Solar Phys (2011) 273:267–293
DOI 10.1007/s11207-010-9515-6

# VFISV: Very Fast Inversion of the Stokes Vector for the Helioseismic and Magnetic Imager

**J.M. Borrero · S. Tomczyk · M. Kubo · H. Socas-Navarro · J. Schou · S. Couvidat · R. Bogart**

Received: 18 January 2009 / Accepted: 12 January 2010 / Published online: 19 February 2010

**Abstract** In this paper we describe in detail the implementation and main properties of a new inversion code for the polarized radiative transfer equation (VFISV: Very Fast Inversion of the Stokes Vector). VFISV will routinely analyze pipeline data from the Helioseismic and Magnetic Imager (HMI) on-board of the Solar Dynamics Observatory (SDO). It will provide full-disk maps (4096×4096 pixels) of the magnetic field vector on the Solar Photosphere every ten minutes. For this reason VFISV is optimized to achieve an inversion speed that will allow it to invert sixteen million pixels every ten minutes with a modest number (approx. 50)

J.M. Borrero (✉) · S. Tomczyk · M. Kubo · H. Socas-Navarro
High Altitude Observatory, National Corporation for Atmospheric Research, 3080 Center Green CG-1, Boulder, CO 80301, USA
e-mail: borrero@ucar.edu

S. Tomczyk
e-mail: tomczyk@ucar.edu

M. Kubo
e-mail: kubo@ucar.edu

H. Socas-Navarro
e-mail: navarro@ucar.edu

H. Socas-Navarro
Instituto de Astrofísica de Canarias, Avda Vía Láctea S/N, La Laguna 38205, Tenerife, Spain
e-mail: hsocas@iac.es

J. Schou · S. Couvidat · R. Bogart
Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA 94305, USA

J. Schou
e-mail: schou@stanford.edu

S. Couvidat
e-mail: couvidat@stanford.edu

R. Bogart
e-mail: rick@stanford.edu

of CPUs. Here we focus on describing a number of important details, simplifications and tweaks that have allowed us to significantly speed up the inversion process. We also give details on tests performed with data from the spectropolarimeter on-board of the *Hinode* spacecraft.

**Keywords** Magnetic Fields · Photosphere

## 1. Introduction and Motivation

The Solar Dynamics Observatory will be launched from Cape Canaveral in February 2010 on an Atlas V Booster. On-board this satellite there will be several instruments dedicated to the study of the solar photospheric and coronal magnetic fields and their relation to the interplanetary medium, space weather and Earth climatology (Norton *et al.*, 2006). The Helioseismic and Magnetic Imager is an instrument developed at Stanford University, Lockheed-Martin Solar and Astrophysics Laboratory and at the High Altitude Observatory (Graham *et al.*, 2003). HMI consists of a combination of a Lyot filter and two Michelson interferometers. Between the two Michelson interferometers there is a set of three 10th-order retarders: $\lambda/2$, $\lambda/4$ and $\lambda/2$. Located after these, there is a beam splitter that divides the light between two twin 4096×4096 CCD cameras. Each camera will acquire full-disk images of the Sun with a pixel size of 0.5 arc sec.

The first camera, to be referred to as *Doppler camera*, will be devoted to the measurement of the right and left circular polarization, $I \pm V$, at six wavelength positions across the Fe I ($g_{\mathrm{eff}} = 2.5$) 6173.35 Å line every 50 s or less. With this, full-disk maps of the line-of-sight components of the magnetic field and velocity will be produced with a 50 s cadence for helioseismic studies. The second one, hereafter referred to as *vector camera*, will measure the full Stokes vector $\mathbf{I} = (I, Q, U, V)$ at the same wavelengths as the Doppler camera. The cadence in the vector camera will be about 120 s. Borrero *et al.* (2007) have conducted a detailed study of the vector camera performance, and concluded that photon noise and $p$-mode cross-talk between the different Stokes parameters makes the two-minute data inappropriate to determine the magnetic field vector accurately. They suggest averaging the observed Stokes vector every ten minutes before it is inverted.

Therefore our objective has been to create an inversion code for the radiative transfer equation able to process 13.5 million pixels[1] in ten minutes: 22 500 pixels s$^{-1}$. Our task benefits from the fact that each pixel is treated independently, and therefore our problem can be parallelized in a straight-forward fashion, resulting in a time saving which is directly proportional to the number of CPUs employed. Consequently, very fast (3000 – 10 000 pixels s$^{-1}$ inversion strategies such a Principal Component Analysis (PCA; Rees *et al.*, 2000; Socas-Navarro et al., 2001) or Artificial Neural Networks (ANNs; Carroll and Staude, 2001; Socas-Navarro, 2003, 2005) could meet the HMI requirements with a very small number of CPUs. Without entering into details about the different methods, their accuracy and robustness (the interested reader is referred to the recent reviews by Del Toro Iniesta, 2003a and Bellot Rubio, 2006), it is generally acknowledged that these fast inversion techniques retrieve a magnetic field vector that is more suitable for a qualitative analysis such as active region evolution and tracking, global magnetic field appearance. Our aim with HMI is to

[1] The CCD has sixteen million pixels; however, only about 13.5 million will be left after removing off-limb pixels.

provide the magnetic field vector on the solar surface with an accuracy good enough to carry out more quantitative studies. A further concern, from the point of view of HMI characteristics and scientific objectives, is the fact that current implementations of ANNs do not allow to obtain a measure of the goodness of the inversion, *e.g.* $\chi^2$. PCA does retrieve such a measure in terms of the so-called PCA distance. However, Principal Component Analysis is usually employed to reduce the dimensionality of the problem (*e.g.* by inverting the first PCA or Fourier coefficients of the observed Stokes vector). It is unclear how PCA will perform when inverting data from filtergram instruments, such as HMI, where the number of data points is already very small (six points in wavelength across the four Stokes parameters).

A good compromise between accuracy and speed is achieved by traditional iterative nonlinear least-squares fitting algorithms, specially if applied to the Milne – Eddington (M – E) solution of the radiative transfer equation (Landolfi and Landi Degl'Innocenti, 1982). The reason for this lies in the simplified thermodynamics of this approximation, that avoids tedious iterative calculations and evaluations of the partial pressures, ionization and hydrostatic equilibrium, *etc.* (Ruiz Cobo, 2006). In addition, the analytical nature of the M – E solution allows one to calculate also analytical derivatives, which significantly speeds up the inversion process. The first M – E inversion code that fully takes into account magneto-optical effects dates back, to the best of our knowledge, to Auer, House, and Heasley (1977) (see also Skumanich and Lites, 1987). This has been for many years the standard inversion technique employed in the analysis of Advanced Stokes Polarimeter data (ASP; previously Stokes II: Baur *et al.*, 1981), and thus it is widely known as the HAO/ASP inversion code. As a consequence of the increase in the CPU speed since the late 80 s, the HAO/ASP code has gone from inverting several pixels per hour to about 10 pixels $\mathrm{s}^{-1}$. Thus, meeting HMI's goal would require more than 2000 CPUs. Clearly further modifications and refinements over these type of M – E inversion codes are necessary if we are to meet HMI's requirements with a limited number (<60) of CPUs.

After a brief introduction to Stokes inversion (Section 2), we have divided these modifications into three groups: those made on the synthesis module of the code (Section 3); modifications made on the inversion module of the code (Section 4), and implementation of an accurate initial guess model for the inversion process (Section 5). In Section 6 we present a study of the inversion code's profile and speed. Section 7 shows results obtained from the application of VFISV to *Hinode*/SP data and a comparison with the SIR inversion code, and finally, Section 8 summarizes our results.

## 2. Introduction to Stokes Inversions

In this section we will describe very briefly how a Stokes inversion code typically works. The idea is to familiarize the reader with the common procedures and nomenclature to facilitate the understanding of the following sections. More details about this topic can be found in (with different levels of complexity) Ruiz Cobo and Del Toro Iniesta (1992), Frutiger (2000), Del Toro Iniesta (2003b), Borrero (2004) and references therein.

The basic idea of any Stokes inversion code is to iteratively fit the observed Stokes vector at each wavelength position $\mathbf{I}^{\mathrm{obs}}(\lambda) = (I, Q, U, V)$.[2] This fit is done by producing a synthetic Stokes vector $\mathbf{I}^{\mathrm{syn}}(\lambda, \mathcal{M})$, that it is then compared at each wavelength position

[2] $I$ is the total intensity, $Q$ and $U$ are the linear polarization profiles and $V$ is the circular polarization.

with the observed one via the $\chi^2$ of the fit:

$$\chi^2 = \frac{1}{4L-F}\sum_{i=1}^{L}\sum_{j=1}^{4}\big[I_j^{\text{obs}}(\lambda_i) - I_j^{\text{syn}}(\lambda_i,\mathcal{M})\big]^2\frac{w_{ij}^2}{\sigma_j^2} \tag{1}$$

where $L$ is the number of wavelength points observed ($L=6$ in HMI's case) and $F$ is the number of free parameters. Therefore $4L-F$ refers to the number of degrees of freedom in the inversion. $I_j^{\text{obs}}(\lambda_i)$ and $I_j^{\text{syn}}(\lambda_i)$ refer to each of the four Stokes parameters ($j$ index) at each wavelength position ($i$ index) for both the observed and synthetic Stokes vector. Section 4.3 discusses in detail the parameters $\sigma_j$ and $w_{ij}$.

The synthetic profiles depend on a series of model parameters $\mathcal{M}=\{M_f\}$ with $f=1,\dots,F$. These parameters define the physical model that we presume as valid: LTE, non-LTE, multi-component atmospheres, gradients along the line of sight, *etc.* In the case of a Milne – Eddington atmosphere, these typically are: $\mathcal{M}=\{S_0, S_1, \eta_0, a, \Delta\lambda_{\text{D}}, B, \gamma, \Psi, V_{\text{los}}, V_{\text{mac}}, \alpha_{\text{mag}}\}$. The first five of them are the thermodynamic parameters: source function at the observer ($S_0$), gradient of the source function ($S_1$), center to continuum absorption coefficient ($\eta_0$), damping ($a$), and Doppler width of the spectral line ($\Delta\lambda_{\text{D}}$). The next three refer to the three components of the magnetic field vector: strength $B$, inclination with respect to the observer $\gamma$, and azimuth of the magnetic field vector in the plane perpendicular to the observer $\Psi$. The following two are kinematic parameters: line-of-sight velocity of the plasma harboring the magnetic field ($V_{\text{los}}$), and the macroturbulent velocity ($V_{\text{mac}}$; used to model unresolved velocity fields). The last parameter, $\alpha_{\text{mag}}$, is a geometrical parameter that defines what portion of the resolution element is filled with a magnetized plasma.

$\mathbf{I}^{\text{syn}}(\lambda,\mathcal{M})$ is computed in the synthesis module of the inversion code. This module also retrieves the $F$ derivatives of the synthetic Stokes vector with respect to the model parameters (free parameters): $\partial\mathbf{I}^{\text{syn}}(\lambda,\mathcal{M})/\partial M_f$. Once the synthetic Stokes profiles are obtained, we can determine the $\chi^2$ of the fit (Equation (1)). With the derivatives it is possible to compute $\nabla\chi^2$, which is an $F$-dimensional vector where each of its components can be expressed as

$$\nabla\chi_f^2 = \frac{\partial\chi^2(\mathcal{M})}{\partial M_f} = -\frac{2}{4L-F}\sum_{i=1}^{L}\sum_{j=1}^{4}\big[I_j^{\text{obs}}(\lambda_i) - I_j^{\text{syn}}(\lambda_i,\mathcal{M})\big]^2\frac{w_{ij}^2}{\sigma_j^2}\frac{\partial I_j^{\text{syn}}(\lambda_i,\mathcal{M})}{\partial M_f}. \tag{2}$$

We can also compute the modified Hessian matrix: $\mathcal{H}_{\text{mod}}$. This matrix is basically the regular Hessian matrix, where the second-order derivatives of $\chi^2$ are approximated using the first-order derivatives of the Stokes vector, and where diagonal elements are modified by means of a parameter $\epsilon$:

$$\mathcal{H}_{nm}^{\text{mod}} = \frac{\partial^2\chi^2}{\partial M_n\partial M_m} \simeq -\frac{2}{4L-F}\sum_{i=1}^{L}\sum_{j=1}^{4}\frac{w_{ij}^2}{\sigma_j^2}\left[\frac{\partial I_j(\lambda_i,\mathcal{M})}{\partial M_n}\frac{\partial I_j(\lambda_i,\mathcal{M})}{\partial M_m}\right] \quad n\neq m, \tag{3}$$

$$\mathcal{H}_{nn}^{\text{mod}} = (1+\epsilon)\left[\frac{\partial^2\chi^2}{\partial^2 M_n}\right]. \tag{4}$$

According to the recipe of the Levenberg – Mardquart algorithm (Press *et al.*, 1986), using $\nabla\chi^2$ and $\mathcal{H}_{\text{mod}}$ allows us to determine the perturbations, $\delta\mathcal{M}$, that must be applied to the model parameters at iteration $h$, $\mathcal{M}_h$, in order to improve $\chi^2$.

$$\delta\mathcal{M} = -\mathcal{H}_{\text{mod}}^{-1}\nabla\chi^2, \tag{5}$$

$$\mathcal{M}_{h+1} = \mathcal{M}_h + \delta\mathcal{M}. \tag{6}$$

To avoid singularities in the inversion of the modified Hessian matrix (Equation (5)) the Singular Value Decomposition method is often employed. Once the new model parameters $\mathcal{M}_{h+1}$ have been determined, the synthesis module is called again to produce the synthetic Stokes vector and its derivatives, as depicted in Figure 1. Then, a new $\chi^2$ is computed and the iteration continues.

## 3. Synthesis Module

### 3.1. Zeeman Pattern

VFISV will be employed for the automatic inversion of data of the Fe I ($g_{\mathrm{eff}} = 2.5$) 6173.33 Å line (see Table 1). Therefore a straightforward implementation of its Zeeman pattern can be hardcoded into the inversion code, avoiding needless extra calculations using the quantum numbers $L$, $S$ and $J$ of the atomic levels involved.

Fe I 6173.33 Å is a normal Zeeman triplet ($J_{\mathrm{l}} = 1 \rightarrow J_{\mathrm{u}} = 0$) and therefore its Zeeman pattern will be composed by one $\pi$ and two $\sigma$ (red and blue) components. In the absence of LOS velocities the former is unshifted with respect to its central laboratory wavelength $\lambda_0$, whereas the latter are shifted (with respect to $\lambda_0$) by an amount given by $\Delta\lambda_B$:

$$\Delta\lambda_B = \pm 4.66685 \times 10^{-10} g_{\mathrm{eff}} B \lambda_0^2 = \pm 0.044635 \times B \quad \left[\mathrm{m\AA}\right] \tag{7}$$

here the positive and negative signs correspond to the red and blue components. $\Delta\lambda_B$ is the only value that VFISV uses internally. Note that the fact that we are using a Zeeman triplet does not allow VFISV to treat most of the magnetic sensitive spectral lines in the solar spectrum. However it is important to point out that some of the most widely used ones are indeed also Zeeman triplets: Fe I ($g_{\mathrm{eff}} = 3$) 5250.2089 Å with $\Delta\lambda_B = 0.03859 \times B$, Fe I ($g_{\mathrm{eff}} = 2.5$) 6302.4936 Å with $\Delta\lambda_B = 0.04634 \times B$ and Fe I ($g_{\mathrm{eff}} = 3$) 15648.515 Å with $\Delta\lambda_B = 0.34284 \times B$. Therefore, VFISV can be modified to invert those spectral lines by simply changing the value of $\Delta\lambda_B$.

Having the Zeeman pattern hardcoded inside the code does not increase its speed, since this one needs to be calculated only once. The real benefit comes from a simplicity point of view, making the synthesis, and even more the inversion module, easier to write and debug.

### 3.2. Voigt and Faraday – Voigt Functions

The evaluations of the Voigt and Faraday functions are regarded as one of the most computationally expensive operations in the synthesis of solar and stellar spectra. A large number of algorithms with different degrees of sophistication have been proposed, yielding different speeds and accuracies (*e.g.* Letchworth and Benner, 2007). A widely used method is one that obtains the Voigt and Faraday – Voigt functions as the real and imaginary parts of the quotient of two complex polynomials of a certain order (Hui, Armstrong, and Wray, 1977). This procedure yields very high accuracies at reasonable speeds. However, we have found a method, based on the Taylor expansion of the Voigt and Faraday functions that, although less accurate (but still within our requirements) is significantly faster. This method takes advantage of the fact that the derivatives of the Voigt $V(a,u)$ and Faraday $F(a,u)$ functions with respect to the damping $a$ and frequency $u$ have the following analytical form (see *e.g.*

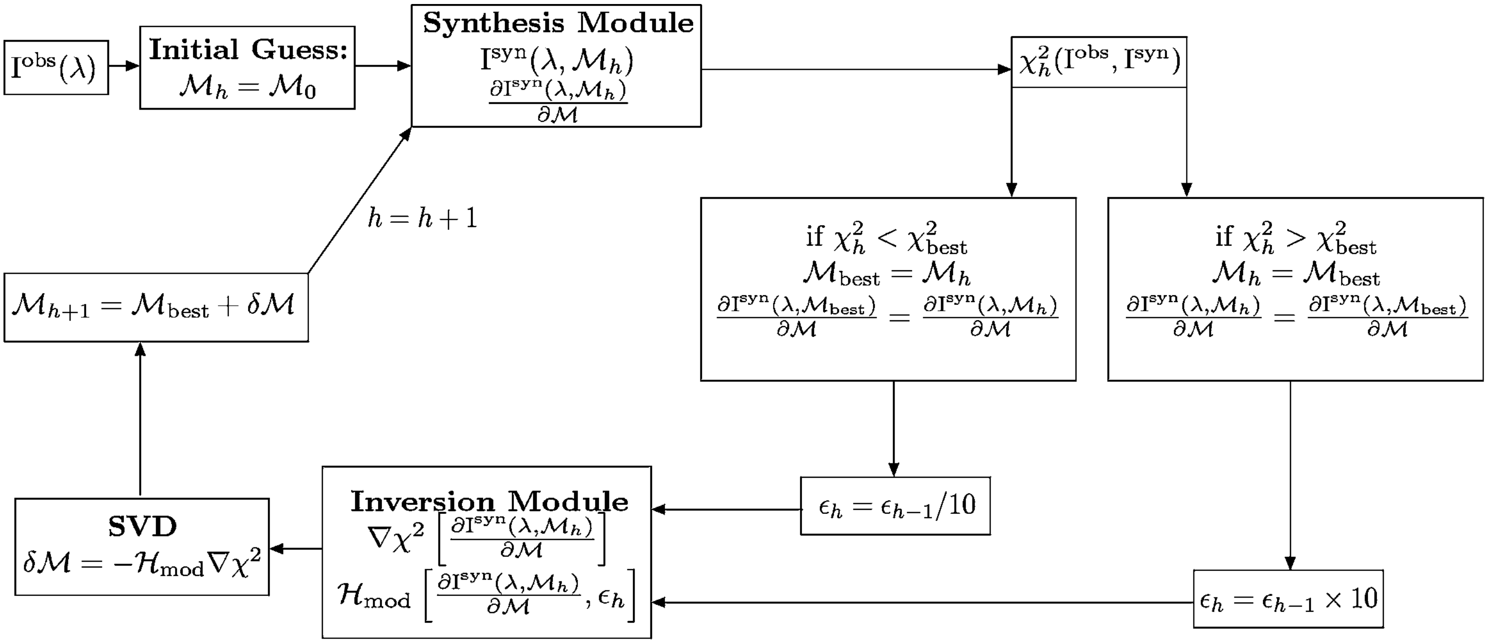
$I^{obs}(\lambda)$
Initial Guess:
$\mathcal{M}_h = \mathcal{M}_0$
Synthesis Module
$I^{syn}(\lambda, \mathcal{M}_h)$
$\frac{\partial I^{syn}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}$
$\chi^2_h(I^{obs}, I^{syn})$
if $\chi^2_h < \chi^2_{best}$
$\mathcal{M}_{best} = \mathcal{M}_h$
$\frac{\partial I^{syn}(\lambda,\mathcal{M}_{best})}{\partial \mathcal{M}} = \frac{\partial I^{syn}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}$
if $\chi^2_h > \chi^2_{best}$
$\mathcal{M}_h = \mathcal{M}_{best}$
$\frac{\partial I^{syn}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}} = \frac{\partial I^{syn}(\lambda,\mathcal{M}_{best})}{\partial \mathcal{M}}$
$\epsilon_h = \epsilon_{h-1}/10$
$\epsilon_h = \epsilon_{h-1} \times 10$
Inversion Module
$\nabla\chi^2 \left[\frac{\partial I^{syn}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}\right]$
$\mathcal{H}_{mod} \left[\frac{\partial I^{syn}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}, \epsilon_h\right]$
SVD
$\delta\mathcal{M} = -\mathcal{H}_{mod}\nabla\chi^2$
$\mathcal{M}_{h+1} = \mathcal{M}_{best} + \delta\mathcal{M}$
$h = h + 1$

**Figure 1** Typical iterative scheme for the inversion of Stokes profiles using the Levenberg – Marquardt algorithm.

**Table 1** Atomic parameters for the lower and upper levels of the atomic transition originating the Fe I 6173.33 Å spectral line (from Nave *et al.*, 1994). The Landé factors have been calculated in the LS approximation (see for example Equation (8.30) in Del Toro Iniesta, 2003b).

| | | $\lambda_0 = 6173.3356$ Å | | | $g_{\text{eff}} = 2.5$ | |
|---|---|---|---|---|---|---|
| Level | Electronic config. | $L$ | $S$ | $J$ | Landé factor | Excitation potential [eV] |
| Lower | $a^5P$ | 1 | 2 | 1 | $g_l = 2.5$ | 2.223 |
| Upper | $y^5D^0$ | 2 | 2 | 0 | $g_u = 0$ | 4.230 |

Del Toro Iniesta, 2003b):

$$\frac{\partial H(a,u)}{\partial a} = -\frac{\partial F(a,u)}{\partial u} = -\frac{2}{\sqrt{\pi}} + 2aH(a,u) + 2uF(a,u) = \alpha, \tag{8}$$

$$\frac{\partial H(a,u)}{\partial u} = \frac{\partial F(a,u)}{\partial a} = 2aF(a,u) - 2uH(a,u) = \beta. \tag{9}$$

With this, it is straightforward to write the second derivatives as

$$\frac{\partial^2 H(a,u)}{\partial a^2} = -\frac{\partial^2 H(a,u)}{\partial u^2} = -\frac{\partial^2 F(a,u)}{\partial a \partial u} = 2H(a,u) + 2a\alpha + 2u\beta, \tag{10}$$

$$\frac{\partial^2 F(a,u)}{\partial a^2} = \frac{\partial^2 H(a,u)}{\partial a \partial u} = -\frac{\partial^2 F(a,u)}{\partial u^2} = 2F(a,u) + 2a\beta - 2u\alpha. \tag{11}$$

Let us now create a two-dimensional table with tabulated values for $H$ and $F$ for different pairs of $(a,u)$. This table can be evaluated using any existing algorithm. In our case we have employed Hui, Armstrong, and Wray (1977). By knowing $H$ and $F$ at those grid points, it is possible to evaluate $H$ and $F$ in the vicinity by extrapolating using a Taylor expansion (second order in this example) around the tabulated value:

$$\begin{aligned} H(a+\mathrm{d}a, u+\mathrm{d}u) \simeq{}& H(a,u) + \alpha\,\mathrm{d}a + \beta\,\mathrm{d}u + \big[H(a,u) + a\alpha + u\beta\big]\big(\mathrm{d}a^2 - \mathrm{d}u^2\big) \\ &+ 2\big[F(a,u) + a\beta - u\alpha\big]\,\mathrm{d}a\,\mathrm{d}u, \end{aligned} \tag{12}$$

$$\begin{aligned} F(a+\mathrm{d}a, u+\mathrm{d}u) \simeq{}& F(a,u) + \beta\,\mathrm{d}a - \alpha\,\mathrm{d}u + \big[F(a,u) + a\beta - u\alpha\big]\big(\mathrm{d}a^2 - \mathrm{d}u^2\big) \\ &- 2\big[H(a,u) + a\alpha + u\beta\big]\,\mathrm{d}a\,\mathrm{d}u. \end{aligned} \tag{13}$$

For a fixed size of the tabulated values of $H(a,u)$ and $F(a,u)$ increasing the order of the Taylor expansion increases of course the accuracy to which $H(a+\mathrm{d}a, u+\mathrm{d}u)$ and $F(a+\mathrm{d}a, u+\mathrm{d}u)$ can be determined. The additional number of mathematical operations also increases the computation time. Figure 2 shows the difference in the determination of $H(a+\mathrm{d}a, u+\mathrm{d}u)$ (at $10^6$ frequency $u$ points for a fixed damping of $a = 1$) between our method and the algorithm by Hui, Armstrong, and Wray (1977). The upper panel shows the results using only a first-order Taylor expansion, whereas the lower panel was obtained using a second-order expansion. The table of tabulated $H(a+\mathrm{d}a, u+\mathrm{d}u)$ values had 500 points in frequency and 100 points in damping. The differences between Hui's algorithm and our approach are always smaller than $10^{-4}$, even for a first-order expansion. As expected the difference drops to zero at the frequencies where the table has been evaluated, and is maximum in between them. Figure 3 shows the time employed to evaluate the Voigt, $H(a,u)$

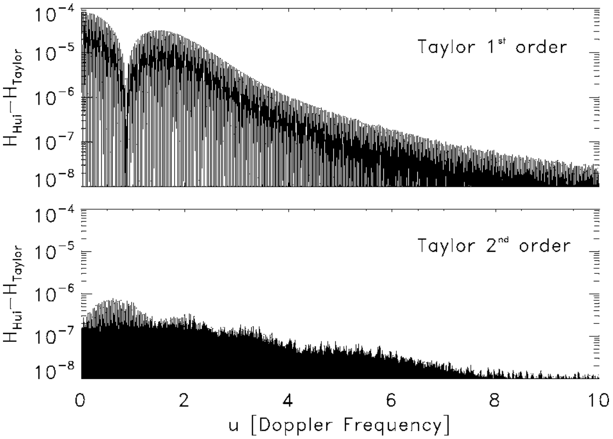

**Figure 2** Top panel: differences between the Voigt function calculated using the algorithm of Hui, Armstrong, and Wray (1977) and the Voigt function calculated with a first-order Taylor expansion. In both cases the Voigt function $H(a,u)$ was evaluated for $10^6$ frequency $u$ points for a fixed damping $a=1$. Lower panel: same but this time using a second-order Taylor expansion.

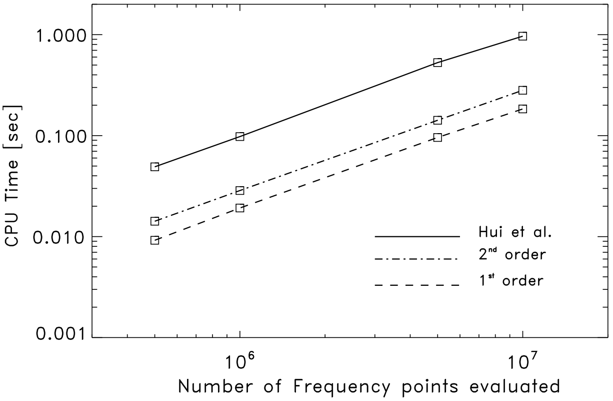

**Figure 3** Time required to evaluate the Voigt and Faraday functions with increasing number of frequency points. Solid line corresponds to the algorithm by Hui, Armstrong, and Wray (1977), whereas dashed and dashed-dotted lines correspond to Taylor expansions of first and second order, respectively.

and Faraday – Voigt $F(a,u)$, functions for different numbers of frequency points. The comparison reveals that using a second-order Taylor expansion decreases the time needed by a factor of 3.4 as compared to Hui, Armstrong, and Wray (1977). The improvement reaches a factor 5.4 when using a first-order Taylor expansion.

We have also studied the performance of interpolating $H(a+\mathrm{d}a, u+\mathrm{d}u)$ and $F(a+\mathrm{d}a, u+\mathrm{d}u)$ in the original table (100 points in damping and 500 in frequency) using the four nearest tabulated points. It turns out that bilinear interpolation is less accurate that using Taylor expansions, because it does not take advantage of Equations (8) and (9). In addition, it is far slower since it requires a larger number of evaluations. In fact it is even slower than the algorithm of Hui, Armstrong, and Wray (1977).

## 4. Inversion Module

### 4.1. Damping and Macroturbulence

The damping parameter $a$ (see Section 2) controls the distance from line center at which the spectral profile shifts from a Gaussian-like (thermal broadening) to a Lorentzian-like profile (radiative and collisional broadening). Even in its classical approximation, $a$ is a complicated function of the temperature and partial pressures (Unsold, 1955; Wittmann, 1974). In the Milne – Eddington approximation the damping is considered to be constant and is treated as a free parameter during the inversion.

The macroturbulent velocity $V_{\mathrm{mac}}$ is another free parameter that accounts for other sources of broadening, such as limited spectral resolution and velocity fields occurring at horizontal scales smaller than the resolution element. This is usually taken into account by convolving the synthetic Stokes profiles with a Gaussian kernel of certain width (measured typically in km s$^{-1}$ instead of wavelength units).

In M – E atmospheres these two parameters are usually degenerated with other thermodynamic quantities: $\eta_0$ and $\Delta\lambda_{\mathrm{D}}$ (Section 2; see Orozco Suárez and Del Toro Iniesta, 2007). This means that these four free parameters have very similar effects on the synthetic Stokes profiles and thus, it is not generally possible to distinguish, for instance, between a large macroturbulence $V_{\mathrm{mac}}$ and a combination of a small $\eta_0$ with a large Doppler width $\Delta\lambda_{\mathrm{D}}$. This situation is even more pronounced when we are dealing with low spectral resolution line profiles ($\simeq$69 mÅ in HMI's case).

VFISV allows one to consider all four of them as free parameters ($a$, $\eta_0$, $\Delta\lambda_{\mathrm{D}}$ and $V_{\mathrm{mac}}$) but in order to speed up the inversion process it is important to see if some of them can be skipped. Besides the immediate benefit in the speed of the calculation of $\mathcal{H}_{\mathrm{mod}}$ (Equations (3) and (4)) and its singular values, not having $a$ as a free parameter increases also the speed at which we can calculate the Voigt function, since $da = 0$ in Equations (12) and (13). In addition, having $V_{\mathrm{mac}} = 0$ during the inversion saves precious time that is otherwise spent in expensive Fast Fourier Transforms. The question is, of course, how does this affect the determination of important quantities such as the three components of magnetic field vector and line-of-sight velocity of the plasma?

To study this we have carried out a series of inversions over spectro polarimetric data of AR 10953 (see also Section 7). This region was observed, on 14 November 2006, with the spectropolarimeter on the Solar Optical Telescope on-board the Japanese spacecraft *Hinode* (Lites, Elmore, and Streander, 2001; Kosugi *et al.*, 2007). Only the Fe I line at 6302.5 Å was inverted (see Section 3.1). We have carried out five different inversions. In the first one, 31 wavelength points around the selected spectral line (with a wavelength separation of 21.5 mÅ) were inverted having both $a$ and $V_{\mathrm{mac}}$ as free parameters. The remaining four inversions were carried out over only six wavelength positions, that were obtained after applying the HMI filters to the original profiles. In each of these three inversions the parameters $a$ and $V_{\mathrm{mac}}$ were fixed or left free.

Using the 31-wavelength inversion as a reference, we have calculated the errors in the three components of the magnetic field vector: $B$ (field strength), $\gamma$ (field inclination) and $\Psi$ (field azimuth) and LOS velocity $V_{\mathrm{los}}$ for the other four inversions, where only six wavelengths were considered. Results are displayed in Figure 4 for $a =$ free and $V_{\mathrm{mac}} =$ free (solid), $a =$ free and $V_{\mathrm{mac}} = 0$ (dotted), $a = 1$ and $V_{\mathrm{mac}} =$ free (dashed) and finally: $a = 1$ and $V_{\mathrm{mac}} = 0$. This figure shows that it is possible to fix $a$ and ignore the macroturbulent velocity (making $V_{\mathrm{mac}} = 0$) without significantly affecting the accuracy in the determination of the magnetic field vector and LOS velocity. It also highlights that main source of errors is the limited spectral sampling: going from 31 to six wavelength positions.

### 4.2. Convergence, Exits and Restarting

As already mentioned in Section 2, the inversion algorithm is based on the Levenberg – Marquardt least-square fitting algorithm. This iterative algorithm combines the *steepest descent* and *Hessian* methods to search for a minimum in the $\chi^2$-surface, both far away and

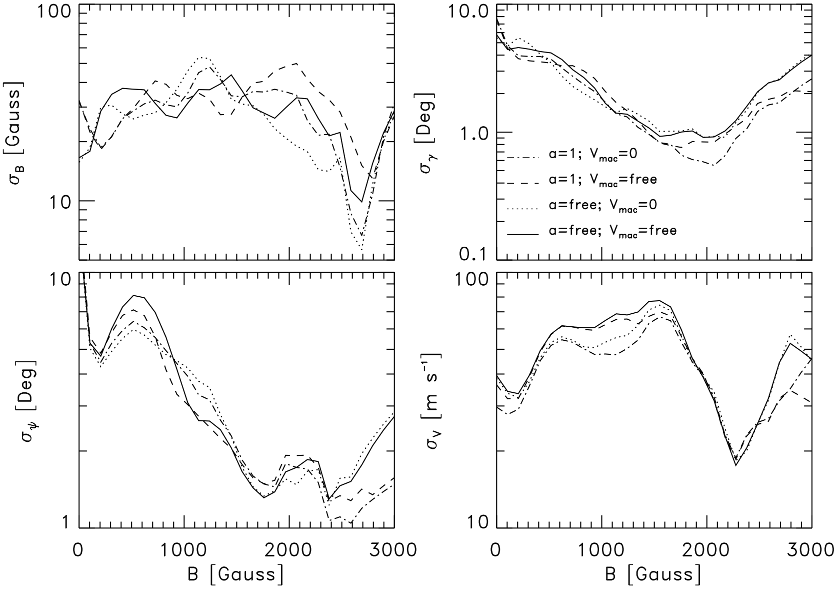

**Figure 4** Errors in the determination of the magnetic field strength $B$ (upper left), magnetic field inclination $\gamma$ (upper right), magnetic field azimuth $\Psi$ (lower left) and line-of-sight velocity $V_{\text{los}}$ (lower right) when the damping $a$ and macroturbulence $V_{\text{mac}}$ are considered free parameters (solid), only $a$ is free (dotted), only $V_{\text{mac}}$ is free (dashed) or neither of them are free parameters (dashed-dotted).

close to this minimum. It uses a modified Hessian matrix $\mathcal{H}_{\text{mod}}$, where the diagonal elements $\mathcal{H}_{nn}^{\text{mod}}$ are modified with a parameter $\epsilon$ (see Equations (3) and (4)) that weighs them more or less, as compared to the non-diagonal elements, whenever the minimum is believed to be far away or close, respectively. The traditional recipe for the Levenberg – Marquardt algorithm (see Press *et al.*, 1986) advises to decrease or increase $\epsilon$ by one order of magnitude, depending upon a successful ($\chi^2$ decreased) or unsuccessful ($\chi^2$ increased) iteration. However, in practice this recipe is not always reliable. For instance, after a number of consecutive successful iterations $\epsilon$ becomes very small. This implies that the modified Hessian matrix corresponds basically to the real Hessian and therefore the algorithm assumes that we are close to the minimum. However, if after this pattern an unsuccessful iteration appears, then $\epsilon$ starts increasing again. The problem is that since $\epsilon$ is so small it will take a large number of extra unsuccessful iterations to bring the modified Hessian matrix to a more diagonal topology, where the algorithm mimics the steepest descent. For this reason, in our implementation, the parameter $\epsilon$ is not always decreased by one order of magnitude after a good iteration. An analogous argument can be made whenever a successful iteration appears after a pattern of bad ones, and therefore we do not always increase $\epsilon$ by one order of magnitude after a bad iteration. Table 2 gives details on how VFISV treats the parameter $\epsilon$ depending on the success of the previous iteration.

Unlike other inversion codes that stop iterating if $\chi^2$ is already small, VFISV performs always a fixed number of iterations.[3] This is done purposely because a small $\chi^2$ does not guarantee that the global minimum in the $\chi^2$-surface has been found (*i.e.*, local minima). In order to minimize the chances of falling into a local minimum we prefer to restart the

[3] This number can be changed by individual users running the code. Currently it is set to 30.

**Table 2** Modification (new) on the $\epsilon$ parameter during the inversion with the Levenberg – Marquardt algorithm according to the previous (old) value and whether the previous iteration was successful, $\chi^2_{\text{new}} < \chi^2_{\text{old}}$, or not, $\chi^2_{\text{new}} > \chi^2_{\text{old}}$.

| $\chi^2$ | $\epsilon_{\text{old}}$ | $\epsilon_{\text{new}}$ |
|---|---|---|
| decreases | $>10^4$ | $\epsilon_{\text{old}}/100$ |
| | $\geq 10^{-4}$ | $\epsilon_{\text{old}}/10$ |
| | $<10^{-4}$ | $\epsilon_{\text{old}}/2$ |
| increases | $<10^{-4}$ | $\epsilon_{\text{old}} \times 100$ |
| | $>10^4$ | $\epsilon_{\text{old}} \times 10$ |
| | $\leq 10^4$ | $\epsilon_{\text{old}} \times 2$ |

inversion using a randomly perturbed model, obtained out of the best model so far, and continue the inversion until the maximum number of iterations has been completed.

This restarting method is performed if: *i*) after the first 10 iterations $\chi^2$ has not decreased by at least one order of magnitude from its first value; *ii*) $\chi^2$ has not improved after five consecutive iterations. The restart can happen more than once, with increasing amplitude of the perturbations each time it happens. This level of randomization enables VFISV to escape from local minima. Of course, it cannot be compared to the more elaborated randomizations performed by other methods (simulated annealing or genetic algorithms), but at least it includes this possibility while retaining the speed of the Levenberg – Marquardt algorithm.

### 4.3. Definition and Normalization of $\chi^2$

The definition of $\chi^2$ adopted is shown in Equation (1). Here we discuss the meaning of $\sigma_i$ and $w_{ij}$, which were left apart intentionally back then. $\sigma_i$ refers to the noise level, which can be different for $I$, $Q$, $U$ and $V$ depending on the polarization calibration and the observing scheme used. The factor $w_{ij}$ is a weighting function that is used to give different weights to the different polarization signals. $w_{ij}$ are invoked because the amplitude of Stokes $I$ signal is typically much larger than that of the Stokes $Q$, $U$ and $V$. If $w_{ij} = 1$ always, it often happens that the derivatives of $\chi^2$, and consequently the Hessian matrix too, are dominated by the derivatives of the Stokes $I$. This leads the inversion code to focus on fitting only this Stokes parameter. For this reason most inversion codes try to compensate for this by making $w_{Q,U} > w_V > w_I$. For example, the SIR inversion code (Ruiz Cobo and Del Toro Iniesta, 1992) adjusts those weights such that the amplitudes in each of the observed Stokes parameter are the same. Unfortunately this procedure makes $\chi^2$ normalization pixel dependent and, thus, makes it impossible to decide whether the fit in a given pixel is better than in another one.

Other inversion codes adopt constant values for $w_{ij}$ such that $\chi^2$ values can be directly compared to each other. In the case of MERLIN (Milne – Eddington gRid Linear Inversion Network) and LILIA (Lites *et al.*, 2007) these are set by default to: $w_{Q,U} = 10w_V = 100w_I$ (linear polarization is given ten times more weight than circular polarization, which is given ten times more weight than total intensity). Since HMI will take full-disk observations, it is very important that $\chi^2$ normalization is as homogeneous as possible, therefore we have decided to follow the same approach as MERLIN, that is, employing constant weights. However, we find that the recipe of $w_{Q,U} = 10w_V = 100w_I$ is somewhat unreliable when inverting pixels inside the sunspot umbra, where the amplitude of all four Stokes profiles is very similar. We have also tested the other possibility (give all four Stokes parameters the same weight: $w_{Q,U} = w_V = w_I$). This approach seems to work well in active regions, but in the quiet Sun the inversion code tends to ignore, as expected, the polarization signals and fit only the intensity $I$. This translates into a pattern for the magnetic field strength and retrieved $\chi^2$ that closely follows the continuum intensity. These results seem to indicate that the larger the continuum intensity the smaller weight should be given to the Stokes $I$.

After several tests (in which many maps from *Hinode*'s spectropolarimeter have been inverted) we have come up with the following empirical recipe. The weights in the polarization signals are left to 1: $w_Q = w_U = w_V = 1$, but the weight in the Stokes $I$ is calculated as

$$w_I = 0.7778 \big\| I_{\text{cont}}(\mu) - I_{\text{cont}}(x, y) \big\| + 0.3, \tag{14}$$

where $I_{\text{cont}}(x, y)$ is the quiet-Sun continuum intensity of a given pixel with coordinates $(x, y)$ on the solar disk. $I_{\text{cont}}(\mu)$ is an estimation of the continuum intensity depending on the $\mu = \cos\Theta$ angle (see Neckel and Labs, 1994; $\Theta$ is the heliocentric angle) at a reference wavelength close to 6173 Å:

$$\begin{aligned} I_{\text{cont}}(\mu) = {} & 0.33644 + 1.30590\mu - 1.79238\mu^2 + 2.45050\mu^3 \\ & - 1.89979\mu^4 + 0.59943\mu^5. \end{aligned} \tag{15}$$

To avoid negative values of $w_I$ whenever the continuum intensity at $(x, y)$ is very large, we are taking the absolute value of the difference between $I_{\text{cont}}(\mu)$ and $I_{\text{cont}}(x, y)$ (see Equation (14)). These formulas ensure that $w_I \simeq 0.3$ in quiet Sun regions, where the continuum intensity is large, but $w_I \simeq 1$ in sunspots. Equations (14) and (15) make the $\chi^2$ normalization pixel dependent, therefore one of our objectives to retain an homogeneous normalization is not totally fulfilled. However, considering that the only one that changes is $w_I$ (the rest are fixed to one) and taking into account that in the worst case scenario $w_I \simeq w_{Q,U,V}$, we can consider that $\chi^2$ normalization changes very slowly in neighboring pixels.

Finally, it is important to mention that VFISV also includes the possibility to assign different weights to different wavelength positions, hence the $w_{ij}$ indexes. The reason for this is that the data compression and transfer is never 100% accurate and therefore there will surely be some missing filtergrams (combinations of the four Stokes parameters at a given wavelength). Although we have not carried out any experiments with it, this will certainly be something to look into during commissioning time.

### 4.4. Derivatives and Hessian

Derivatives of $\chi^2$ with respect to the free parameters (Section 2) are the basic constituents of the divergence vector and Hessian matrix needed in the Levenberg – Marquardt minimization algorithm (Section 2). VFISV calculates these derivatives analytically in its synthesis module, by taking advantage of the fact that the synthetic profiles $I_{\text{syn}}(\lambda)$ are analytical functions of those free parameters. This saves a significant amount of time as compared to other Milne – Eddington inversion codes where derivatives are obtained numerically by calling multiple times the synthesis model, each time slightly modifying the free parameters (*e.g.* MELANIE; see Lites *et al.*, 2007). For instance, calculating numerical derivatives for ten free parameters require twice as many calls to the synthesis module, while having analytical derivatives requires only one call to the synthesis module.

Although in the case of numerical derivatives the synthesis module runs faster (since it does not calculate internally the derivatives), it is not fast enough to compensate the extra number of calls needed. We have confirmed this point by comparing the speed of the

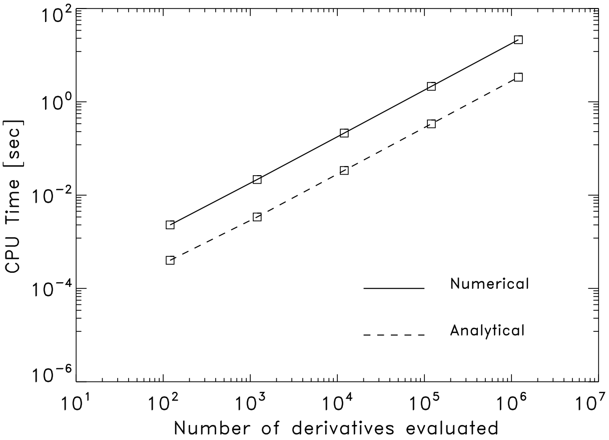

**Figure 5** Time needed by VFISV's synthesis module to compute analytical (dashed line) and numerical derivatives (solid line). The analytical calculation is 6.36 times faster than the numerical one.

derivatives by *i*) calling the VFISV synthesis module and asking it to calculate the analytical derivatives internally, and *ii*) calculating the numerical ones via multiple callings to the synthesis module, while commenting the parts of the code where analytical derivatives are computed. As can be seen in Figure 5, calculating derivatives analytically is almost a factor 6.4 times faster than doing it numerically.

Another place where we have improved significantly the speed of the code is in the way the derivatives are called. As stated in Section 2 (see also Figure 1), some Stokes inversion codes determine the synthetic profiles $I_{\mathrm{syn}}(\mathcal{M}, \lambda)$ and their derivatives $\partial I_{\mathrm{syn}}(\lambda, \mathcal{M})/\partial M_f$ every time the synthesis module is called. It turns out that these derivatives are only needed if the iteration is successful: $\chi_h^2 < \chi_{\mathrm{best}}^2$, otherwise the derivatives employed are those corresponding to the previous best iteration. In a typical inversion about 50% of the iterations are unsuccessful, meaning that about 50% of the derivatives (costly calculated) are unnecessary. In this sense, we have modified VFISV to call the synthesis module twice: once to determine only $I_{\mathrm{syn}}(\mathcal{M}, \lambda)$, and a second time to calculate $\partial I_{\mathrm{syn}}(\lambda, \mathcal{M})/\partial M_f$ only if $\chi^2$ has decreased (see modified scheme in Figure 6).

Another computationally expensive part of VFISV is the calculation, at each iteration step, of the modified Hessian matrix. To avoid this, VFISV does not calculate the modified Hessian matrix whenever we are close to a minimum. Instead, VFISV reuses the Hessian matrix from the previous iteration (if that one was also close to the minimum). This can be done because close to the minimum the modified Hessian matrix represents the curvature of the $\chi^2$-surface, and this curvature is approximately the same around the minimum. This is also indicated in Figure 6 (*cf.* Figure 1).

## 5. Initialization Module

A very critical part in the inversion of Stokes profiles is the selection of a suitable set of model parameters to start the iterative process: $\mathcal{M}_0$ (see Figure 1). Although in general these inversion codes are robust enough (Westendorp Plaza *et al.*, 1998) to guarantee that similar results are achieved regardless of the starting point (specially if the inversion includes some randomization; see Section 4.2), the closer the initial model is to the solution, the fewer iterations will be needed. In a problem such as ours, where speed is critical, it is mandatory to reduce these iterations by determining a good initial model. Of course, whatever method

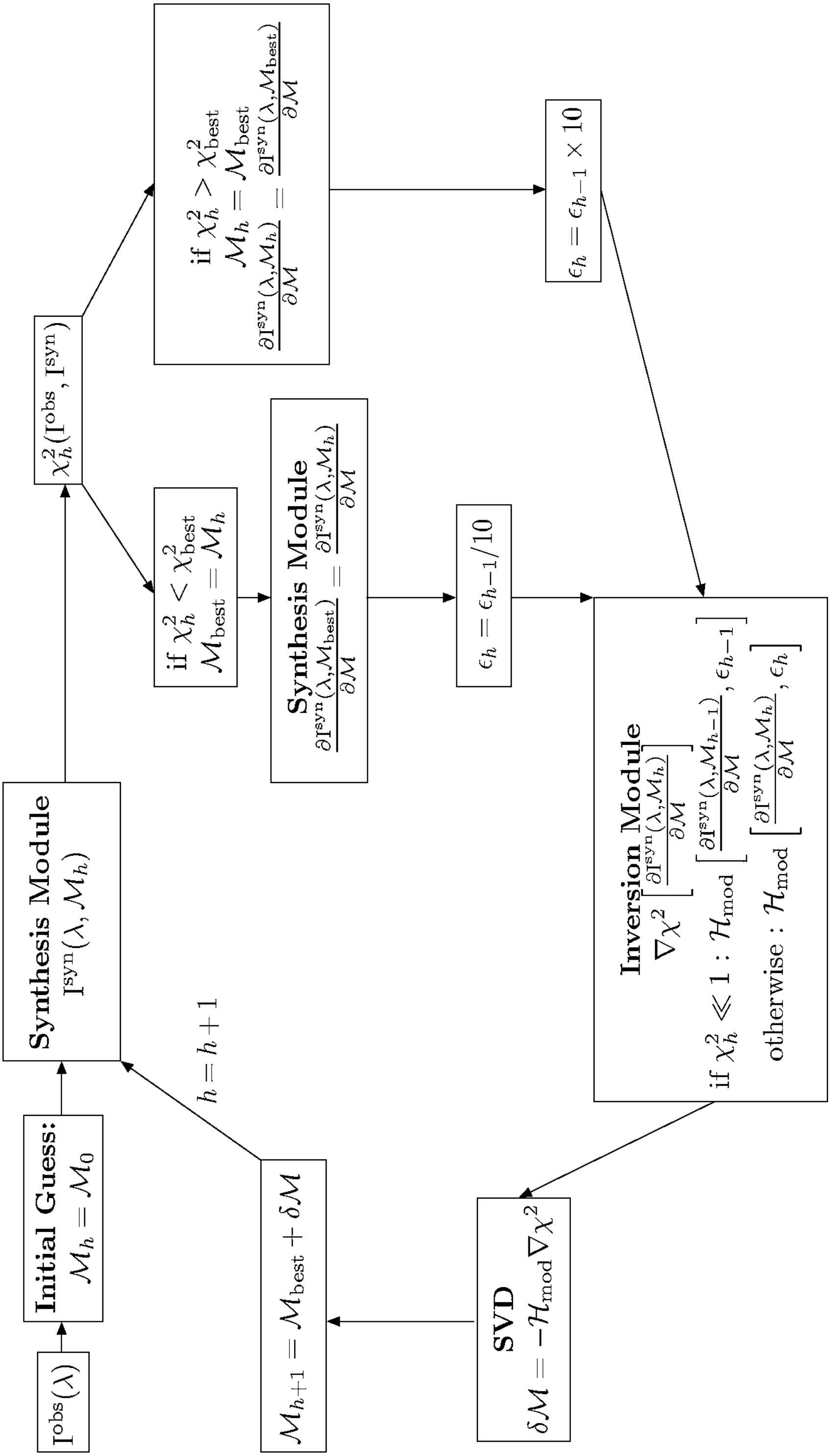
$\mathrm{I}^{\mathrm{obs}}(\lambda)$
**Initial Guess:**
$\mathcal{M}_h = \mathcal{M}_0$
**Synthesis Module**
$\mathrm{I}^{\mathrm{syn}}(\lambda, \mathcal{M}_h)$
$\chi^2_h(\mathrm{I}^{\mathrm{obs}}, \mathrm{I}^{\mathrm{syn}})$
if $\chi^2_h < \chi^2_{\mathrm{best}}$
$\mathcal{M}_{\mathrm{best}} = \mathcal{M}_h$
**Synthesis Module**
$\frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_{\mathrm{best}})}{\partial \mathcal{M}} = \frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}$
$\epsilon_h = \epsilon_{h-1}/10$
if $\chi^2_h > \chi^2_{\mathrm{best}}$
$\mathcal{M}_h = \mathcal{M}_{\mathrm{best}}$
$\frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}} = \frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_{\mathrm{best}})}{\partial \mathcal{M}}$
$\epsilon_h = \epsilon_{h-1} \times 10$
**Inversion Module**
$\nabla\chi^2 \left[\frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}\right]$
if $\chi^2_h \ll 1$ : $\mathcal{H}_{\mathrm{mod}} \left[\frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_{h-1})}{\partial \mathcal{M}}, \epsilon_{h-1}\right]$
otherwise : $\mathcal{H}_{\mathrm{mod}} \left[\frac{\partial \mathrm{I}^{\mathrm{syn}}(\lambda,\mathcal{M}_h)}{\partial \mathcal{M}}, \epsilon_h\right]$
**SVD**
$\delta\mathcal{M} = -\mathcal{H}_{\mathrm{mod}}\nabla\chi^2$
$\mathcal{M}_{h+1} = \mathcal{M}_{\mathrm{best}} + \delta\mathcal{M}$
$h = h + 1$

**Figure 6** VFISV iterative scheme for the inversion of Stokes profiles using the Levenberg – Marquardt algorithm. The derivatives are only computed if $\chi^2$ at iteration $h$ is smaller than any previous value. Otherwise the derivatives from the previous best iteration are used. In addition, the Hessian matrix is not calculated by the inversion module if $\chi^2_h \ll 1$.

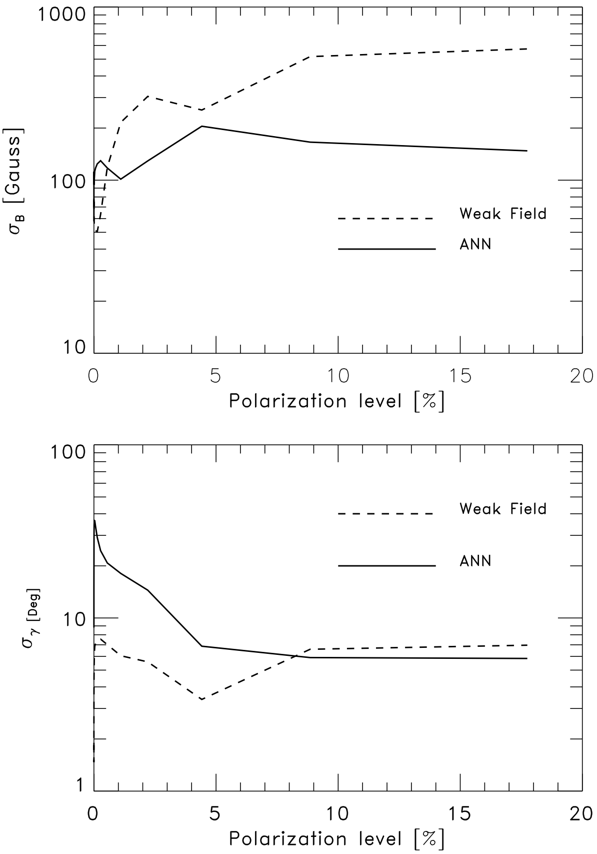

**Figure 7** Top panel: Standard deviation in the magnetic field strength ($\sigma_B$) between the initialized value and the final one obtained from the full inversion, as a function of the polarization level in the observed Stokes profiles. The initializations were obtained with Artificial Neural Networks (solid line) and the Weak Field Approximation (dashed line). Bottom panel: same but for the inclination of the magnetic field $\gamma$.

is used to initialize the full inversion, it must require only a very small fraction of the total inversion time.

VFISV uses two different methods to calculate an initial guess model. The first method, based on Artificial Neural Networks, is applied to determine $B$ and $\gamma$ whenever the total polarization signal $p$ is above 6% (see Equation (3.22) in Del Toro Iniesta, 2003b). The second technique, based on a combination of the magnetograph formula and the Weak Field Approximation (Jefferies and Mickey, 1991), is also applied to $B$ and $\gamma$ if the polarization level is below 6%. This threshold was selected by determining how close each of two different initializations was to the final solution obtained through the full inversion (see Figure 7).

The azimuthal angle $\Psi$ is always determined using the approach proposed by Auer, House, and Heasley (1977). We find this method more reliable than applying the well known formula: $\Psi = \frac{1}{2}\tan^{-1}\frac{U}{Q}$. It is also far superior than results from Neural Networks. Finally, the line-of-sight velocity $V_{\mathrm{los}}$ is always determined using Artificial Neural Networks.

Artificial Neural Networks for $B$, $\gamma$ and $V_{\mathrm{los}}$ were individually trained using a back-propagation method (Bishop, 1994). We employed a three-layered net with 30 neurons per layer. Non-linear transformations were performed between the layers. For each of the training model parameters two different sets of profiles were created: the training and the control set (with 250 000 Stokes profiles each). The Stokes profiles were obtained by producing synthetic profiles using the synthesis module of the SIR inversion code and randomizing all its properties: $V_{\mathrm{mac}}$, $T(\tau)$ (temperature stratification), $B$, $\gamma$, $\Phi$ and $V_{\mathrm{los}}$. The profiles

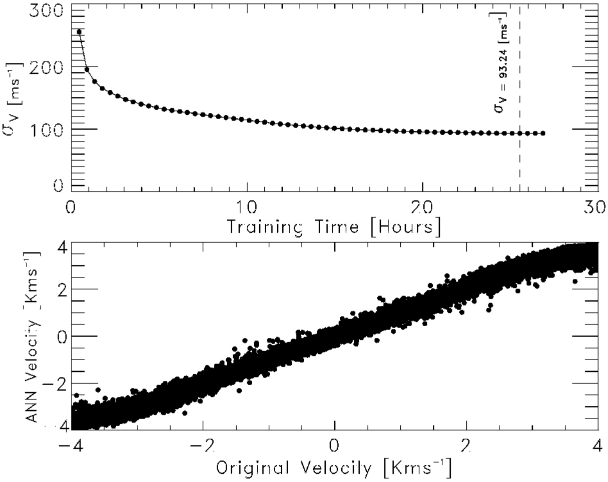

**Figure 8** Top panel: error (standard deviation) in the determination of the line-of-sight velocity $\sigma_v$, using neural networks, as a function of the training time. Bottom panel: scatter plot of the real $V_{\rm los}$ versus the one determined using ANNs.

were convolved with the theoretical HMI filter functions to obtain six wavelength positions. Finally, we also added random noise to the level of $10^{-3}$ to the synthetic Stokes profiles.

Two different sets are needed because the error in the training set always drops to zero with time (the ANN specializes), but in the control set it only decreases initially. The optimum moment to stop the learning process is when the error in the control set starts to increase, as this indicates that Neural Network is loosing generality and is becoming too specialized. Figure 8 (top panel) shows and example of how the error in the determination of the line-of-sight velocity $V_{\rm los}$ decreases with the training time (in the control set). After 24 hours of training $\sigma_v$ started to increased again (see vertical dashed line). At that point the error was $\sigma_v \simeq 93$ m s$^{-1}$. Figure 8 (bottom panel) shows a scatter plot of the velocity in the control set of Stokes profiles and the velocity determined by the Neural Network.

## 6. VFISV Profiling and Speed

In this section we present Tables 3 – 5, where we show the time spent in each of the VFISV modules and subroutines, as well as a brief description. These results correspond to the different levels of optimization described in the previous sections. They were obtained after inverting a sample of 25 FITS files, using a single CPU, from *Hinode*/SP data (a total of 25 600 Stokes profiles). The inversion was performed not considering the damping parameter nor the macroturbulent velocities as free parameters in the inversion, instead they were fixed to $V_{\rm mac} = 0$ and $a = 1$ (see also Section 4.1).

Between Tables 3 and 4 the only change was the optimization in the calculation of the Voigt and Faraday – Voigt functions, which was originally computed using the algorithm by Hui, Armstrong, and Wray (1977) and changed into a second-order Taylor expansion. In the first case, the subroutine *Voigt* took 3.99 seconds, whereas in the second case *Voigt_taylor* needed only 1.30 seconds. This is very close to the factor 3.4 already foreseen in Section 3.2. Note also that, as requested in Section 5, the initialization performed in subroutine *Guess* needs a negligible amount of time compared to the rest of the inversion.

**Table 3** Profiling of the Very Fast Inversion of the Stokes Vector (VFISV) without any optimization. The inversion run at a speed of 479.75 profiles $\mathrm{s}^{-1}$/CPU. The indexes for each subroutine correspond to 1: Computes synthetic Stokes vector and its derivatives; 2: Computes singular values of modified Hessian matrix; 3: Computes elements of absorption matrix and its derivatives; 4: Computes the modified Hessian matrix $\mathcal{H}$; 5: Computes Voigt and Faraday – Voigt functions; 6: Main driver of the program; 7: Computes divergence of $\chi^2$vector: $\nabla\chi^2$; 8: Normalizes derivatives of Synthetic Stokes profiles; 9: Solves linear system of equations (see Equation (5)); 10: Determines initial guess model $\mathcal{M}_0$ (see Figures 1 and 6); 11: Determines $\chi^2$ at each iteration step; 12: Checks for overflows, NaN and Inf.

| Subroutine name | Module | Time [%] | Time [s] | Calls |
|---|---|---|---|---|
| Forward[1] | Synthesis | 38.82 | 18.14 | 742 400 |
| Svdcmp[2] | Inversion | 21.28 | 9.94 | 742 400 |
| Gethess[4] | Inversion | 12.07 | 5.64 | 742 400 |
| Absmat[3] | Synthesis | 10.10 | 4.72 | 742 400 |
| Voigt[5] | Synthesis | 8.53 | 3.99 | 2 227 200 |
| VFISV[6] | MAIN | 2.44 | 1.14 | 1 |
| Getdiv[7] | Inversion | 1.78 | 0.83 | 742 400 |
| Normalize_dsyn[8] | Inversion | 1.26 | 0.59 | 530 336 |
| Svbksb[9] | Inversion | 0.83 | 0.39 | 742 400 |
| Getchi2[11] | Inversion | 0.30 | 0.14 | 742 400 |
| Dnanchk[12] | MAIN | 0.22 | 0.11 | 19 302 400 |
| Guess[10] | MAIN | 0.21 | 0.10 | 25 600 |

**Table 4** Profiling of the Very Fast Inversion of the Stokes Vector (VFISV) including optimization for the Voigt function (see Section 3.2). The inversion run is at a speed of 514.66 profiles $\mathrm{s}^{-1}$/CPU.

| Subroutine name | Module | Time [%] | Time [s] | Calls |
|---|---|---|---|---|
| Forward[1] | Synthesis | 42.86 | 18.71 | 742 400 |
| Svdcmp[2] | Inversion | 21.22 | 9.26 | 742 400 |
| Absmat[3] | Synthesis | 12.74 | 5.56 | 742 400 |
| Gethess[4] | Inversion | 10.67 | 4.67 | 742 400 |
| Voigt_Taylor[5] | Synthesis | 2.98 | 1.30 | 2 227 200 |
| VFISV[6] | MAIN | 2.43 | 1.06 | 1 |
| Getdiv[7] | Inversion | 1.82 | 0.80 | 742 400 |
| Normalize_dsyn[8] | Inversion | 1.26 | 0.55 | 514 146 |
| Svbksb[9] | Inversion | 1.21 | 0.53 | 742 400 |
| Getchi2[11] | Inversion | 0.39 | 0.17 | 742 400 |
| Guess[10] | MAIN | 0.21 | 0.09 | 2 5600 |
| Dnanchk[12] | MAIN | 0.07 | 0.03 | 19 302 400 |

In Table 5 we show a similar profiling where we now also include the optimization in the derivatives and Hessian matrix described in Section 4.4. Note that the Synthesis module (subroutines *Forward* and *Absmat*) are now called about 50% more times (see Figure 6). However, the total amount of time they run was significantly less (about 50% faster). In addition, the Hessian matrix was calculated much less often, which translated into a smaller time running subroutine *Gethess*.

**Table 5** Profiling of the Very Fast Inversion of the Stokes Vector (VFISV) including optimization for the Voigt function (see Section 3.2) and optimization in the derivatives and Hessian matrix (see Section 4.4). The inversion run is at a speed of 686.24 profiles $\mathrm{s}^{-1}$/CPU.

| Subroutine name | Module | Time [%] | Time [s] | Calls |
|---|---|---|---|---|
| Forward[1] | Synthesis | 33.80 | 10.86 | 1 055 429 |
| Svdcmp[2] | Inversion | 28.99 | 9.31 | 742 400 |
| Absmat[3] | Synthesis | 14.85 | 4.77 | 1 055 429 |
| Gethess[4] | Inversion | 5.42 | 1.74 | 296 833 |
| Voigt_Taylor[5] | Synthesis | 4.90 | 1.58 | 3 166 287 |
| VFISV[6] | MAIN | 3.42 | 1.10 | 1 |
| Getdiv[7] | Inversion | 2.62 | 0.84 | 742 400 |
| Svbksb[9] | Inversion | 1.15 | 0.37 | 742 400 |
| Normalize_dsyn[8] | Inversion | 0.84 | 0.27 | 313 029 |
| Getchi2[11] | Inversion | 0.56 | 0.18 | 74 2400 |
| Guess[10] | MAIN | 0.39 | 0.13 | 2 5600 |
| Dnanchk[12] | MAIN | 0.06 | 0.02 | 19 302 400 |

## 7. Testing VFISV

In this section we present results from the application of VFISV to spectropolarimetric data from *Hinode*/SP (Kosugi *et al.*, 2007; Tsuneta *et al.*, 2008; Suetmatsu *et al.*, 2008; Ichimoto *et al.*, 2008; Shimizu *et al.*, 2008). We have used here observations carried out on 14 November 2006. This dataset contains active region (AR 10923), which was located at $\mu = 0.99$. These observations contain 538 FITS files, therefore have 550 912 pixels. Since *Hinode*/SP record the full Stokes vector from the Fe I line pair at 6301.5 and 6302.5 Å we have trimmed the observations to use only 31 wavelength points across the second spectral line (see Section 3.1).

Since VFISV is a code that can be used both for low and high spectral resolution data we have tested our code with two different kind of inversions. The first one has been carried out to showcase the capabilities of VFISV when applied *Hinode*/SP data and to encourage users to employ this inversion code when analyzing data from this instrument. To that end we have performed an inversion of the aforementioned map using *Hinode*/SP full spectral resolution: 31 wavelength positions with a spectral resolution of 21.5 mÅ.

The second type of inversion has been carried out over only six wavelengths points out of the original 31 points across the Fe I 6302.5 Å spectral line. These six wavelength positions have been obtained after applying HMI's filter transmission functions (see Figure 1 in Borrero *et al.*, 2007) to the observed *Hinode*/SP data. This test was done in order to study VFISV's performance when applied to HMI-like data.

### 7.1. Test with *Hinode*/SP Data

In this case, the inversion of the full Stokes vector (on 31 wavelength positions) of 550 912 pixels took just 405.88 seconds running on a Quad-core (4-CPUs) Intel Xeon machine at 2.66 GHz. Thus, inverting 22 500 profiles/seconds (as required for HMI; see Section 1) is possible with only 35 – 40 CPUs. The results from the inversion for the magnetic field strength, inclination and line-of-sight velocity for AR 10923 are presented in Figure 9.

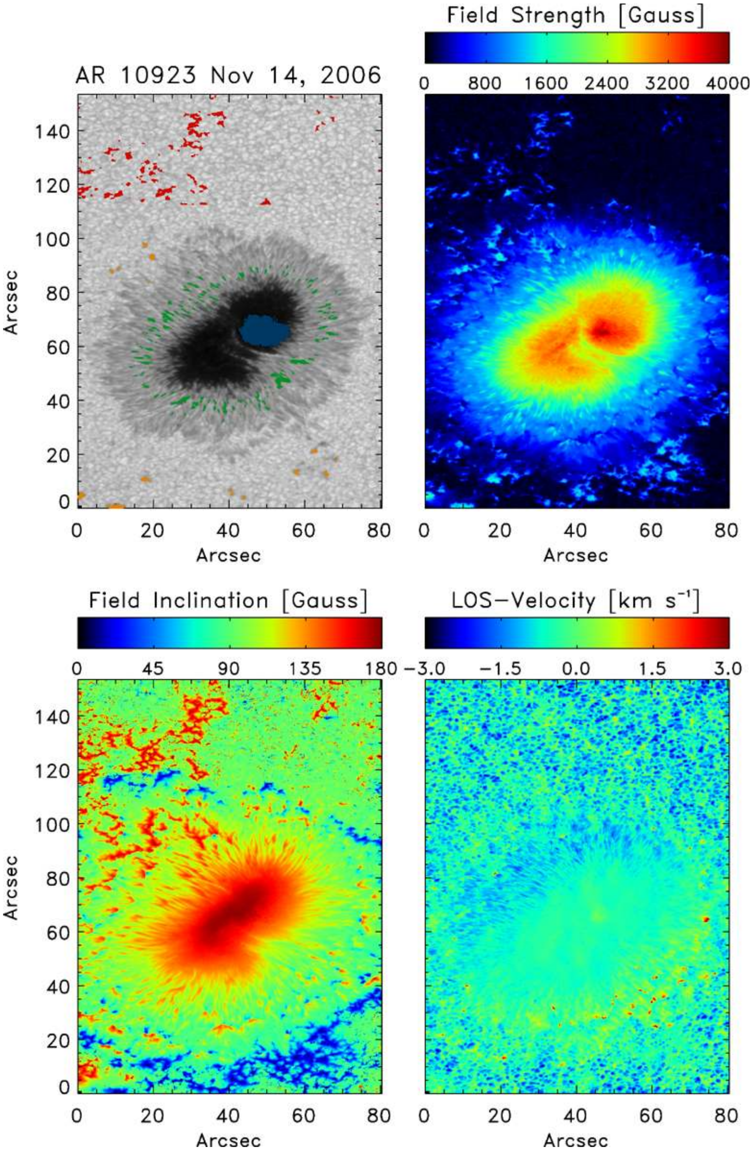

**Figure 9** Results from the inversion, using VFISV, of the spectropolarimetric data from *Hinode*/SP of the active region AR 10923. This region was observed in 14 November 2006 close to disk center: $\mu = 0.99$. Continuum intensity map is displayed on the upper left panel, whereas magnetic field strength, magnetic field inclination and line-of-sight velocity are presented in the upper-right, lower-left and lower-right panels, respectively. In this inversion we used 31 wavelength positions across Fe I 6302.5 Å. The colors in the continuum intensity map refer to pixels selected as representative of: umbra (blue), penumbra (green), pores (orange), and plage (red).

The maps of the different physical parameters indicate that VFISV yields adequate (*i.e.* expected within values in the literature) results in different photospheric structures: network, plage, penumbra and umbra. To illustrate the reliability of the results we show also in

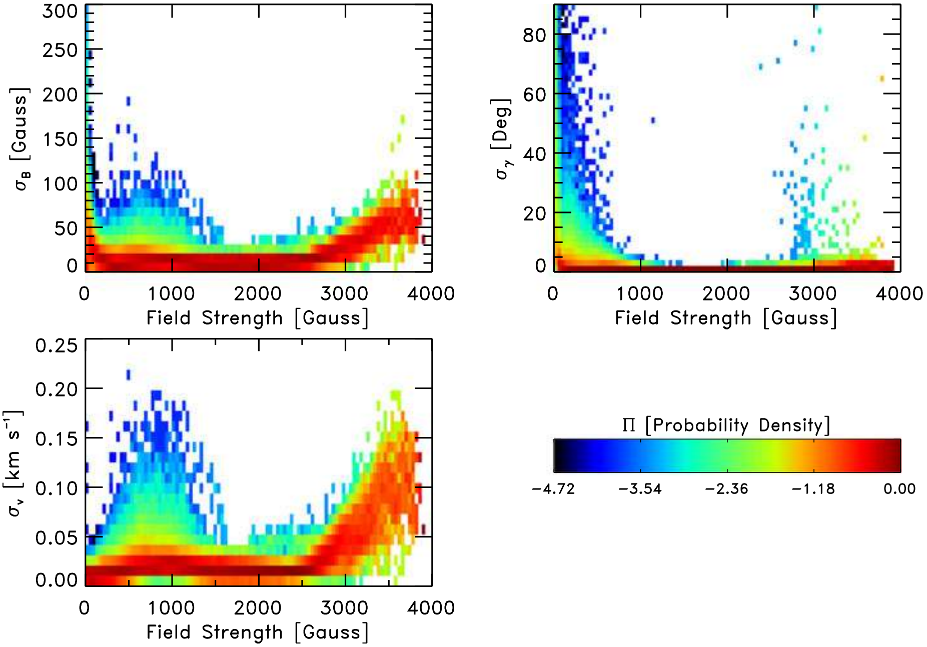

**Figure 10** Probability Densities, $\Pi(B,\sigma)$, for the errors in the magnetic field strength B (upper-left), inclination of the magnetic field with respect to the observer $\gamma$ (upper right), and line-of-sight velocity (lower left), as a function of the magnetic field strength. The errors have been obtained following the formal calculation in Equation (16). Here, the VFISV inversion code was applied to *Hinode*/SP data with full spectral resolution: 31 wavelength points with a resolution of 21.5 mÅ across Fe I 6302.5.

Figure 10, the errors in the magnetic field strength and inclination, as well as line-of-sight velocity, as retrieved by the inversion code. VFISV performs formal error calculations following the approach by Sánchez Almeida (1997). This basically shows that the error, due to the lack of guarantee when reaching the true-$\chi^2$ minimum during the inversion, in the parameter $f$ (see Section 2) is given by the $f$-diagonal element of the inverse of the modified Hessian matrix:

$$\sigma_f^2 = \frac{\chi^2}{F}\left[\mathcal{H}_{\mathrm{mod}}^{-1}\right]_{ff}. \tag{16}$$

Note that this equation is valid both close and far away from the true minimum in the $\chi^2$-surface. Figure 10 actually shows the density distribution function of these errors. The density distributions, $\Pi(B,\sigma)$, have been defined as

$$\Pi(B,\sigma) = \log\left\{\frac{N(\Delta B,\Delta\sigma)}{\int N(\Delta B,\Delta\sigma)\,\mathrm{d}\sigma}\right\}. \tag{17}$$

According to Equation (17), $\Pi(B,\sigma)$ has been obtained by calculating the number of points within one rectangular bin $(\Delta B\,\Delta\sigma)$ and diving it by the total number of points within one field strength bin (*i.e.* total number of points lying across one column). By doing so we remove any selection effects that would arise from having many more points concentrated around one particular regime of field strengths, inclinations, *etc.*. Note that Figure 10 shows a logarithmic color scale, thus red indicates that 70–90% of the pixels within that range of magnetic fields (vertical column) posses that error ($\sigma$), orange: $\approx$10% of pixels, green: $\approx$0.1% of the pixels, and blue: $\approx$0.01%. For instance, Figure 10 (top left panel) shows that the error given by VFISV (Equation (16)) is about $20-30$ gauss in about 90% of the pixels (red region) for all regimes of field strengths, although increases a little bit towards the umbra, $B > 3000$ gauss. At the low end of field strengths, $B < 1000$ gauss, still the vast majority of pixels posses an error of about $20-30$ gauss, but a long tail with larger errors ($\sigma > 70$) and a small amount of pixels appears. The amount of pixels with large errors becomes more and more significant as the field strength weakens.

These formal errors (for each pixel inverted) are a default output from the VFISV inversion code and they should be used as a guidance of the reliability of the inversion at each spatial location. For scientific research these errors should be used instead of the values of $\chi^2$. This is so because small $\chi^2$ values do not directly translate into small errors (Equation (16)). One should indeed consider also the degeneracy and interdependency of the different free parameters of the inversion (given my the inverse of the modified Hessian matrix), and therefore Equation (16) gives a much more reliable tool to estimate those uncertainties.

### 7.2. Test with HMI-like Data

In this case, the inversion of the full Stokes vector (on six wavelength positions) of 550 912 pixels took 387.56 seconds running on a Quad-core (4-CPUs) Intel Xeon machine at 2.66 GHz. The speed gain with respect to the previous case with 31 wavelengths is only marginal. The reason for this is that any speed gained by inversion module, which runs on six wavelength instead of 31, is compensated by a larger computing time taken by the synthesis module. The synthesis module now takes longer time because all calculations are performed over the original wavelength resolution, and then converted into six wavelength points by applying HMI's filters curves. The added *convolutions* is what makes the synthesis module to run significantly slower than before.[4]

The results from the inversion for the magnetic field strength, inclination and line-of-sight velocity for AR 10923 are presented in Figure 11. Visual inspection and comparison with Figure 9 reveals very similar results to the case when the full spectral profile of Fe I 6302.5 Å was used. This supports our argument (see Borrero *et al.*, 2007) that it is still possible to perform reliable inversions with only a few wavelength points. Of course this applies mostly to regions on the solar photosphere with strong magnetic fields: $B > 200$ gauss or so. For weaker magnetic fields (*i.e.* internetwork) we cannot expect VFISV to provide reliable inversions when applied to HMI-like data. Nevertheless, our original intent is to invert all observed pixels on the 4096×4096 CCD regardless of the polarization signal.[5] The reason for that is that there are still physical parameters that can be obtained accurately (*e.g.* line-of-sight velocity) even if the polarization signals are very weak.

Similarly to Figure 10, Figure 12 displays the probability distribution functions of the formal errors (Equation (17)) in the different physical parameters as a function of the magnetic field strength. Note that the errors in all parameters are now larger than in the previous case (Section 7.1). This is due to the smaller number of wavelengths observed in HMI-like data as compared to *Hinode*/SP.

[4] The synthesis module convolves, not only the full Stokes vector evaluated with the original wavelength resolution, but also its derivatives with respect to the free parameters.

[5] This option is still under debate. A final decision will be made only during commissioning time, and once we have the chance to evaluate the quality of the real data and the reliability of the inversion.

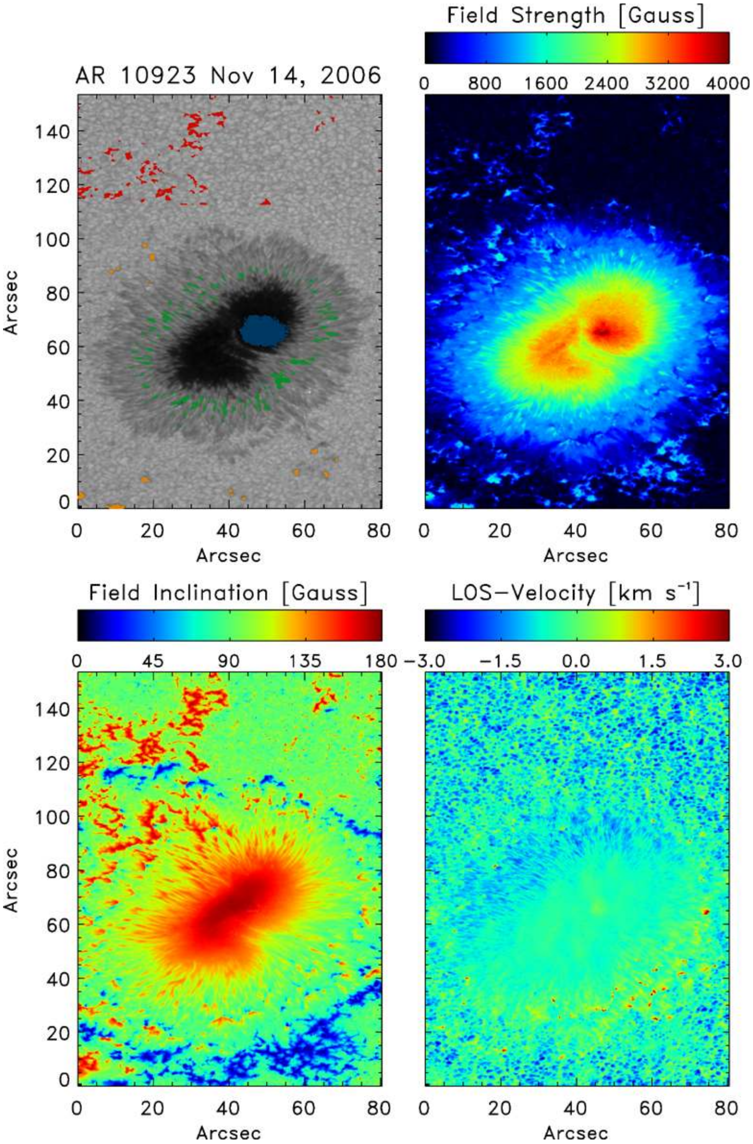

**Figure 11** Same as Figure 9 but inverting the full Stokes vector only at six wavelength positions (HMI-like) across Fe I 6302.5 Å. These six wavelengths were obtained after applying HMI's transmission filters to the original *Hinode*/SP data.

### 7.3. Comparison VFISV/SIR

Until now we have studied the levels of uncertainty in the different physical parameters expected from the application of the VFISV code to *Hinode*/SP and HMI-like data. However, another very important study is to test whether VFISV yields results that are compatible with those from other well tested inversion codes available. To that end we have performed inversions of a subset of AR 10923 (see Figure 9) with the SIR inversion code (Stokes In-

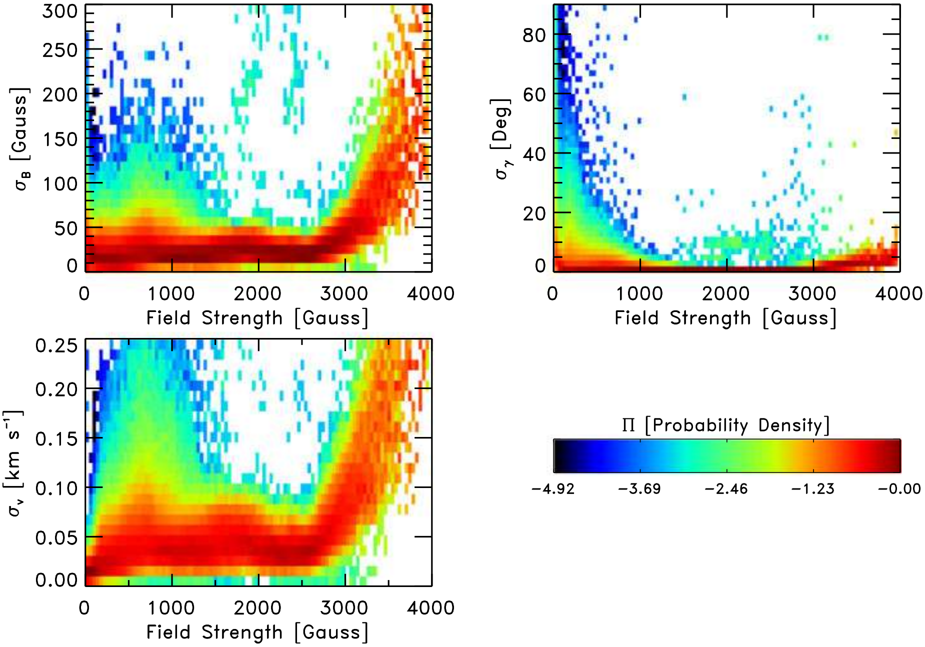

**Figure 12** Same as Figure 10 but inverting only 6 wavelength points (HMI-like) across Fe I 6302.5 Å. Note the larger errors as compared to the case when 31 wavelength points were considered.

version based on Response functions; Ruiz Cobo and Del Toro Iniesta, 1992) and compared them with our results.

The pixels inverted with SIR for this comparison amount to a total of 12 946. The limited amount of pixels was constrained by SIR being about a factor 1000 slower than VFISV. In addition, SIR does not include a proper initialization module as VFISV does (Section 5), and therefore, the likelyhood of SIR not converging was found to be much larger. To avoid this possibility we repeated each inversion with SIR as many as 10 times starting each time from a randomly generated initial model and retaining the one yielding the smallest $\chi^2$. Furthermore, to facilitate the comparison, SIR was set up to simulate the properties of a Milne – Eddington atmosphere: $B$, $\gamma$, $\phi$, $V_{\text{los}}$ were assumed to be constant with optical depth. The temperature stratification, $T(\tau)$ was allowed to change with three nodes. In addition, a macroturbulent $V_{\text{mac}}$, and microturbulent velocities, $V_{\text{mic}}$, were introduced. As explained in Section 4.1 VFISV (despite having this capability) did not consider the macroturbulence nor the damping $a$ as free parameters. This is so because their effect can be efficiently mimicked by the Doppler width $\Delta\lambda_{\text{D}}$ and abortion of the line $\eta_0$. Note that even though SIR can simultaneously fit both Fe I lines in the 630 nm range, we considered only the same spectral region (containing only Fe I 6302.5 Å) as with VFISV. An added complication was to run the inversion with VFISV using different weighting schemes different from the ones described in Section 4.3. Again, this was necessary because each inversion code has its own approach to the weights given to each of the Stokes parameters.

All these complications illustrate the difficulties and caveats of comparing two inversion codes. In fact, to properly carry out this comparison, a much more detailed analysis would be needed. However, here we only aim at showing that results from both inversion codes

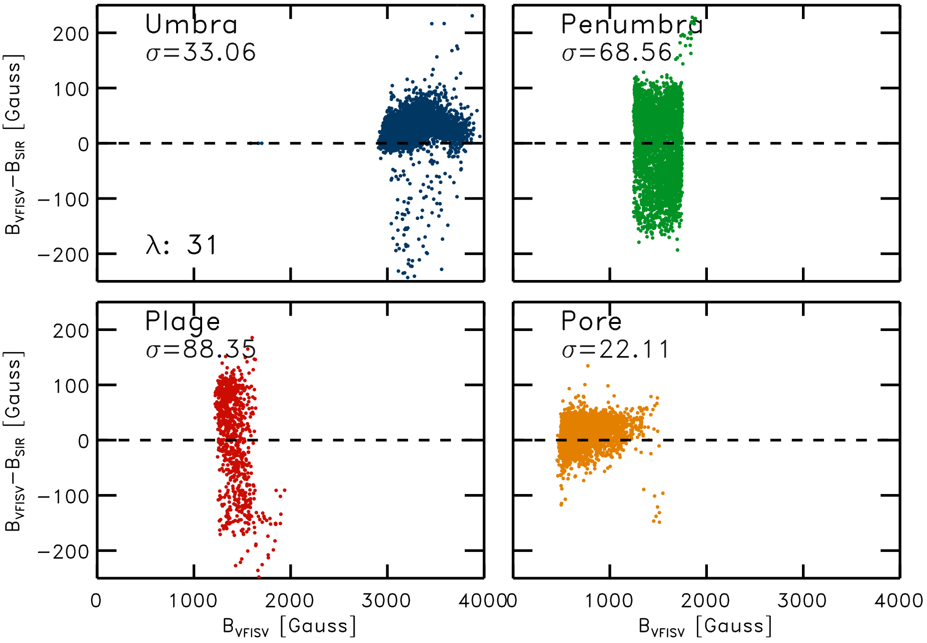

**Figure 13** Differences between the retrieved values for the magnetic field strength between the VFISV and the SIR inversion code as a function of the magnetic field strength. The standard deviation of these differences is also displayed. The color represent the same regions as the upper-left panels in Figures 9 and 11: umbra (blue; upper left), penumbra (green; upper right), plage (red; lower left), and pores (orange; lower right).

are compatible, without attempting to study the sources of the differences nor to minimize them.

The selection of pixels to be inverted with SIR has been done attending to their polarization and continuum intensity levels. We have chosen these such that our subset contains from: umbra (blue; 4558 pixels), penumbra (green; 3782 pixels), plage (red; 3893 pixels) and pores (orange; 916 pixels). The selected pixels have been overplotted with their corresponding colors in Figures 9 and 11.

The comparison between the results yielded by VFISV and HMI can be found in Figures 13, 14 and 15. Despite the obvious differences, caused by the two different kinds of codes employed and by our limited control over SIR's inner workings, these plots demonstrate a large degree of compatibility between SIR and VFISV. In the case of the magnetic field, the standard deviation between SIR's and VFISV's results is below 90 gauss, being much smaller for the case of umbra and pores. The magnetic field inclination retrieved is also highly compatible, with differences (except for the pixels at the pore) below 3 degrees. Finally the line-of-sight velocity is also very similar in both cases, with a standard deviation smaller than 60 $\mathrm{m\,s^{-1}}$ in all cases.

## 8. Conclusions

We have developed an inversion code for the polarized radiative transfer equation. The code's name is VFISV: Very Fast Inversion of the Stokes Vector. This code assumes that

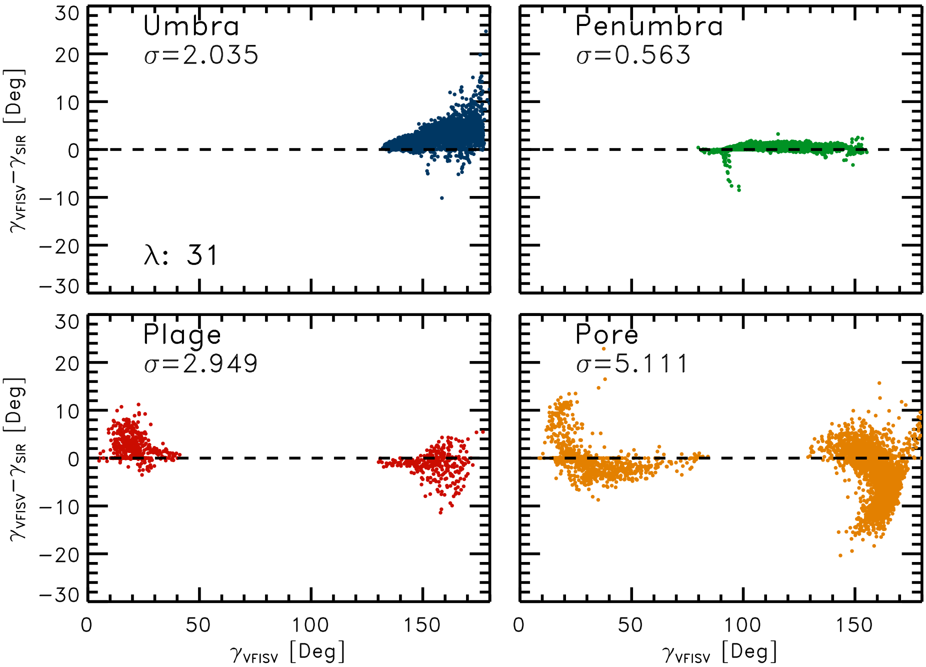

**Figure 14** Same as Figure 13, but for the inclination of the magnetic field with respect to the observer: $\gamma$.

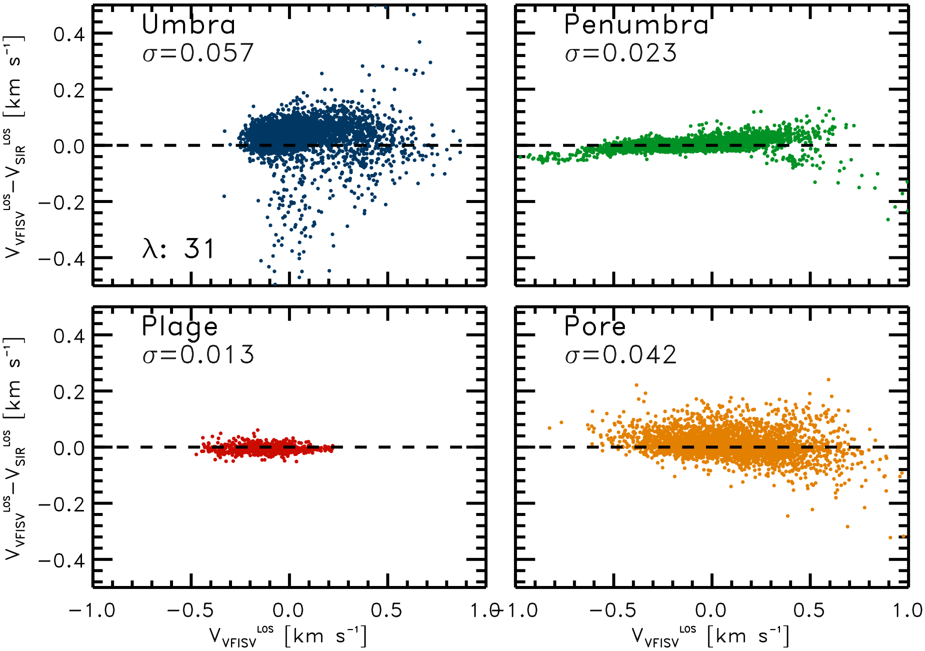

**Figure 15** Same as Figure 13, but for the line-of-sight velocity: $V_{\mathrm{los}}$.

the properties of the solar photosphere are well described by the Milne – Eddington approximation. It will be employed to routinely invert polarimetric data from the Helioseismic and Magnetic Imager (HMI) that will fly on-board of the Solar Dynamics Observatory (SDO) in February 2010. The code is written in Fortran 90 and parallelized with MPICH-2. We have introduced a number of improvements in this code that makes it able to achieve an inversion speed of about 600 pixels per seconds per CPU. At this pace it will be possible to provide solar full-disk maps (with $1''$ resolution) of the solar magnetic field vector every ten minutes, using less than 40 CPUs.

The code has been tested with observations from the spectropolarimeter on-board *Hinode*. It is freely available for download through the Community Spectropolarimetric Analysis Center initiative at the High Altitude Observatory and National Center for Atmospheric Research (Lites *et al.*, 2007; http://www.hao.ucar.edu/projects/csac/).

**Acknowledgements** *Hinode* is a Japanese mission developed and launched by ISAS/JAXA, collaborating with NAOJ as a domestic partner, NASA and STFC (UK) as international partners. Scientific operation of the *Hinode* mission is conducted by the *Hinode* science team organized at ISAS/JAXA. This team mainly consists of scientists from institutes in the partner countries. Support for the post-launch operation is provided by JAXA and NAOJ (Japan), STFC (U.K.), NASA, ESA, and NSC (Norway). This research has made use of NASA's Astrophysics Data System.

## References

Auer, L.H., House, L.L., Heasley, J.N.: 1977, *Solar Phys.* **55**, 47.

Baur, T.G., Elmore, D.E., Lee, R.H., Qerfeld, C.W., Rogers, S.R.: 1981, *Solar Phys.* **70**, 395.

Bellot Rubio, L.R.: 2006, In: Casini, R., Lites, B.W. (eds.) *Solar Polarization 4, ASP Conf. Ser.* **358**, 107.

Bishop, C.M.: 1994, *Neural Networks for Pattern Recognition*, Oxford University Press, London.

Borrero, J.M.: 2004, The fine structure of the sunspot penumbra. Ph.D. Thesis, Göttingen University.

Borrero, J.M., Tomczyk, S., Norton, A., Darnell, T., Schou, J., Scherrer, P., Bush, R., Liu, Y.: 2007, *Solar Phys.* **240**, 177.

Carroll, T.A., Staude, J.: 2001, *Astron. Astrophys.* **378**, 316.

Del Toro Iniesta, J.C.: 2003a, *Astron. Nachr.* **324**, 383.

Del Toro Iniesta, J.C.: 2003b, *Introduction to Spectropolarimetry*, Cambridge University Press, Cambridge.

Frutiger, C.: 2000, Ph.D. Thesis, Institute of Astronomy, ETH-Zurich.

Graham, J.D., Norton, A., López Ariste, A., Lites, B., Socas-Navarro, H., Tomczyk, S.: 2003, In: Trujillo-Bueno, J., Sánchez Almeida, J. (eds.) *Solar Polarization, ASP Conference Series* **307**, 131.

Hui, A.K., Armstrong, B.H., Wray, A.A.: 1977, *J. Quant. Spectrosc. Radiat. Transf.* **18**, 509.

Ichimoto, K., Lites, B.W., Elmore, D., Suematsu, Y., Tsuneta, S., Katsukawa, Y., *et al.*: 2008, *Solar Phys.* **249**, 233.

Jefferies, J.T., Mickey, D.L.: 1991, *Astrophys. J.* **372**, 694.

Kosugi, T., Matsuzaki, K., Sakao, T., Shimuzu, T., Sone, Y., Tachikawa, S., *et al.*: 2007, *Solar Phys.* **243**, 3.

Landolfi, M., Landi Degl'Innocenti, E.: 1982, *Solar Phys.* **78**, 355.

Letchworth, K.L., Benner, D.C.: 2007, *J. Quant. Spectrosc. Radiat. Transf.* **107**, 173.

Lites, B.W., Elmore, D.F., Streander, K.V.: 2001, In: Sigwarth, M. (ed.) *Advanced Solar Polarimetry, ASP Conference Series* **236**, 33.

Lites, B., Casini, R., Garcia, J., Socas-Navarro, H.: 2007, *Mem. Soc. Astron. Ital.* **78**, 148.

Nave, G., Johansson, S., Learner, R.C.M., Thorne, P.: 1994, *Astrophys. J. Suppl.* **94**, 221.

Neckel, H., Labs, D.: 1994, *Solar Phys.* **153**, 91.

Norton, A.A., Pietarila, Graham J., Ulrich, R.K., Schou, J., Tomczyk, S., Liu, Y., *et al.*: 2006, *Solar Phys.* **239**, 69.

Orozco Suárez, D., Del Toro Iniesta, J.C.: 2007, *Astron. Astrophys.* **462**, 1137.

Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: 1986, *Numerical Recipes – The Art of Scientific Computing*, Cambridge University Press, Cambridge, 387.

Rees, D.E., López Ariste, A., Thatcher, J., Semel, M.: 2000, *Astron. Astrophys.* **355**, 759.

Ruiz Cobo, B.: 2006, In: Kneer, F., Puschmann, K.G., Wittmann, A.D. (eds.) *Modern Solar Facilities – Advanced Solar Science*, Universitätsverlag, Göttingen, 287.

Ruiz Cobo, B., Del Toro Iniesta, J.C.: 1992, *Astrophys. J.* **398**, 375.

Sánchez Almeida, J.: 1997, *Astron. Astrophys.* **491**, 993.

Shimizu, T., Nagata, S., Tsuneta, S., Tarbell, T., Edwards, C., Shine, R., *et al.*: 2008, *Solar Phys.* **249**, 221.

Skumanich, A., Lites, B.W.: 1987, *Astrophys. J.* **322**, 473.

Socas-Navarro, H.: 2003, *Neural Netw.* **16**, 355.

Socas-Navarro, H.: 2005, *Astrophys. J.* **621**, 545.

Socas-Navarro, H., López Ariste, A., Lites, B.W.: 2001, *Astrophys. J.* **553**, 949.

Suematsu, T., Tsuneta, S., Ichimoto, K., Shimizu, T., Otsubo, M., Katsukawa, Y., *et al.*: 2008, *Solar Phys.* **249**, 197.

Tsuneta, S., Ichimoto, K., Katsukawa, Y., Nagata, S., Otsubo, M., Shimizu, T., *et al.*: 2008, *Solar Phys.* **249**, 167.

Unsold, A.: 1955, *Physik der Sternatmospharen*, Springer, Berlin.

Westendorp Plaza, C., del Toro Iniesta, J.C., Ruiz Cobo, B., Martínez Pillet, V., Lites, B.W., Skumanich, A.: 1998, *Astrophys. J.* **494**, 453.

Wittmann, A.: 1974, *Solar Phys.* **35**, 11.