

New generation Stokes inversion codes

A. Asensio Ramos^{1,2} and J. de la Cruz Rodríguez³

¹Instituto de Astrofísica de Canarias, 38205, La Laguna, Tenerife, Spain

²Departamento de Astrofísica, Universidad de La Laguna, E-38205 La Laguna, Tenerife, Spain

³Institute for Solar Physics, Dept. of Astronomy, Stockholm University, Albanova University Center, 10691 Stockholm, Sweden

Abstract. We utilize the concept of sparsity or compressibility to develop a new generation of inversion codes for the Stokes parameters. The inversion code uses numerical optimization techniques based on the idea of proximal algorithms to impose sparsity. In so doing, we allow to exploit the presence of spatial correlation on the maps of physical parameters. Sparsity also regularizes the solution by reducing the number of unknowns. The solution has an increased robustness.

Keywords. Sun: magnetic fields, atmosphere, line: profiles, methods: data analysis

1. Introduction

An inversion code is a computer program based on algorithms that allows the user to extract information about the parameters of a physical system from the interpretation of observables. In solar physics, nonlinear inversion codes are routinely applied after their development in the '70s to get the thermodynamical and magnetic properties of different regions of the solar atmosphere from the interpretation of the Stokes parameters (Harvey *et al.* 1972, Auer, Heasley & House 1977 or Skumanich & Lites 1987).

The appearance of the first powerful computers allowed researchers to apply optimization techniques for nonlinear functions and use more elaborate models. A χ^2 merit function (a direct consequence of the assumption that the observations are corrupted with additive Gaussian noise) is minimized with respect to the physical parameters defining the specific model. The first efforts (with the application of the computers available at that time) made use of the Milne-Eddington (ME) approximation to analytically solve the radiative transfer equation (Landi degl'Innocenti & Landolfi 2004). Although the simplifying assumptions that one needs to use when using the ME approximation may not be fully fulfilled in real solar plasmas, it is still one of the most widely used models, in part because of its simplicity. This simplicity leads to very fast inversion codes that can be applied to the large number of observations that we currently obtain. State-of-the-art inversion codes such as VFISV (Borrero *et al.* 2010), used for inferring magnetic field vectors from the Helioseismic and Magnetic Imager (HMI; onboard the Solar Dynamics Observatory) data, MILOS (Orozco Suárez *et al.* 2007) and MERLIN (Lites *et al.* 2007), currently applied to data from the Hinode spacecraft, or the codes based on look-up tables and the principal component analysis (PCA) decomposition (Rees *et al.* 2000) used for the inversion of THEMIS data, are based on the ME approximation.

The availability of more powerful computers in recent years allowed us to use more complex and more realistic models. One of the essential ingredients of this revolution was the application of the idea of response functions (Landi Degl'Innocenti & Landi Degl'Innocenti 1977) to the inversion of Stokes profiles with non-trivial depth stratifications of the physical quantities. The first representative of this family of codes was

SIR (Stokes Inversion based on Response functions; Ruiz Cobo & del Toro Iniesta 1992). Such evolution occurred naturally at that time because observations were showing strong asymmetries in the Stokes profiles in magnetized regions. The explanation of such asymmetries requires the presence of gradients along the line-of-sight (LOS) of the physical properties. Another representative of these inversions code is SPINOR (Frutiger *et al.* 2000). Based on the same strategy, Socas-Navarro *et al.* (2000) developed the NICOLE code (Socas-Navarro *et al.* 2014), capable of dealing with lines in NLTE (non-local thermodynamical equilibrium). This model has been mainly applied for the inversion of Ca II infrared triplet lines, which are formed under strong NLTE conditions (Socas-Navarro, Trujillo Bueno & Ruiz Cobo 2000, de la Cruz Rodríguez *et al.* 2012).

Another step forward in the field of inversion codes was carried out by Asensio Ramos, Martínez González & Rubiño Martín (2007), who introduced Bayesian inference for spectropolarimetric observations. This allows the user to obtain posterior probability distributions for any model parameter and their correlation with the remaining ones. This probabilistic inference is extremely powerful but requires a huge effort in terms of computational power.

With the use of fast slit spectropolarimeters and also two-dimensional filterpolarimeters, we routinely have 2D maps of regions in the solar atmosphere with the four Stokes parameters observed at several points along one or several spectral lines. This rate of new high-quality 2D observations will increase in the future with the advent of bi-dimensional spectropolarimeters based on image slicers or optical fibers. The interpretation of these observations have been done in the past by assuming that all pixels are completely unrelated and applying the inversion codes in a pixel-by-pixel basis. After this pixel-by-pixel inversion, the spatial smoothness of the derived quantities is taken as an indication of the success of the inversions. Salt-and-pepper noise present in the inverted maps of physical parameters is an indication of problems: either the required information cannot be extracted from the Stokes profiles, or the inversions failed to converge to a good solution.

It has become more and more evident that two-dimensional observations and the ensuing inversions are needed to fully understand the physical processes in the solar atmosphere. This is the case even when one assumes local thermodynamical equilibrium (LTE), which relates radiation to the local properties of the plasma. The first representatives of such approach are the already described codes of van Noort (2012) and Ruiz Cobo & Asensio Ramos (2013), where the point spread function of the telescope couples the observed Stokes profiles of nearby pixels.

2. The general sparsity prior

We develop the general idea of regularized inversion codes, with the regularization being based on the idea of *sparsity*. Sparsity or compressibility idealizes the concept that the data can be projected to a parameter space where a *reduced* set of variables can be used to fully describe that dataset. The motivation for this regularization resides on a very simple observation. When one saves a continuum image as a raw file (for instance, a standard 512×512 pixel image), the size of the file is roughly 1 MB (using 4 bytes per pixel). The same image compressed using a lossless file format reduces the size by a factor 3-4, while a lossy format can go further and increase the compression ratio to a factor ~ 10 . In general, the compression is possible because of the existence of spatial correlation on the image. With that term we usually refer to properties like smoothness, the presence of edges, repeated structures in the image, etc. If appropriately exploited by a suitable algorithm, it is possible to predict the value of a certain pixel from the values of other pixels thus making it unnecessary to store the value of all pixels. This is the

case, for instance, of the JPEG compression algorithm (or the more modern JPEG2000 version). In this algorithm, the image is divided in 8×8 portions. The discrete cosine transform (DCT) is obtained for each portion (a specific wavelet transformation is used in the JPEG2000 version of the algorithm) and only a fraction of the coefficients is saved. The uncompressed image is obtained by reversing the algorithm.

Helped with these ideas, let us assume that we have observed the Stokes parameters on a 2D grid of $N_x \times N_y$ pixels for a set of N_λ wavelength points that sample one or several spectral lines. We propose a model atmosphere for every pixel to explain the observations, each model atmosphere being defined with a set of N_{par} parameters. These parameters can be, for instance, the standard parameters of the Milne-Eddington atmosphere, or the depth stratification of the physical parameters needed to synthesize a line either in local or in non-local thermodynamical equilibrium. Consequently, we assume that our forward model depends on a set of parameters that we encode on the vector $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \dots\}$. Each parameter \mathbf{p}_i encodes the value of a certain parameter (i.e., magnetic field inclination, Doppler velocity, etc.) as a 2D map. The aim of any inversion code is, then, to infer the full vector \mathbf{p} that better reproduces the observed Stokes profiles. As explained in the introduction, classical inversion codes deal with this problem by inverting the observed Stokes profiles pixel by pixel. In so doing, we fully neglect the presence of spatial correlation. This spatial correlation is usually a consequence of a specific physical mechanism in the Sun but we do not require any knowledge of it here whatsoever.

In our approach, instead of working directly on the real space of parameters, we use a linear transformation so that the i -th parameter is given by:

$$\mathbf{q}_i = \mathbf{W}[\mathbf{p}_i]. \quad (2.1)$$

In the previous equation, \mathbf{W} is a linear operator that transforms from the real space to the transformed space. This transformation is usually orthogonal (Fourier, wavelet or discrete cosine transforms) and are associated with a standard basis set. However, they can also be non-orthogonal, and they are associated with a *dictionary*. It is clear that by working on the transformed space and using an inversion code to infer the transformed parameters we have gained nothing. However, here we can impose a fundamental key ingredient: *the assumption that the transformed image is sparse in the transformed domain*. In other words, if the appropriate transformation \mathbf{W} is used, many elements of the transformed image (that we term modes in the following for simplicity) are zero or very close to zero in absolute value. Sparsity, that is usually fulfilled in nature, has been shown to be an extremely powerful assumption. Techniques like compressed sensing (Candès, Romberg & Tao 2006, Donoho 2006) or exact matrix completion from partial measurements (Candès & Recht 2009) strongly rely on the sparsity assumption. Note that, although orthogonal transformations are easier to deal with because they usually have a fast transform, it has become clear in the last years that non-orthogonal (or overcomplete) transformations greatly enhance the sparsity level. This is not difficult to understand in the simple case of an overcomplete transformation (or dictionary) made of the union of several orthogonal basis. Parts of the signal not sparse in one of the orthogonal basis, can be sparse in another of the basis used in the dictionary.

As a motivating example, Fig. 1 shows the compressibility of model parameters using the Daubechies-1 (db1 or Haar) orthonormal wavelet (Jensen & la Cour-Harbo 2001), specially suited for discontinuous signals, the Daubechies-8 (db8) orthonormal wavelet, that is a wavelet appropriate for representing efficiently maps with smooth maps and the DCT, also appropriate for smooth and periodic signals. Each panel shows the reconstruction of the original image (the rightmost image in each panel) when the image is

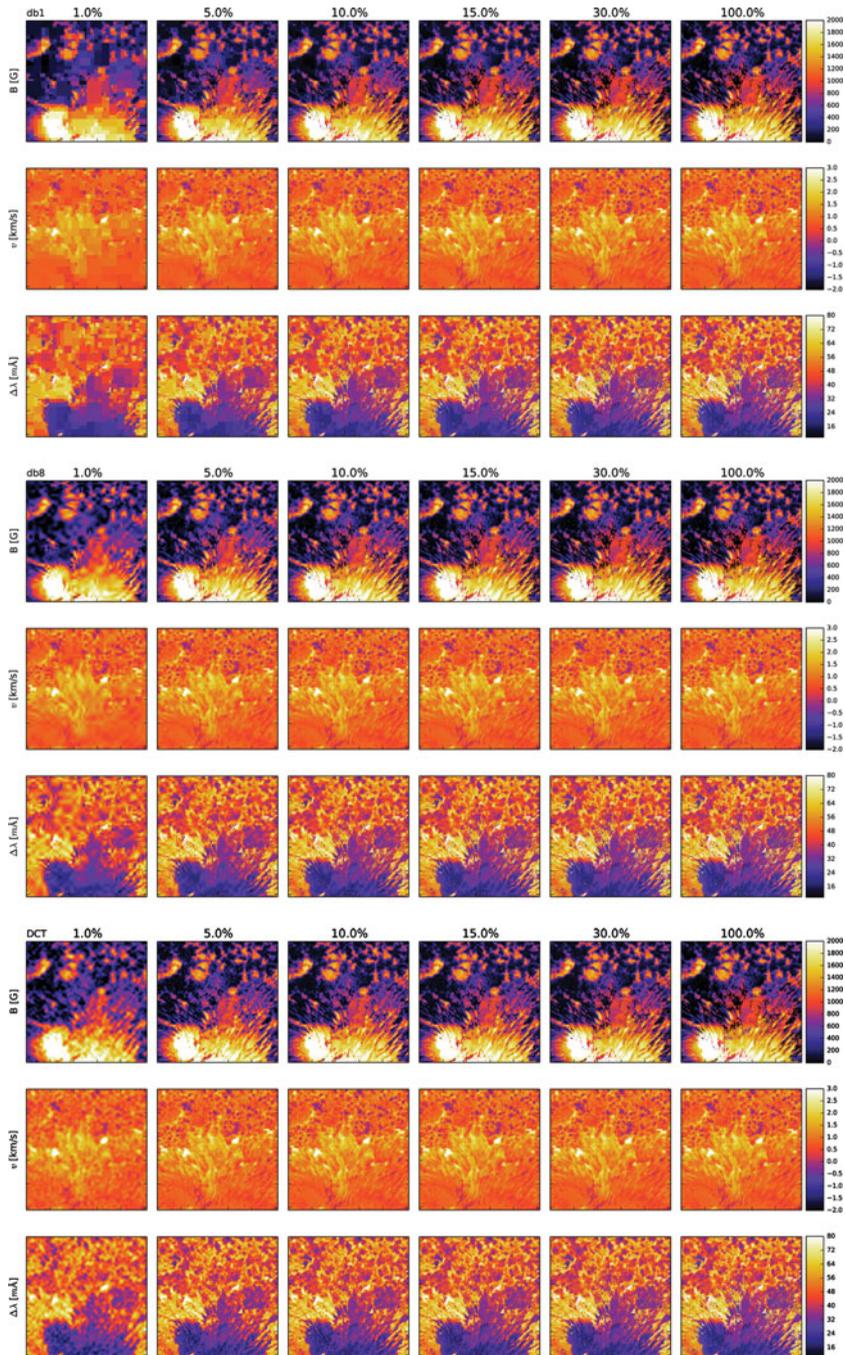


Figure 1. Testing for sparsity using the Daubechies-1 (upper panels), Daubechies-8 (middle panels) and discrete cosine (lower panels) transforms for a sunspot map. The rows display maps of the parameters of the Milne-Eddington model. Each column indicates the fraction of non-zero coefficients of the original map that are retained. The last column is the original map.

transformed, thresholded leaving only a certain percentage of the largest coefficients and transformed back. The percentage of non-zero coefficients is shown as a label in each column. The figure demonstrates that one gets an image that is essentially indistinguishable

from the original one when the number of non-zero coefficients is as low as 15%. This compression can be obtained because the linear transformations capture part of the spatial correlation in the image. As a consequence, the value of ~85% of the pixels can be predicted from the value of just 15% of the image plus the presence of spatial correlation.

An inversion code based on the previous idea of applying a sparsity regularization leads to the following obvious advantages:

- Imposing that the solution to the inversion has to be sparse introduces a strong regularization because the number of unknowns is heavily reduced. From the potential $N_{\text{par}}N_xN_y$ to only $sN_{\text{par}}N_xN_y$, with $s \ll 1$. We have found that typical values of the sparsity s will be around 20-30%. Given that a large fraction of the coefficients can be set to zero without degrading the solution, the number of free variables that we are currently using in pixel-by-pixel inversions is highly overestimated. An overly large number of degrees of freedom can produce nonphysical fluctuations of the physical parameters.
- The inverse linear transformation gives the value of a physical parameters at a specific observed pixel as a linear combination of all modes in the map. This global character induces changes in all the pixels of the original space simultaneously when perturbing the value of a single mode. Reversing the argument, the observed Stokes profiles of a single pixel provide a little amount of information to all modes simultaneously. This large redundancy is a very interesting advantage of our approach and introduces a strong regularization of the solution. This global character also introduces strong regularization due to the presence of spatial correlation in the observables and the physical parameters.

3. Numerical methods

All standard inversion codes work by optimizing a merit function that measures the ℓ_2 -norm of the residuals with respect to the vector \mathbf{p} of physical parameters. If the observed Stokes parameters are corrupted with Gaussian noise with diagonal covariance matrix, the merit function is:

$$\chi_{\mathbf{p}}^2 = \frac{1}{4N_{\lambda}N_{\text{pix}}} \sum_{k=1}^{N_{\text{pix}}} \sum_{i=1}^4 \sum_{j=1}^{N_{\lambda}} w_i \frac{[S_i(\lambda_j, \hat{\mathbf{p}}_k) - O_i(\lambda_j, k)]^2}{\sigma_{ijk}^2} \tag{3.1}$$

where $\mathbf{S}(\lambda_j, \hat{\mathbf{p}}_k) = (I(\lambda_j, \hat{\mathbf{p}}_k), Q(\lambda_j, \hat{\mathbf{p}}_k), U(\lambda_j, \hat{\mathbf{p}}_k), V(\lambda_j, \hat{\mathbf{p}}_k))$ refers to the synthetic Stokes vector at wavelength position j and position k , while $\mathbf{O}(\lambda_j, k)$ is the equivalent observed Stokes vector at this very same wavelength and pixel. The symbol σ_{ijk} stands for the standard deviation of the noise at wavelength position i , for the Stokes parameter j and pixel k , and w_i is the weight associated to each Stokes parameter. Note that $\hat{\mathbf{p}}_k$ are the model parameters for pixel k seen as a vector. This weight is introduced for technical reasons to help improve the convergence during the optimization. Classically, it is customary to solve the problem

$$\text{argmin}_{\mathbf{p}} \chi_{\mathbf{p}}^2, \tag{3.2}$$

where the operator $\text{argmin}_{\mathbf{p}}$ returns the value of \mathbf{p} that minimizes the function to which the operator is applied. Problem (3.2) is usually solved by direct application of the Levenberg-Marquardt algorithm, which is specially suited to the optimization of such ℓ_2 -norms.

When working on the transformed domain, we substitute the merit function of Eq. (3.1) by:

$$\chi_{\mathbf{q}}^2 = \frac{1}{4N_{\lambda}N_{\text{pix}}} \sum_{k=1}^{N_{\text{pix}}} \sum_{i=1}^4 \sum_{j=1}^{N_{\lambda}} w_i \frac{[S_i(\lambda_j, [\mathbf{W}^{-1}[\mathbf{q}]]_k) - O_i(\lambda_j, k)]^2}{\sigma_{ijk}^2}. \quad (3.3)$$

Additionally, if we impose the sparsity constraint, we want to solve the following problem:

$$\operatorname{argmin}_{\mathbf{q}} \chi_{\mathbf{q}}^2, \text{ subject to } \|\mathbf{q}\|_0 \leq s \quad (3.4)$$

where $\|\mathbf{q}\|_0$ is the ℓ_0 -norm of \mathbf{q} , equivalent to counting the number of non-zero elements. The desired sparsity level is set by the upper limit $s \ll N_{\text{par}}N_xN_y$. In other words, we minimize the merit function χ^2 with respect to the modes \mathbf{q} but using only a very small number of non-zero elements in \mathbf{q} . The problem can be written equivalently in lagrangian form as:

$$\operatorname{argmin}_{\mathbf{q}} (\chi_{\mathbf{q}}^2 + \lambda\|\mathbf{q}\|_0). \quad (3.5)$$

This is the form of the optimization that is commonly known as *synthesis prior*. The idea is that we work directly with the transformed coefficients and impose the sparsity on them. However, it has been realized recently that another dual view of the problem appears if we optimize the following *analysis prior* problem:

$$\operatorname{argmin}_{\mathbf{p}} (\chi_{\mathbf{p}}^2 + \lambda\|\mathbf{W}\mathbf{p}\|_0). \quad (3.6)$$

It is clear that both schemes are equivalent when \mathbf{W} is an orthogonal transformation, but it remains to check what are the differences when the transformation is non-orthogonal. It is obvious that the analysis prior problem is much more convenient for overcomplete dictionaries because the number of unknowns is smaller. Our computer code uses the two approaches although we focus now on the synthesis prior.

As a consequence of the recent huge increase on the interest of compressed sensing, big data and related techniques based on the idea of sparsity, several algorithms have been developed to optimize convex functions with non-convex constraints (ℓ_0 or ℓ_1 norms). One of the most successful ones belong to a class of methods termed proximal algorithms Parikh & Boyd (2014), which can be viewed as the equivalent of the Newton method for non-smooth, constrained, and large-scale optimization problems. All these methods try to solve the following problem:

$$\operatorname{argmin}_{\mathbf{q}} f(\mathbf{q}) + g(\mathbf{q}), \quad (3.7)$$

where $f(\mathbf{q})$ is a convex and differentiable function, while $g(\mathbf{q})$ is another convex function, possible non-smooth. The solution is given by the following iteration:

$$\mathbf{q}_{i+1} = \operatorname{prox}_g [G(\mathbf{q}_i)], \quad (3.8)$$

where $G(\mathbf{q})$ is an operator that carries out a step along the gradient of $f(\mathbf{q})$. Given that $f(\mathbf{q})$ is differentiable, it is possible to use all the machinery used in all the inversion codes discussed in the introduction.

The key ingredient of these algorithms is the application of the *proximity operator* of the regularization function $h(\mathbf{x})$, defined as

$$\operatorname{prox}_h(\mathbf{x}) = \operatorname{argmin}_{\mathbf{v}} \left[\frac{1}{2} \|\mathbf{v} - \mathbf{x}\|_2^2 + \lambda h(\mathbf{v}) \right]. \quad (3.9)$$

Examples of this operator for different regularization functions can be found in Parikh & Boyd (2014). For our case of using the ℓ_0 -norm as regularization, the proximity operator

is simply given by:

$$\text{prox}_{\ell_0}(\mathbf{x}) = H_s(\mathbf{x}), \quad (3.10)$$

where $H_s(\mathbf{x})$ is the nonlinear hard thresholding operator. This operator keeps the s elements of \mathbf{x} with largest absolute value untouched and sets to zero the remaining elements. Although all the results presented in this paper use the ℓ_0 -norm as regularization, sometimes it is interesting to use the ℓ_1 -norm, which is given by $\|\mathbf{x}\|_1 = \sum_i |x_i|$. In this case, the proximal operator is the smooth thresholding operator, which is given by:

$$\text{prox}_{\ell_1}(\mathbf{x}) = \text{sign}(\mathbf{x})(|\mathbf{x}| - \lambda)_+, \quad (3.11)$$

where $(\cdot)_+$ denotes the positive part.

Concerning the operator $G(\mathbf{q})$, it is desirable to use one that produces a very fast convergence to the solution. Second order iterative algorithms (for instance the Levenberg-Marquardt algorithm) would be desirable. However, even with the imposed compression, the number of unknowns in the problem we want to solve is still extremely large. Therefore, the use of second order algorithms like LM is computationally prohibitive. The main reason is that the Hessian matrix that is used in these algorithms turns out to be immensely large and constructing and inverting it becomes unfeasible on most computers. Despite these complications, it would still be possible to use these methods when the Hessian is sparse. This is the case of the strategy followed by van Noort (2012), who finds a sparse diagonally-dominant Hessian, a direct consequence of the local character of the PSF. Our case is much less favorable because the global character of the linear transformation \mathbf{W} produces a dense Hessian matrix and no shortcut can be used to build, store or invert it.

The only current way to avoid this obstacle, that is widespread in big data applications, is the application of first order algorithms, which are in essence equivalent to the gradient descent method. Using Eq. (3.8), the simplest proximal algorithm using the gradient descent method to solve Eq. (3.4) is given by (Parikh & Boyd 2014):

$$\mathbf{q}_{k+1} = H_s [\mathbf{q}_k - \mathbf{h} \nabla_{\mathbf{q}} \chi^2(\mathbf{q}_k)]. \quad (3.12)$$

This iteration is just a plain gradient descent algorithm (which only makes use of first order derivatives) which is augmented by using a hard thresholding projection operator in each iteration. In other words, after moving the solution on the direction of the negative gradient (controlled by the step-size \mathbf{h}), one sets to zero all elements that do not fulfill the sparsity constraint. This simple iterative scheme displays a sublinear convergence rate $O(1/k)$. Recently, a trivial improvement of the gradient descent method known as Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) has been developed by Beck & Teboulle (2009) showing a quadratic convergence rate $O(1/k^2)$. We utilize a combination of FISTA and the restarting scheme developed by O'Donoghue & Candès (2013).

Eq. (3.12) shows that the optimization method relies on the computation of the gradient of the merit function with respect to the modes, that we have made explicit by using the subindex \mathbf{q} on $\nabla_{\mathbf{q}} \chi^2$. Given the linear character of the transformation of Eq. (2.1), this gradient can be trivially related to the gradient with respect to the physical parameters (the response functions), that can be analytically written as:

$$\nabla_{\mathbf{q}} \chi^2 = \mathbf{W}^{-1} \nabla_{\mathbf{p}} \chi^2 \quad (3.13)$$

Even though gradient descent methods are known to be fast when approaching the minimum, they become slower in the refinement phase of the solution (that is precisely the reason why the Levenberg-Marquardt method is a combination of a gradient descent method and a Newton method, controlled by the Hessian). However, the non-convex

optimization of large-scale problems has to rely on first-order derivatives because the calculation and storage of the Hessian is impractical. In our experience and that of many others in the literature, the FISTA algorithm is a competitive optimization technique. To achieve a good convergence speed, the vector of step-sizes \mathbf{h} has to be tweaked. Our experiments have shown that these steps can be kept fixed and used for inverting completely different data sets without any special impact on the convergence speed. However, we are currently investigating the possibility of improving the convergence speed along two lines. First, using approximations to the diagonal Hessian matrix that can be efficiently computed using quasi-Newton methods. Quasi-Newton methods update the Hessian matrix by analyzing successive gradient vectors using a generalization of the secant method. We anticipate that the methods developed by Becker & Fadili (2012) or Marjugi & Leong (2013) can be of help. Second, using conjugate gradient methods.

Acknowledgements

Financial support by the Spanish Ministry of Economy and Competitiveness through projects AYA2010–18029 (Solar Magnetism and Astrophysical Spectropolarimetry) and Consolider-Ingenio 2010 CSD2009-00038 are gratefully acknowledged. AAR also acknowledges financial support through the Ramón y Cajal fellowships. JdlCR acknowledges financial support from the CHROMOBS project funded by the Knut and Alice Wallenberg Foundation. Part of the computations included in this paper were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the National Supercomputer Centre (Linköping University), with project id *snic2014-1-273*. This research has made use of NASA's Astrophysics Data System Bibliographic Services.

References

- Asensio Ramos, A., Martínez González, M. J., & Rubiño Martín, J. A. 2007, *A&A* 476, 959
- Auer, L. H., House, L. L., & Heasley, J. N. 1977, *Sol. Phys.* 55, 47
- Beck, A. & Teboulle, M. 2009, *SIAM Journal on Img. Sci.* 2, 183
- Becker, S. & Fadili, M. J. 2012, *eprint arXiv:1206.1156*
- Borrero, J. M., Tomczyk, S., Kubo, M., *et al.* 2010, *Sol. Phys.* 35
- Candès, E., Romberg, J., & Tao, T. 2006, *Comm. Pure Appl. Math.* 59, 1207
- Candès, E. J. & Recht, B. 2009, *Found. Comput. Math.* 9, 717
- de la Cruz Rodríguez, J., Socas-Navarro, H., Carlsson, M., & Leenaarts, J. 2012, *A&A* 543, A34
- Donoho, D. 2006, *IEEE Trans. on Information Theory* 52, 1289
- Frutiger, C., Solanki, S. K., Fligge, M., & Bruls, J. H. M. J. 2000, *A&A* 358, 1109
- Harvey, J., Livingston, W., & Slaughter, C. 1972, in: Manuscripts presented at a conference held in Boulder, Colorado, 20 August - 2 September, 1971, Boulder, CO: National Center for Atmospheric Research, 227
- Jensen, A. & la Cour-Harbo, A. 2001, *Ripples in Mathematics: The Discrete Wavelet Transform* (Springer Verlag)
- Landi Degl'Innocenti, E. & Landi Degl'Innocenti, M. 1977, *A&A* 56, 111
- Landi Degl'Innocenti, E. & Landolfi, M. 2004, *Polarization in Spectral Lines* (Kluwer Academic Publishers)
- Lites, B., Casini, R., Garcia, J., & Socas-Navarro, H. 2007, *Mem. Soc. Astron. Ital.* 78, 148
- Marjugi, S. M. & Leong, W. J. 2013, *J. Appl. Math.* 2013, 523476
- O'Donoghue, B. & Candès, E. 2013, *Foundations of Computational Mathematics* (3 July 2013, online)
- Orozco Suárez, D., Bellot Rubio, L. R., del Toro Iniesta, J. C., *et al.* 2007, *ApJL* 670, L61
- Parikh, N. & Boyd, S. 2014, *Foundations and Trends in Optimization*, 1
- Rees, D. E., López Ariste, A., Thatcher, J., & Semel, M. 2000, *A&A* 355, 759
- Ruiz Cobo, B. & Asensio Ramos, A. 2013, *A&A* 549, L4

- Ruiz Cobo, B. & del Toro Iniesta, J. C. 1992, *ApJ* 398, 375
- Skumanich, A. & Lites, B. W. 1987, *ApJ* 322, 473
- Socas-Navarro, H., de la Cruz Rodriguez, J., Asensio Ramos, A., Trujillo Bueno, J., & Ruiz Cobo, B. 2014, *eprint arXiv:1408.6101*
- Socas-Navarro, H., Trujillo Bueno, J., & Ruiz Cobo, B. 2000, *ApJ* 530, 977
- van Noort, M. 2012, *A&A*, 548, A5