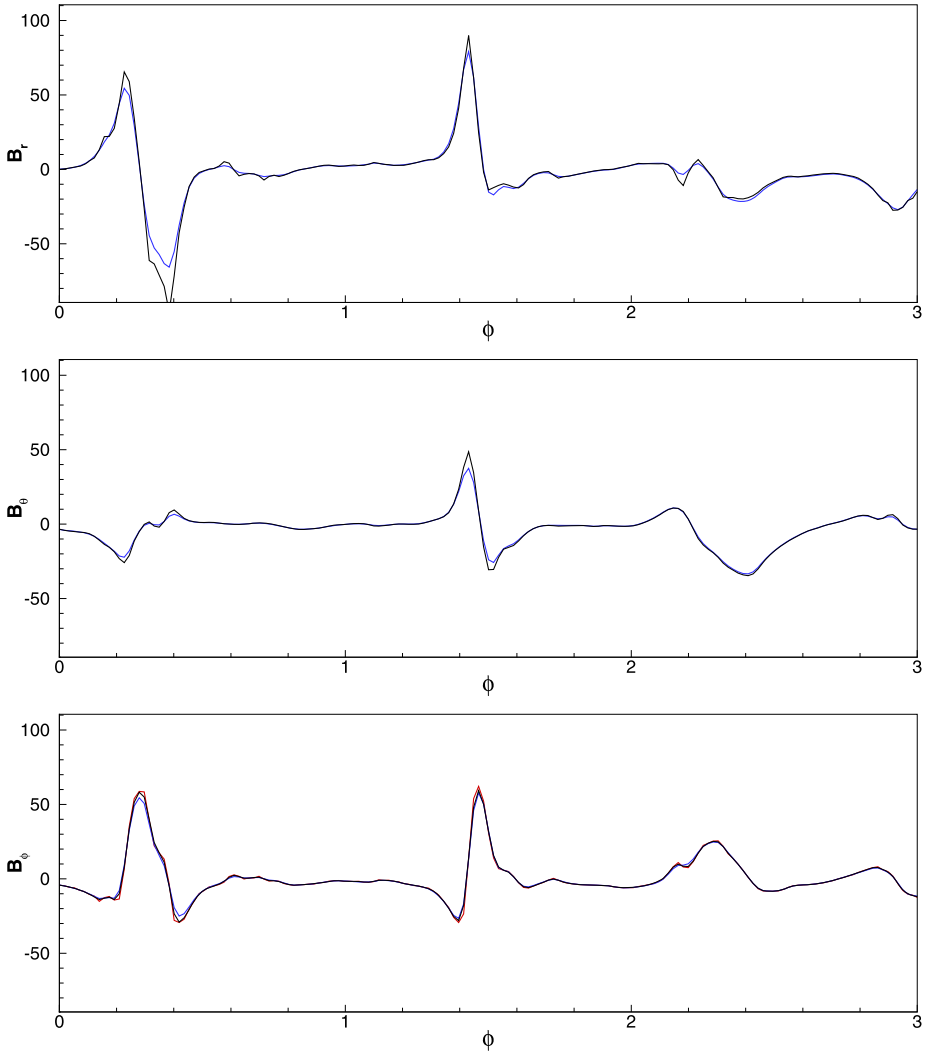

Figure 6 Same as Figure 5 but for the field at radius $r_2 \approx 1.035R$.

discontinuities) more precisely than the full spectral method, which uses superposition of smooth functions to express the solution.

A great advantage of our MSFD solver is its extremely fast computing speed. In Table 2 we list the computing times of a series of tests with increasing mesh resolutions up to $177(r) \times 600(\theta) \times 1200(\phi)$. These tests are performed on a single Intel Xeon CPU E5430 (2.66 GHz). For a relatively low resolution ($\Delta = 2^\circ$) the solver needs only less than one fifth of a second, and for a higher resolution ($\Delta = 0.5^\circ$) it only requires a quarter of a minute. Even for the highest resolution (600×1200) the computation is completed nearly within one minute. In the MSFD method, most of the computing time is consumed by the Helmholtz solver. For each Fourier coefficient ψ_n we need $\mathrm{O}(ML \log_2 L)$ operations and by taking advantage of the conjugate symmetry of the coefficients, *i.e.*, $\psi_{-n} = \overline{\psi_n}$, we have to solve

Table 2 Computing time needed by our MSFD solver and the SSW/PFSS package for different resolutions.

Resolution	CPU time (MSFD)	CPU time (SSW/PFSS)
$28 \times 90 \times 180$	0.16 s	14.20 s
$54 \times 180 \times 360$	1.47 s	3 m 33 s
$107 \times 360 \times 720$	14.64 s	2 h 5 m
$159 \times 540 \times 1080$	54.81 s	
$177 \times 600 \times 1200$	1 m 18 s	

totally $N/2+1$ coefficients. Thus the total number of operations is $\mathrm{O}(\frac{N}{2}ML\log_2 L)$. When the resolution is doubled, the computing time increases about eight times because

$$\frac{N \times 2M \times 2L\log_2 2L}{\frac{N}{2}ML\log_2 L} = 8(1+1/\log_2 L) \approx 8. \tag{30}$$

Table 2 also shows the computing times of SSW/PFSS, and our code performs better. The highest order $L_{\max}$ for each resolution is given by $L_{\max} = 360^\circ/(2\Delta)$. The SPHE method computes relatively fast on low-resolution data, but the CPU time drastically increases for high resolutions. As shown in the table when we double the resolution to 360×720, the computing time increases almost 40 times and we give up the tests of higher resolution using the SSW/PFSS because it may take tens of hours. Since the SSW/PFSS is coded within IDL (Interactive Data Language), it could be much faster if coded using Fortran but a SPHE implementation in Fortran 90 by Tóth, van der Holst, and Huang (2011) still takes tens of minutes to compute the 180×360 solution.

Our method is also much faster than full-3D finite-difference methods, *e.g.*, a code developed by Tóth, van der Holst, and Huang (2011). As reported, their FDIPS solver takes about one hour on a 2.66 GHz Intel CPU even for data of 180×360 points, and it is necessary to use a massively parallel computer for larger magnetograms if one expects to obtain the result in an acceptable amount of time (for example it takes over five hours on 108 CPUs for a high resolution of 540×1200). It is also worth noting that the 3D-difference method with an iterative solver is problem dependent, which means that the convergence speed may also differ for different magnetograms. For instance the computation on a very discontinuous magnetogram is slower than on a much smoothed one. For real solar magnetic fields, small structures with rather strong magnetic gradient usually show up in a high-resolution magnetogram, and thus it will become more unfavorable for the 3D-difference scheme. Our method simply gives the same computing speed for any magnetogram as long as the resolution is fixed, because the linear equation after discretization is solved in a direct way. The fast computing speed without a problem-dependent nature makes our method very suitable for routine pipeline computation for the high-resolution magnetogram, *e.g.*, from SDO/HMI.

Finally in Figure 7 we show the results for LFFF solutions for the same CR2029 magnetogram, with several different values of α. The computing speed for LFFF is the same for PFSS as reported above. These computations are only numerical tests and do not represent the real coronal magnetic field, which is very nonlinear. The field lines are twisted as shown and the open fields begin to wind around the Sun (very clearly if seen in a polar view), which is unreasonable for real fields. Also as expected the twist for positive α is opposite to the results for negative α.

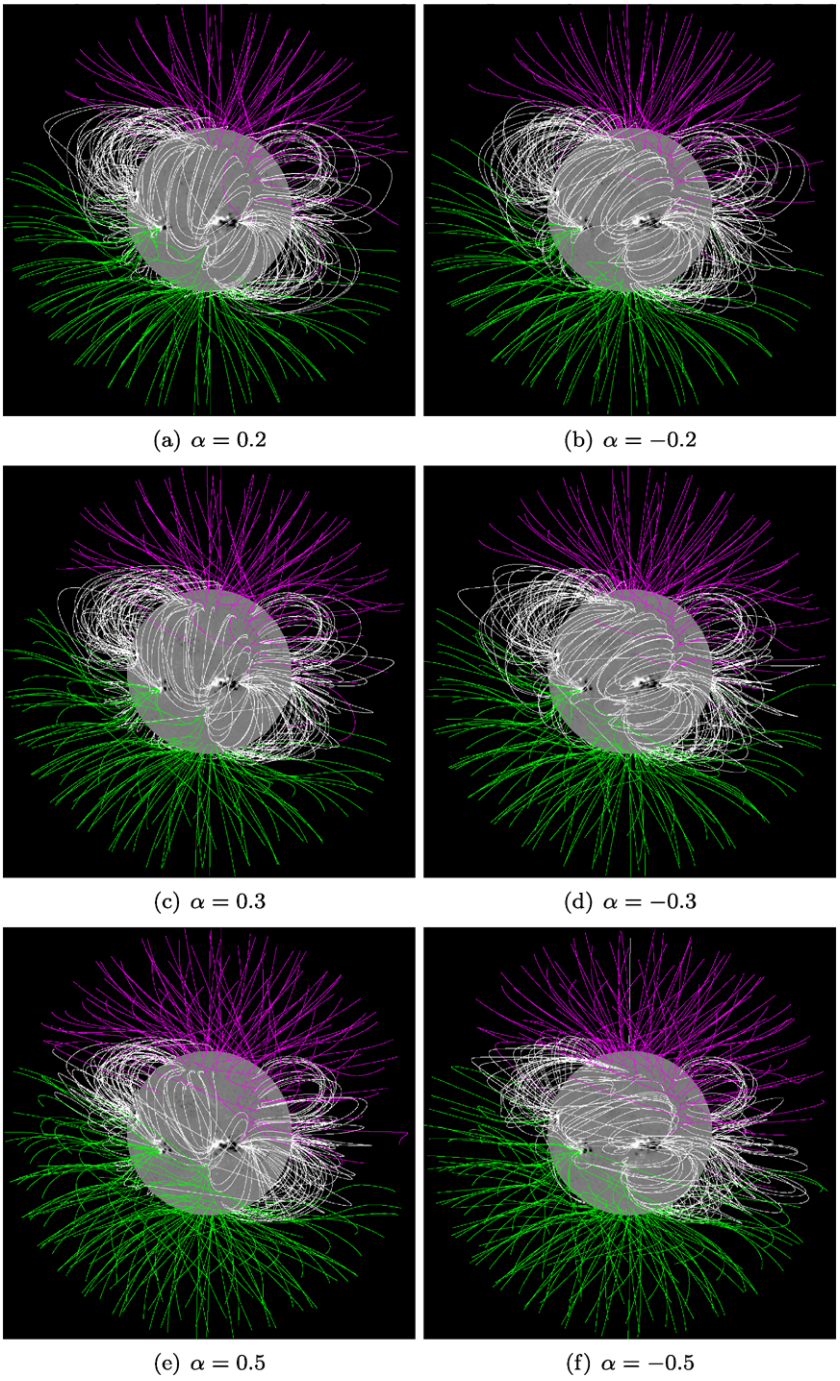

Figure 7 LFFF solutions for the data of Carrington rotation 2029 using different values of α.

5. Conclusions

In this paper, we have developed a fast solver for the global potential fields and linear force-free fields which solves under a common formulation a 3D Helmholtz equation in a spherical shell and a 2D Poisson equation on the photosphere. The solver is based on a combination of the spectral method and the finite-difference scheme. In the longitudinal direction the equation is transformed into the Fourier spectral space and the resulting 2D equations in the r–θ plane for the Fourier coefficients are solved in a direct way. The solver can flexibly handle either uniform or non-uniform mesh in the radial direction. We have validated the solver by comparing the results with the potential-field solution obtained by the SSW/PFSS package on the same grid. The comparison shows that both solvers give quite the same solution above $1.1R$, but slightly different results near the solar surface. Near the solar surface, our solver is more accurate than the SPHE method and the boundary condition is exactly matched. Unlike the full-3D finite-difference iterative solver, our method is problem-independent, which means that it requires the same computing time for any complex magnetogram as long as the resolution is fixed. This solver is capable of solving a LFFF on the same magnetogram with an arbitrarily given force-free parameter α and without any additional computational resource needed.

Our solver shows a distinct virtue of extremely fast speed. The computation on a magnetogram with a size of 180×360 points is completed in less than two seconds. Even on very high-resolution 600×1200 data, the solution can be obtained within only about one minute on a single 2.66 GHz CPU. The computing time scales with the resolution as $\approx \mathrm{O}(MNL)$ and applications to the original resolution of 1080×3600 points of a SOHO/MDI synoptic map can be expected to be processed within one hour. Such high-resolution computation may only be prohibited by the memory limitation of a single computer. The problem can be solved by installing the code on a distributed-memory parallel computer with the message-passing-interface (MPI) library. Such installation can be easily done because the code is simple and the parallelization can be naturally performed among the coefficient equations with each equation handled independently. It is expected that the parallelized solver will be speeded up further and be applied to the original resolution of SDO/HMI for routinely analyzing daily full-disk data. Our code is freely available for download on the website of https://sourceforge.net/projects/glfff-solver or C.W. Jiang's homepage http://www.cwjiang.net/code.

Acknowledgements The work is jointly supported by the 973 program under grant 2012CB825601, the Chinese Academy of Sciences (KZZD-EW-01-4), the National Natural Science Foundation of China (41031066, 40921063, 40890162, and 41074122), and the Specialized Research Fund for State Key Laboratories. We thank M.L. DeRosa for help on the use of the SSW/PFSS package. Special thanks go to our anonymous reviewer for valuable suggestions for the improvement of the paper.

References

Adams, J., Swarztrauber, P., Sweet, R.: 1980, *Fishpack – A Package of Fortran Subprograms for the Solution of Separable Elliptic Partial Differential Equations*, http://www.netlib.org/fishpack.

Altschuler, M.D., Newkirk, G.: 1969, *Solar Phys.* **9**, 131.

Aly, J.J.: 1989, *Solar Phys.* **120**, 19.

Amari, T., Aly, J.J., Luciani, J.F., Boulmezaoud, T.Z., Mikic, Z.: 1997, *Solar Phys.* **174**, 129.

Arge, C.N., Pizzo, V.J.: 2000, *J. Geophys. Res.* **105**, 10465.

Aschwanden, M.J.: 2005, *Physics of the Solar Corona. An Introduction with Problems and Solutions*, 2nd edn., Praxis Publishing Ltd., Chichester, 175.

Bao, S., Zhang, H.: 1998, *Astrophys. J. Lett.* **496**, L43.

Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: 2006, *Spectral Methods: Fundamentals in Single Domains*, Springer, Berlin, 39.
Chiu, Y.T., Hilton, H.H.: 1977, *Astrophys. J.* **212**, 873.
Durrant, C.J.: 1989, *Aust. J. Phys.* **42**, 317.
Feng, X., Yang, L., Xiang, C., Jiang, C., Ma, X., Wu, S.T., Zhong, D., Zhou, Y.: 2012, *Solar Phys.* **279**, 207.
Gary, G.A.: 1989, *Astrophys. J. Suppl.* **69**, 323.
Hoeksema, J.T.: 1984, "Structure and Evolution of the Large Scale Solar and Heliospheric Magnetic Fields", Ph.D. thesis, Stanford Univ., CA.
Hu, Q., Dasgupta, B.: 2008, *Solar Phys.* **247**, 87.
Jiang, C.W., Feng, X.S.: 2012, *Astrophys. J.* **749**, 135.
Jiang, C.W., Feng, X.S., Xiang, C.Q.: 2012, *Astrophys. J.*, in press.
Lai, M., Lin, W., Wang, W.: 2002, *IMA J. Numer. Anal.* **22**, 537.
McClymont, A.N., Jiao, L., Mikic, Z.: 1997, *Solar Phys.* **174**, 191.
Pevtsov, A.A., Canfield, R.C., Metcalf, T.R.: 1995, *Astrophys. J. Lett.* **440**, L109.
Sakurai, T.: 1989, *Space Sci. Rev.* **51**, 11.
Schatten, K.H., Wilcox, J.M., Ness, N.F.: 1969, *Solar Phys.* **6**, 442.
Seehafer, N.: 1978, *Solar Phys.* **58**, 215.
Swarztrauber, P.: 1974, *SIAM J. Numer. Anal.* **11**, 1136.
Tóth, G., van der Holst, B., Huang, Z.: 2011, *Astrophys. J.* **732**, 102.
Tran, T.: 2009, "Improving the Predictions of Solar Wind Speed and Interplanetary Magnetic Field at the Earth", Ph.D. thesis, University of California, Los Angeles.
Van der Vorst, H.: 1992, *SIAM J. Sci. Stat. Comput.* **13**, 631.
Wang, Y., Sheeley, N.R., Jr.: 1990, *Astrophys. J.* **355**, 726.
Wiegelmann, T.: 2008, *J. Geophys. Res.* **113**, 3.