# A COMPARISON OF THE COMPUTATIONAL PERFORMANCE OF ITERATIVELY REWEIGHTED LEAST SQUARES AND ALTERNATING MINIMIZATION ALGORITHMS FOR $\ell_1$ INVERSE PROBLEMS

*Paul Rodríguez*

Electrical Department
Pontificia Universidad Católica del Perú
Lima, Peru

*Brendt Wohlberg*[*]

Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545, USA

**ABSTRACT**

Alternating minimization algorithms with a shrinkage step, derived within the Split Bregman (SB) or Alternating Direction Method of Multipliers (ADMM) frameworks, have become very popular for $\ell^1$-regularized problems, including Total Variation and Basis Pursuit Denoising. It appears to be generally assumed that they deliver much better computational performance than older methods such as Iteratively Reweighted Least Squares (IRLS). We show, however, that IRLS type methods are computationally competitive with SB/ADMM methods for a variety of problems, and in some cases outperform them.

***Index Terms*—** Inverse Problems, Total Variation, Split-Bregman, Iteratively Reweighted Least Squares

## 1. INTRODUCTION

The solution of $\ell^1$-regularized problems such as Total Variation (TV) [1] and Basis Pursuit Denoising (BPDN) [2] are of broad interest because they are used to solve a variety of important problems, including image restoration (image denoising, deconvolution, inpainting, etc.), compressed sensing [3], and classification problems. In general, $\ell^1$-regularized problems have the form

$$\min_{\mathbf{u}} T(\mathbf{u}) \qquad T(\mathbf{u}) = F(\mathbf{u}) + \lambda R(\mathbf{u}) \tag{1}$$

where $F(\cdot)$ is the data fidelity term (which depends on the noise model), $R(\cdot) = \|\mathbf{r}(\cdot)\|_1$ is the regularization term, and the scalar $\lambda$ is the regularization parameter. For instance, under a Gaussian noise model, typically $F(\mathbf{u}) = \frac{1}{2}\|A\mathbf{u} - \mathbf{b}\|_2^2$ where the interpretation of the linear operator $A$ and observed data $\mathbf{b}$ change depending on whether we are solving a TV problem, for which $\mathbf{r}(\mathbf{u}) = \nabla \mathbf{u}$, or a sparse representation problem, for which $\mathbf{r}(\mathbf{u}) = \mathbf{u}$.

While both terms $F(\cdot)$ and $R(\cdot)$ are convex, the development of computationally efficient numerical algorithms for problem (1) has attracted considerable interest over the past decade since it has proved to be a difficult problem. In this paper we focus on numerical algorithms for solving (1) that are based on the Iteratively Reweighted Least Squares (IRLS) [4, 5] method and on the Split Bregman (SB) [6] or Alternating Direction Method of Multipliers (ADMM) [7, 8] methods.

IRLS based methods have been successfully used to solve problems with the form of (1); in particular, we consider the Iteratively Reweighted Norm (IRN) algorithm [9] for the TV case and the Focal Underdetermined System Solver (FOCUSS) algorithm [10, 11] for the BPDN case. Besides the good computational performance and convergence proofs reported in [9, 10, 11], a detailed analysis of IRLS based methods for sparse representations is presented in [12]. Similarly, the equivalent [13] SB and ADMM methods can solve a wide variety of problems summarized by (1). In particular, in the seminal paper [6], the elegance and simplicity of the SB method, as well as its computational properties are described and applied to solve the TV and BPDN problems.

It appears that the SB/ADMM method is widely considered to be the current leader in terms of computational performance for $\ell^1$-regularized problems. We show, however, that IRLS based methods are computationally competitive with SB/ADMM methods for the TV and BPDN problems, outperforming them in several cases. The outline of the paper is as follows: we summarize the IRLS and SB/ADMM algorithms in Sections 2 and 3 respectively, report and analyze their computational performance in Section 4, and conclude in Section 5.

## 2. IRLS BASED METHODS

Since its introduction [4] the IRLS method has been applied to a variety of optimization problems. Originally, the IRLS method was used to solve the $\ell^p$ minimization problem $T(\mathbf{u}) = \frac{1}{p}\|A\mathbf{u} - \mathbf{b}\|_p^p$ by iteratively approximating it by a weighted $\ell^2$ norm. At iteration $k$ the solution $\mathbf{u}^{(k+1)}$ is the minimizer of $\frac{1}{2}\|W^{(k)^{1/2}}(A\mathbf{u} - \mathbf{b})\|_2^2$, with weighting

[*]This research was supported by the NNSA's Laboratory Directed Research and Development Program.

matrix $W^{(k)} = \text{diag}\left(|A\mathbf{u}^{(k)} - \mathbf{b}|^{p-2}\right)$. For $2 \leq p \leq 3$ this algorithm converges to the global minimizer [14], while for $1 \leq p < 2$ the definition of the weighting matrix $W^{(k)}$ must be modified to avoid numerical instability due to division by zero or by a very small value. A standard approach is to threshold elements of $A\mathbf{u}^{(k)} - \mathbf{b}$ in constructing the corresponding elements of $W^{(k)}$, but other choices are also possible [5, 12]. For $0 < p < 1$ it has been shown [12], within a sparse representation framework, that the IRLS algorithm not only converges, but increases its convergence rate as $p$ goes to zero.

### 2.1. IRLS for TV restoration

Without loss of generality, we will focus on the $\ell^2$-TV [1] case: $T(\mathbf{u}) = \frac{1}{2}\|A\mathbf{u} - \mathbf{b}\|_2^2 + \lambda R(\mathbf{u})$, where $R(\mathbf{u}) = \|\sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2}\|_1$ is the discrete version of $\|\nabla\mathbf{u}\|_1$, with $D_x$ and $D_y$ representing the horizontal and vertical discrete derivative operators respectively, $A$ is the forward operator, $\mathbf{b}$ is the observed noisy data, $\lambda$ is a weighting factor controlling the relative importance of the data fidelity and regularization terms, and $\mathbf{u}$ is the restored image data.

The key idea [9] is to express the regularization term by the quadratic approximation $Q_R^{(k)}(\mathbf{u}) = \frac{1}{2}\|{W_R^{(k)}}^{1/2} D\mathbf{u}\|_2^2$, where $W_R^{(k)} = I_2 \otimes \Omega_R^{(k)}$, $D = [D_x^T D_y^T]^T$, $\Omega_R^{(k)} = \text{diag}\left(\left((D_x\mathbf{u}^{(k)})^2 + (D_y\mathbf{u}^{(k)})^2\right)^{-0.5}\right)$, $I_N$ is a $N \times N$ identity matrix and $\otimes$ is the Kronecker product. The resulting iterations can be expressed in the form of the standard IRLS problem:

$$T^{(k)}(\mathbf{u}) = \frac{1}{2}\left\|\begin{bmatrix} 1 & 0 \\ 0 & W_R^{(k)} \end{bmatrix}^{1/2}\begin{bmatrix} A \\ \sqrt{\lambda}D \end{bmatrix}\mathbf{u} - \begin{bmatrix} b \\ 0 \end{bmatrix}\right\|_2^2 \quad (2)$$

For a given current solution $\mathbf{u}^{(k)}$, the weighting matrix $W_R^{(k)}$ can be easily computed, and the threshold $\tau$ may be automatically adapted to the input image to avoid numerical instability [9, Sec. IV.G]. Finally, the resulting IRN algorithm has to iteratively solve the linear system

$$\left(A^T A + \lambda D^T W_R^{(k)} D\right)\mathbf{u}^{(k+1)} = A^T\mathbf{b}, \quad (3)$$

which is its most computationally demanding part. The same strategy can be used to solve the $\ell^1$-TV (also described in [9]) as well other noise models within the TV framework, including the vector-valued (color) TV.

### 2.2. IRLS for sparse representation

The BPDN problem, originally introduced in [2], seeks to decompose an input signal via a linear combination of atoms from an overcomplete dictionary, where the coefficients of the linear combination are optimized according the sparsity criterion given by $T(\mathbf{u}) = \frac{1}{2}\|\Phi\mathbf{u} - \mathbf{b}\|_2^2 + \lambda R(\mathbf{u})$, where $R(\mathbf{u}) = \|\mathbf{u}\|_1$ is the sparsity term, $\mathbf{b}$ is the observed signal, $\Phi$ is the dictionary matrix, $\lambda$ is a weighting factor controlling the relative importance of the data fidelity and sparsity terms, and $\mathbf{u}$ is the sparse representation.

In [10, 11] the sparsity term is represented by the quadratic approximation $Q_S^{(k)}(\mathbf{u}) = \frac{1}{2}\|{W_S^{(k)}}^{1/2}\mathbf{u}\|_2^2$ to solve the BPDN problem. As for any IRLS type algorithm, the weighting matrix $W_S^{(k)} = \text{diag}((\mathbf{u}^{(k)})^{-1})$ has to be computed so as to avoid numerical instability. The particular cost functional in the BPDN problem, however, allows the substitution of an auxiliary variable $\mathbf{v}$ defined by $\mathbf{u} = {W_S^{(k)}}^{1/2}\mathbf{v}$, for which the corresponding quadratic problem is $T^{(k)}(\mathbf{v}) = \frac{1}{2}\|\Phi {W_S^{(k)}}^{-1/2}\mathbf{v} - \mathbf{b}\|_2^2 + \frac{\lambda}{2}\|\mathbf{v}\|_2^2$ [10, 11]. The expression involving this $W_S^{(k)}$ raised to a negative power does not generate any numerical instability, and it has been shown computationally [10, 11] that the use of the auxiliary variable helps to reduce the number of conjugate gradient (CG) iterations needed to solve the resulting linear system when compared with the number of CG iterations needed to solve the resulting system that does not use this change of variable.

The most computationally demanding part of the resulting FOCUSS algorithm [10, 11] is to solve the linear system

$$({W_S^{(k)}}^{-1/2}\Phi^T\Phi{W_S^{(k)}}^{-1/2} + \lambda I)\mathbf{v} = {W_S^{(k)}}^{-1/2}\Phi^T\mathbf{b}\,. \quad (4)$$

Additionally, as mentioned in [10, 11], depending on characteristics of the overcomplete dictionary $\Phi$, this sub-problem could be further simplified to solving $\chi = (\Phi{W_S^{(k)}}^{-1}\Phi^T + \lambda I)^{-1}\mathbf{b}$ and then computing $\mathbf{v} = {W_S^{(k)}}^{-1/2}\Phi^T\chi$.

## 3. ALTERNATING MINIMIZATION METHODS

Alternating minimization methods have become popular in the past few years due to their ability to solve $\ell^1$ regularized problems (1) in a simple and computationally efficient fashion. Although there are several incarnations of these methods [13], we focus on the Split-Bregman (SB) [6] algorithm, while noting that it is now recognized that the SB algorithm is equivalent to the older Alternating Direction Method of Multipliers (ADMM) [7, 8] algorithm.

As before, we consider the TV and BPDN problems in particular. The key idea of the SB method [6] is to introduce an auxiliary variable to modify the original cost functional (1) so that it can be iteratively minimized using two steps per iteration: a generalized Tikhonov problem step, followed by a shrinkage step. Moreover, after algebraic manipulation, it can be shown [6] that the iterative minimization of the cost functional

$$T^{(k)}(\mathbf{u},\mathbf{v}) = F(\mathbf{u}) + \lambda\|\mathbf{v}\|_1 + \frac{\mu}{2}\|\mathbf{v} - \mathbf{r}(\mathbf{u}) - \mathbf{b}^{(k)}\|_2^2\,, \quad (5)$$

where $\mathbf{b}^{(k)} = \mathbf{b}^{(k-1)} + (\mathbf{r}(\mathbf{u}^{(k)}) - \mathbf{v}^{(k)})$ is the cumulative error, leads to the solution of (1).

### 3.1. SB for TV restoration

The general SB algorithm can be easily adapted to handle isotropic TV; among other operations, such as shrinkage and auxiliary vector updates (which are not computationally demanding), the SB-TV algorithm has to solve the linear system

$$\left(A^TA - \mu D^TD\right)\mathbf{u}^{(k+1)} = A^T\mathbf{b} + \mu D^TD(\mathbf{v}^{(k)} - \mathbf{b}^{(k)}) \quad (6)$$

Note that the left hand side (LHS) of (6) is constant across different iterations while its right hand side (RHS) changes at each iteration; the opposite is true for the resulting linear system in (3). Furthermore, for the denoising case ($A = I$) both linear systems (6) and (3) are block tri-diagonal, but since the LHS of (6) is constant across iterations (and strictly diagonally dominant), it seems natural (as suggested in [6]) to use the Gauss-Seidel method to solve it; this is not the case for (3), which is most efficiently solved via the CG algorithm, typically needing about 4 CG iterations.

One key computational difference between the SB-TV and the IRN-TV (IRLS based) algorithms is that even though the number of floating-point operations for each SB-TV iteration is slightly smaller than that of each IRN-TV iteration, typically the number of global iterations to attain good reconstruction results for the latter algorithm are typically less than 5 [9] whereas the former are typically greater than 10 [6]. See Section 4 for numerical simulations that support our claims.

### 3.2. SB for sparse representation

The BPDN problem can be directly solved by the SB algorithm, as implied by its general derivation (see [6, Sec. 3]). As for the TV case, the shrinkage and auxiliary vector updates are not computationally demanding, but the linear system

$$\left(\lambda\Phi^T\Phi + \mu I\right)\mathbf{u}^{(k+1)} = \mu\mathbf{v}^{(k)} + \lambda\Phi^T(\mathbf{v}^{(k)} - \mathbf{b}^{(k)}) \quad (7)$$

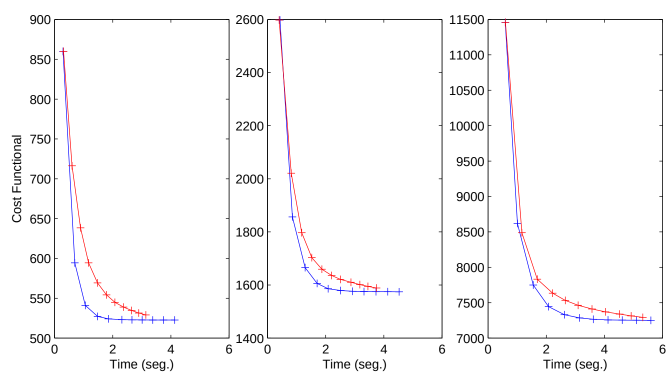
is computationally demanding for some applications. We also note that LHS of (7) is constant across different iterations having a block-diagonal Toeplitz structure for some applications (as the one explored in [6]), while its RHS changes at each iteration. Moreover in (7) the size of the linear system is given by the size of $\Phi^T\Phi$, as well as for (4), although the trick (change of variable in order to solve a linear system of size $\Phi\Phi^T$) can not be efficiently applied for the SB algorithm, since the variable $\mathbf{v}^{(k)}$ would be multiplied by $\Phi^{-T}$.

Finally we mention that, when the dictionary $\Phi$ is available as an explicit matrix (which is usually the case when the dictionary is learned from the input data), it is possible to pre-compute an LU factorization of $\left(\lambda\Phi^T\Phi + \mu I\right)$, allowing a low computational cost solution to the linear system at each iteration.

## 4. EXPERIMENTAL RESULTS

In order to compare the computational performance of the IRLS and SB based algorithms for TV denoising and BPDN we have implemented all the described algorithms in Matlab-only code, as well as in C code (available at [15]) and have used the implementation provided by the authors of [6] (C code only [16]) for the $\ell^2$-TV case. All simulations have been carried out using the above mentioned code on a 1.73GHz Intel core i7 CPU laptop (L2: 6144K, RAM: 6G)

Due to space constraints we only present results for the Matlab-only TV denoising case, but all other simulations can be found at [15]. The observed image $\mathbf{b}$ is corrupted with additive Gaussian noise $\mathbf{b} = \mathbf{u}^* + \sigma \cdot \boldsymbol{\eta}$, for $\sigma = \{0.05, 0.1, 0.25\}$ where $\mathbf{u}^*$ is one of the original test images (Lena, Mandrill and Peppers) and $\boldsymbol{\eta}$ is unit variance Gaussian noise. In Table 1 we provide the original cost functional $T(\mathbf{u}) = \|\mathbf{u} - \mathbf{b}\|_2 + \lambda \cdot \|\sqrt{(D_x\mathbf{u})^2 + (D_y\mathbf{u})^2}\|_1$, SNR, SSIM [17] and elapsed time at each main iteration only (due to space constrains) for $\lambda = 0.045$ (we take $\mu = 2\lambda$ as suggested in [6] for the SB case) and for $\sigma = 0.05$ (which corresponds to Figure 1.a).




(a) (b) (c)

**Fig. 1**. TV denoising cost functional versus time for the IRN-TV (blue) and SB-TV (red) Matlab-only version algorithms (10 main iterations). The input image was grayscale Lena ($512 \times 512$) corrupted with Gaussian noise (a) with $\sigma = 0.05$, denoised with $\lambda = 0.045$, (b) with $\sigma = 0.1$, denoised with $\lambda = 0.065$ and (c) with $\sigma = 0.25$ , denoised with $\lambda = 0.15$.

In Fig. 1 we show a typical evolution of the TV cost functional per main iteration of the IRN-TV and SB-TV algorithms: the first 3-5 iterations of the former have a steeper slope than that of the latter; i.e. the IRN-TV reaches the minimum in fewer iterations than the SB-TV, and although the iterations of the former are more costly than those of the latter, overall the IRN-TV outperforms the SB-TV.

## 5. CONCLUSIONS

Our experimental results show that (in general) each main iteration of the IRN-TV (IRLS based) algorithm takes more time than one main iteration of the SB-TV; but the cost functional reduction and/or the reconstruction quality of the IRN-TV algorithm is more pronounced than the SB-TV algorithm for the first 4 or 5 iterations, giving the former an edge in the overall computational performance.

Although we do not present here computational results for the BP case (they can be found at [15], along with C-code based simulations) the general trend applies: IRLS based methods tend to have a more computationally expensive main iteration than SB methods, but they converge faster to the minimizer (although drastically reducing cost functional reduction after some few iterations); this property makes them computationally competitive with the SB/ADMM methods, outperforming them in some cases.

## 6. REFERENCES


[1] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[2] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[3] Y. Tsaig and D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006.

[4] A. Beaton and J. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, , no. 16, pp. 147–185, 1974.

[5] S. Ruzinsky and E. Olsen, "$L_1$ and $L_\infty$ minimization via a variant of Karmarkar's algorithm," *IEEE Trans. ASSP*, vol. 37, pp. 245–253, Feb. 1989.

[6] T. Goldstein and S. Osher, "The Split Bregman method for L1-regularized problems," *SIAM J. Img. Sci.*, vol. 2, pp. 323–343, April 2009.

[7] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Applic.*, vol. 2, no. 1, pp. 1416–1438, 1976.

[8] R. Glowinski and A. Marrocco, "Sur l'approximation par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non lináires," *Revue Fr. Autom., Inform., Rech. Opér. Analyse numérique*, vol. 9, no. 2, pp. 41–72, 1975.

[9] P. Rodríguez and B. Wohlberg, "Efficient minimization method for a generalized total variation functional," *IEEE Trans. Image Proc.*, vol. 18, no. 2, pp. 322–332, Feb. 2009.

[10] F. Gorodnitsky and D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Trans. Signal Proc.*, vol. 45, no. 3, pp. 600–616, 1997.

[11] B. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Proc.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.

[12] I. Daubechies, R. DeVore, M. Fornasier, and C. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Comm. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, 2010.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Fdns. & Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[14] R. Byrd and D. Payne, "Convergence of the iteratively reweighted least squares algorithm for robust regression," Tech. Rep. 313, Johns Hopkins University, 1979.

[15] P. Rodríguez and B. Wohlberg, "Numerical methods for inverse problems and adaptive decomposition (NUMIPAD)," `http://numipad.sourceforge.net/`.

[16] T. Goldstein, "The Split-Bregman method," `http://www.stanford.edu/~tagoldst`.

[17] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Perceptual image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Proc.*, vol. 13, no. 4, pp. 600–612, April 2004.


| iter. | IRN-TV | | | | SB-TV | | | |
|---|---|---|---|---|---|---|---|---|
| | $T(\mathbf{u})$ | SNR | SSIM | time | $T(\mathbf{u})$ | SNR | SSIM | time |
| -1 | – | 11.66 | 0.52 | – | – | 11.66 | 0.52 | – |
| 0 | 860 | 13.99 | 0.61 | 0.29 | 860 | 13.99 | 0.61 | 0.31 |
| 1 | 594 | 17.35 | 0.76 | 0.69 | 716 | 15.72 | 0.68 | 0.60 |
| 2 | 541 | 18.30 | 0.81 | 1.05 | 638 | 16.84 | 0.73 | 0.90 |
| 3 | 527 | 18.48 | 0.83 | 1.48 | 594 | 17.52 | 0.77 | 1.17 |
| 4 | 524 | 18.49 | 0.84 | 1.85 | 569 | 17.91 | 0.80 | 1.47 |
| 5 | 523 | 18.46 | 0.84* | 2.31 | 554 | 18.13 | 0.81 | 1.78 |
| 6 | 522 | 18.45 | 0.84* | 2.66 | 544 | 18.24 | 0.82 | 2.07 |
| 7 | 522* | 18.44 | 0.84* | 3.00 | 538 | 18.31 | 0.83 | 2.36 |
| 8 | 522* | 18.43 | 0.84* | 3.37 | 534 | 18.34 | 0.83* | 2.65 |
| 9 | 522* | 18.42 | 0.84* | 3.74 | 531 | 18.36 | 0.84 | 2.89 |
| 10 | 522* | 18.42* | 0.84* | 4.12 | 529 | 18.37 | 0.84* | 3.13 |

**Table 1**. Experimental results for (Matlab-only) IRN-TV and SB-TV denoising case. Both algorithms need 3 CG iterations (in average) to solve (3) and (6). (*): variation with respect to the previous iteration is too small to be noticed.