Abstract—We propose a fast and scalable polyatomic Frank-Wolfe (P-FW) algorithm for the resolution of high-dimensional LASSO regression problems. This algorithm improves upon traditional Frank-Wolfe methods by considering generalized greedy steps with polyatomic (i.e., linear combinations of multiple atoms) update directions, hence allowing for a more efficient exploration of the search space. To preserve sparsity of the intermediate iterates, we re-optimize the LASSO problem over the set of selected atoms at each iteration. For efficiency reasons, the accuracy of this re-optimization step is relatively low for early iterations and gradually increases with the iteration count. We provide convergence guarantees for our algorithm and validate it in simulated compressed sensing setups. Our experiments reveal that P-FW outperforms state-of-the-art methods in terms of runtime, both for FW methods and optimal first-order proximal gradient methods such as the Fast Iterative Soft-Thresholding Algorithm (FISTA).

Index Terms—Conditional Gradient, Frank-Wolfe, LASSO, Sparse Recovery, Convex Optimisation.

I. INTRODUCTION

LINEAR regression is at the core of many inference tasks in signal processing, statistics or machine learning. With the advent of computing power and big data, modern regression problems tend to involve a large number of features, from a few thousands to hundreds of millions depending on the applications. Such large-scale regression problems are typically encountered in computational imaging setups, such as for instance interferometric imaging in radio astronomy [1]–[4] and acoustics [5], environmental monitoring [6], or medical imaging [7]–[9]. In such high dimensional settings, regression problems tend to involve a large number of features, from a few thousands to hundreds of millions depending on the applications. Examples include the Least-Angle Regression (LARS) algorithm [27], [28] as well as specialised Frank-Wolfe (FW) methods1 [15], [29]–[36].

• Coordinate Descent (CD) methods. These methods cyclically select one component of $x$ at a time and update it individually via a line search [37]. A very efficient CD method for the LASSO was proposed in [38].

• Proximal Gradient Descent (PGD) methods. These methods update all components of $x$ at each iteration via successive gradient and proximal steps [39]. Examples include the well-known Fast Iterative Soft-Thresholding Algorithm (FISTA) and its variants [40]–[42].

Thanks to their limited update scope, LARS, CD and FW scale relatively well with the number of features $N$. Indeed, the intermediate iterates they produce are often very sparse, hence alleviating the memory and computation bottlenecks of high dimensional regression. This is especially true for FW methods, which often yield significantly sparser solutions than CD or LARS [15]. These algorithms are however relatively slow to converge with high accuracy, due to suboptimal convergence rates of $O(1/k)$ [31], [43], [44]. This is in contrast with PGD methods which, although less scalable due to their dense intermediate iterates, can achieve first-order optimal convergence rates of $O(1/k^2)$ when used in conjunction with Nesterov’s acceleration schemes [40], [42], [45].

1Frank-Wolfe methods are also referred to as Conditional Gradient (CG) methods in some communities.
In this work, we accelerate the FW method to make it competitive with optimal first-order proximal gradient descent methods such as FISTA. To this end, we propose generalized greedy steps, which we refer to as polyatomic updates. Unlike standard greedy steps, the latter can activate multiple variables at the same time, allowing for a more efficient exploration of the search space. To ensure sparse updates at each iteration. To further accelerate the algorithm, we also consider a partially-corrective step, which re-optimizes the LASSO problem over the set of previously activated atoms. This procedure however requires solving at each iteration an optimization problem that may turn out to be more expensive.

The rest of this article is organized as follows. In Sections II and III, we review the classical Frank-Wolfe algorithm and some of its variants, first in a generic setting and then more specifically for the LASSO problem. In Section IV, we derive the FW algorithm and establish its convergence. In Section V, we test the performance of FW on simulated data.

II. FRANK-WOLFE AND ITS VARIANTS

A. Vanilla Frank-Wolfe

The Frank-Wolfe algorithm, as presented in 1956 [46], aims at minimizing a general constrained convex optimization problem of the form

\[ \arg \min_{x \in D} f(x) \]

where \( f \) is a convex and continuously differentiable cost function and the domain \( D \) is a compact convex subset of a some Banach space. The vanilla version of the procedure is provided in Algorithm 1 below.

Algorithm 1: Vanilla Frank-Wolfe Algorithm (V-FW)

1. Initialize \( x_0 \in D \)
2. for \( k = 1, 2 \cdots \) do
3. \hspace{1em} Find an update direction:
4. \hspace{2em} \( s_k \in \arg \min_{s \in D} \langle \nabla f(x_k), s \rangle \)
5. \hspace{1em} Step size: \( \gamma_k \gets 2/(k+2) \)
6. \hspace{1em} Reweight: \( x_{k+1} \gets (1 - \gamma_k)x_k + \gamma ks_k \)
7. end

The update direction search in step 1) consists in minimizing a linear function over a convex set, which leads to solutions lying on the set’s boundary. When the domain \( D \) is a polytope, the minimum is necessarily reached for one of its extreme points or atoms, hence allowing to restrict step 1) to the generating atoms of \( D \) only. The computation of the update direction is very cheap since it is projection-free. This is in contrast with Projected Gradient Descent methods for (2), which could involve very expensive projections onto the set \( D \) at each iteration.

B. Known Frank-Wolfe variants

Several variants of the FW algorithm have been studied in the literature. Adopting the same formalism as in [43], we review some of these variants that we will later rely on.

1) Exact Line Search: It is possible to replace step 2.a) by an exact line search, leading to Algorithm 2.

Algorithm 2: Exact Line Search

... Same as Algorithm 1, replacing steps 2.a) by:

2.a) Solve: \( \gamma_k \gets \arg \min_{\gamma \in [0,1]} f((1 - \gamma)x_k + \gamma s_k) \)

This step is typically performed when a closed-form solution for step 2.a) exists, e.g. with quadratic objective functionals.

2) Approximate Linear Subproblem: In cases where the gradient minimization step 1) is expensive (e.g. complex domain shape), the latter can be performed approximately. This yields Algorithm 3, where \( \delta > 0 \) controls the approximation quality.

Algorithm 3: Approximate Linear Subproblem of quality \( \delta > 0 \)

... Same as Algorithm 1 or 2, replacing step 1) by:

1) Find: \( s_k \in D \) such that

\[ \langle \nabla f(x_k), s_k \rangle \geq \min_{s \in D} \langle \nabla f(x_k), s \rangle + \gamma_k \delta \]

with \( \gamma_k = 2/(k+2) \)

3) Fully-Corrective FW: Another commonly used variant is known as Fully-Corrective FW (FC-FW), which re-optimizes \( f \) over the convex-hull of all previously selected atoms at each iteration. This results in Algorithm 4 below, where \( \text{Conv} \) denotes the convex hull of a given set.

Algorithm 4: Fully-Corrective Variant (FC-FW)

... Same as Algorithm 1 or 3, replacing steps 2.a)/2.b) by:

2) Solve: \( x_{k+1} \gets \arg \min_{x \in \text{Conv}(s_1, \ldots, s_k)} f(x) \)

The re-optimization step 2) allows for more progress to be made at each iteration and can potentially discard wrongly selected atoms. This procedure however requires solving at each iteration an optimization problem that may turn out to be as difficult as the original one, and thus computationally expensive.

C. Convergence guarantees

Theorem 1 of [43] shows that the sequence of iterates \( x_k \) produced by FW or its variants achieves a convergence rate of \( O(1/k) \) in terms of the objective functional. More specifically, let \( C_f > 0 \) denote the curvature constant\(^2\) of \( f \), and \( f^* \) its optimal value, the iterates of Algorithms 1 to 4 satisfy

\[ f(x_k) - f^* \leq \frac{2}{k+2}(C_f + 2\delta), \quad \forall k \geq 1, \tag{3} \]

with \( \delta = 0 \) for Algorithms 1, 2, and potentially 4 if the latter does not perform the optimal approximate gradient minimization step.

\(^2\) See [43] for a definition.
III. FRANK-WOLFE FOR THE LASSO

While not of the form (2) due to its non-differentiable cost functional, the penalized LASSO regression problem (1) can still be brought into the scope of the FW algorithm by means of an epigraphical lift, as proposed in [29], [31]. To this end, we introduce the bounded norm cone

$$C = \{(t, x) \in \mathbb{R}_+ \times \mathbb{R}^N : \|x\|_1 \leq t \leq M\}$$

with \(M = \|y\|_2^2 / 2\lambda\). With similar arguments as in [31, Lemma 4], it is then possible to show that (1) is equivalent to the constrained optimization problem:

$$\arg \min_{(t, x) \in C} \|y - Ax\|_2^2 + \lambda t. \quad (4)$$

Observe that (4) is indeed of the form (2), since the objective function \(f(t, x) = \|y - Ax\|_2^2 + \lambda t\) is continuously differentiable and \(C \subset \mathbb{R}_+ \times \mathbb{R}^N\) is compact and convex. Applied to (4), the update direction step 1) in Algorithm 1 becomes

$$(t_k, s_k) \in \arg \min_{(t, x) \in C} \langle \eta_k, x \rangle + \lambda t \quad (5)$$

where \(\eta_k = \frac{1}{\lambda} A^T(y - Ax_k)\) is the so-called empirical dual certificate at iteration \(k\) [31]. Since the extreme points of \(C\) are easily shown to be of the form \((M, \pm Me_i)\) (with \(e_i\) the \(j\)-th canonical basis vector), solving (5) is equivalent to finding the index \(i_k\) such that

$$i_k = \arg \max_{i \in \{1, \ldots, N\}} |\eta_k(i)|. \quad (6)$$

Note that the lift variable \(t\) is constant across iterations and is thus omitted in what follows. To sum up, the update direction step 1) amounts to the creation of a new atom \(\pm Me_{i_k}\) that will be reweighted and added to the current iterate \(x_k\). The selected index \(i_k\) is associated to the column of \(A\) which is most correlated with the residuals \(y - Ax\). Note that the exact same update step appears in the greedy Matching Pursuit (MP) [47] and Orthogonal Matching Pursuit (OMP) [48], [49] algorithms, commonly used in signal processing for sparse recovery.

When considering the approximated subproblem variant, step 1') of Algorithm 3 becomes

Find \(i_k \in \{1, \ldots, N\}\), such that

$$|\eta_k(i_k)| \geq \max_{i \in \{1, \ldots, N\}} |\eta_k(i)| - \gamma_k \delta = \|\eta_k\|_{\infty} - \gamma_k \delta. \quad (7)$$

Moreover, the fully-corrective step 2') of Algorithm 4 becomes:

$$\bar{x}_{k+1} \leftarrow \arg \min_{\bar{x} \in \mathbb{R}^{\text{Card}(S_k)}} \|y - A_{S_k} \bar{x}\|_2^2 + \lambda \|\bar{x}\|_1 \quad (8)$$

$$x_{k+1}[S_k] \leftarrow \bar{x}_{k+1},$$

$$x_{k+1}[S_k^c] \leftarrow 0,$$

where \(S_k := \{i_1, \ldots, i_k\} \subset \{1, \ldots, N\}\) denotes the set of the active indices up to iteration \(k\) and \(A_{S_k} \in \mathbb{R}^{\text{Card}(S_k)}\) denotes the restriction of \(A\) to its columns with indices in \(S_k\). Note that in comparison with the initial problem (1), the dimension of (8) is dramatically reduced, from \(N\) to \(\text{Card}(S_k)\) at iteration \(k\).

IV. POLYATOMIC FRANK-WOLFE

Observe that there might exist multiple indices that simultaneously satisfy the approximate gradient minimization step (7), each yielding different but equally valid atomic update directions. Unlike state-of-the-art approaches which limit themselves to selecting one of these directions arbitrarily, we propose taking polyatomic update directions (i.e. expanding the set of active indices with multiple indices at once). This is indeed possible since any convex combination of the potential atomic update directions is also a solution to (7) —although in general not on the boundary of the constraint convex set \(C\) anymore. The idea behind such polyatomic update directions is to accelerate the search space exploration, and consequently the convergence of the algorithm. Polyatomic update steps have already been used in other contexts (see for example [50] for OMP), but to the best of our knowledge, its combination with a FW-type algorithm has not yet been explored.

The main difficulty in working with polyatomic update directions consists in correctly estimating the weights associated to each atom in the convex combination. When a fully-corrective step is additionally performed, choosing the convex optimization weights optimally is no longer a necessity since (8) will anyway re-optimize these weights. We therefore propose to use suboptimal polyatomic update steps with arbitrary weights (e.g. the mean of all atoms) followed by fully-corrective steps. In practice, to further reduce the computational cost of each iteration, we also propose to prematurely stop the fully-corrective steps, resulting in partially-corrective steps. Thanks to an adaptive stopping criterion, these steps are designed to be more and more accurate as the number of iterations increases. Indeed, early partially-corrective steps need not be very accurate since the iterates evolve a lot from one iteration to the other.

Putting all these ingredients together results in Algorithm 5, which we call the Polyatomic Frank-Wolfe (P-FW) algorithm. The partial-correction procedure is described in Algorithm 6. It consists in a warm-started iterative soft-thresholding algorithm with a stopping criterion based on the relative improvement of the iterates.

Theorem 1 proves the convergence of P-FW, relying on [43, Theorem 1].

**Algorithm 5**: Polyatomic FW (P-FW) of quality \(\delta > 0\)

1. Initialize: \(x_0 \leftarrow 0, S_0 \leftarrow \emptyset\)

   for \(k = 1, 2 \cdots \) do

   \[\gamma_k \leftarrow 2 / (k + 2)\]

   1'.a) Polyatomic exploration:

   \(I_k = \{1 \leq j \leq N \mid |\eta_k(j)| \geq |\eta_k|_{\infty} - \delta \gamma_k\}\)

   \(s_k \leftarrow (\sum_{i \in I_k} e_i) / \text{Card}(I_k)\)

   1'.b) Update active indices: \(S_k \leftarrow S_{k-1} \cup I_k\)

   2'.a) Set accuracy threshold: \(\varepsilon_k = \gamma_k \delta \)

   2'.b) Update active weights:

   \(x_{k+1/2} \leftarrow (1 - \gamma_k) x_k + \gamma_k s_k\)

   \(x_{k+1} \leftarrow \text{partial_correction}(x_{k+1/2}, S_k, \varepsilon_k)\)

   \(k \leftarrow k + 1\)

end
ISTA is monotonic and initialized with $L_0$ being any convex combination of solutions of (7). Then we let each algorithm run for a maximum of $3\times 10^4$ iterations. Denote by $\mathcal{L}(x) = \|y - Ax\|_2^2 + \lambda \|x\|_1$ the LASSO cost function and $\mathcal{L}^*$ its minimal value. For each $k \geq 1$, the iterates $x_k$ satisfy

$$\mathcal{L}(x_k) - \mathcal{L}^* \leq \frac{2}{k+2}(C_f + 2\delta)$$

where $C_f$ is the curvature constant of (4).

**Proof.** First, we remark that Lemma 5 of [43] holds for $s_k$ being any convex combination of solutions of (7). Then we only need $\mathcal{L}(x_{k+1}) \leq \mathcal{L}(x_{k+1/2})$ to apply the proof of [43]. This is necessarily the case since the partial correction solver ISTA is monotonic and initialized with $x_{k+1/2}$.

Note that Theorem 1 only provides an upper bound on the convergence rate of Algorithm 5. In practice, the algorithm tends to converge much faster, as illustrated in the next section.

### V. Numerical Simulations

For our simulations, we consider the following data model:

$$y = Ax_0 + w,$$

where $x_0 \in \mathbb{R}^N$ is a $k$-sparse vector of dimension $N = 16384$, $A \in \mathbb{R}^{L \times N}$ is a Gaussian random matrix and $w$ an additive Gaussian white noise with a PSNR of 20 dB. We set the number of measurements to $L = \alpha K \ll N$ for some oversampling factor $\alpha > 1$. We then solve a LASSO problem (1) with input data $y$ with four algorithms:\n**FISTA, Vanilla FW (V-FW)** (with optimal line search, Algorithm 2), **Fully-Corrective FW (FC-FW)** Algorithm 4, and our proposed **Polyatomic FW (P-FW)**. We let each algorithm run for a maximum of 4 seconds and monitor the evolution of the LASSO objective functional value across reconstruction time. We perform this experiment for various values of $K$ and $\alpha$, each time re-running the algorithms for 15 independent noise realizations. Fig. 1 shows the median LASSO objective functional value vs time, as well as the interquartile distance (shaded).

Although FC-FW is slightly faster than V-FW, these two algorithms are significantly slower to converge than P-FW or FISTA. Remarkably, P-FW is able to outperform the optimal first-order method FISTA in most setups. The most striking example comes from configuration (d) of Fig. 1, in which FISTA takes on average more than 2.0s to reach an objective functional value reached by P-FW in less than 0.5s ($\sim 4$× speedup). Furthermore, we remark that the performances of V-FW and FC-FW degrade with larger values of $K$ (ground truth $x_0$ less sparse), whereas this is not the case for P-FW.

Our implementation is based on the *Pycsou* optimization package [51].

### VI. Conclusion

In this paper, we introduce a novel polyatomic FW algorithm, tailored to large-scale LASSO problems. Our procedure leverages polyatomic update directions and partially-corrective re-optimization steps for fast convergence. By preserving sparse intermediate iterates, the algorithm is well-suited for high dimensional regression problems. As revealed by our numerical simulations, the algorithm converges faster than standard state-of-the-art LASSO solvers (from 1.5 to 20 times faster depending on the setup and algorithm), notably including the optimal first-order method FISTA. Future work will include the application of this promising algorithm to the challenging problems of sparse radio-interferometric imaging [1].

**Acknowledgement.** The authors are grateful to Martin Vetterli for his expert advice as well as Quentin Denoyelle and Eric Bezzam for their useful feedback and/or proofreading on earlier versions of this work.
REFERENCES

[1] Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife, and P. Vanderheyden, “Compressed sensing imaging techniques for radio interferometry,” Monthly Notices of the Royal Astronomical Society, vol. 395, pp. 1733–1742, May 2009.

[2] H. Pan, M. Simeoni, P. Hurley, T. Blu, and M. Vetterli, “LEAP: Looking beyond pixels with continuous-space Estimation of Point sources,” Astronomy & Astrophysics, vol. 608, p. A136, Dec. 2017.

[3] M. Simeoni, “Towards more accurate and efficient beamformed radio interferometry imaging,” tech. rep., EPFL, 2015.

[4] M. Simeoni and P. Hurley, “SIML: Sieved Maximum Likelihood for array signal processing,” in IEEE ICCASP 2021, pp. 4535–4539, IEEE, 2021.

[5] M. Simeoni, S. Kashani, P. Hurley, and M. Vetterli, “Deepwave: a recurrent neural-network for real-time acoustic imaging,” Advances In Neural Information Processing Systems 32 (Nips 2019), vol. 32, no. CONF, 2019.

[6] M. Simeoni, “Functional penalised basis pursuit on spheres,” Applied and Computational Harmonic Analysis, vol. 53, pp. 1–53, 2021.

[7] S. Ravishankar, J. C. Ye, and J. A. Fessler, “Image Reconstruction: From Sparsity to Data-adaptive Methods and Machine Learning,” Proceedings of the IEEE, vol. 108, pp. 86–109, Jan. 2020.

[8] A. Besson, L. Roquette, D. Perdios, M. Simeoni, M. Arditi, P. Hurley, Y. Wiaux, and J.-F. Thiran, “A physical model of nonstationary blur in ultrasound imaging,” IEEE Transactions on Computational Imaging, vol. 5, no. 3, pp. 381–394, 2019.

[9] L. Roquette, M. Simeoni, and P. Hurley, “A functional framework for ultrasound imaging,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1837–1841, IEEE, 2018.

[10] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 58, no. 1, pp. 267–288, 1996.

[11] R. Tibshirani, “The lasso problem and uniqueness,” Advances In Neural Information Processing Systems 32 (Nips 2019), vol. 32, no. CONF, 2019.

[12] K. Bredies, M. Carioni, S. Fanzon, and D. Walter, “Linear convergence of accelerated generalized conditional gradient methods,” arXiv:2110.06756 [math], Oct. 2021.

[13] N. Boyd, G. Schiebinger, and B. Recht, “The Alternating Descent Conditional Gradient Method for Sparse Inverse Problems,” arXiv:1507.01562 [math], Jul. 2015.

[14] E. Frandi, R. Nanculef, S. Lodi, C. Sartori, and J. A. K. Suykens, “An algorithm for quadratic programming,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 59, no. 4, pp. 769–793, 2007.

[15] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[16] Y. Zhao and X. Huo, “A homotopic method to solve the lasso problems under an l1 penalty or an l1 constraint,” in Advances in Neural Information Processing Systems 32 (Nips 2019), vol. 32, no. CONF, 2019.

[17] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” SIAM Journal on Imaging Sciences, vol. 2, pp. 183–202, Jan. 2009.

[18] M. Frank and P. Wolfe, “An algorithm for quadratic programming,” Optimization, vol. 5, no. 1-2, pp. 95–110, 1974.

[19] Y. Wiaux, L. Jacques, G. Puy, and P. Vandergheynst, “Reciprocal sampling for ML estimation of sparse signals,” arXiv:1507.01562 [math], Jul. 2015.

[20] J. Fageot and M. Simeoni, “Towards more accurate and efficient beamformed radio interferometry imaging,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 58, no. 1, pp. 267–288, 1996.

[21] J. Fageot and M. Simeoni, “Functional penalised basis pursuit on spheres,” Applied and Computational Harmonic Analysis, vol. 53, pp. 1–53, 2021.

[22] S. Ravishankar, J. C. Ye, and J. A. Fessler, “Image Reconstruction: From Sparsity to Data-adaptive Methods and Machine Learning,” Proceedings of the IEEE, vol. 108, pp. 86–109, Jan. 2020.

[23] M. Unser, “A unifying representer theorem for inverse problems and sparsity constraints,” Journal on Optimization Theory and Applications, vol. 20, no. 6, pp. 2807–2832, 2010.

[24] A. Badoual, J. Fageot, and M. Unser, “Periodic splines and gaussian processes for the resolution of linear inverse problems,” IEEE Transactions on Signal Processing, vol. 66, no. 22, pp. 6047–6061, 2018.

[25] H. Pan, M. Simeoni, P. Hurley, T. Blu, and M. Vetterli, “LEAP: Looking beyond pixels with continuous-space Estimation of Point sources,” Astronomy & Astrophysics, vol. 608, p. A136, Dec. 2017.

[26] T. Debarre, Q. Denoyelle, M. Unser, and J. Fageot, “Sparsest continuous piecewise-linear representation of data,” arXiv preprint arXiv:1302.1061, 2013.

[27] B. Efron, T. Hastie, J. Johnstone, and R. Tibshirani, “Least angle regression,” The Annals of Statistics, vol. 32, pp. 407–499, Apr. 2004. Publisher: Institute of Mathematical Statistics.

[28] B. A. Turlach, “On algorithms for solving least squares problems under an l1 penalty or an l1 constraint,” in 2004 Proceedings of the American Statistical Association, Statistical Computing Section [CD-ROM], pp. 2572–2577, Citeseer, 2005.

[29] Z. Harchaoui, A. Juditsky, and A. Nemirovski, “Conditional gradient algorithms for machine learning,” in NIPS Workshop on Optimization for ML, vol. 3, pp. 3–2, Citeseer, 2012.

[30] S. Roquette, M. Simeoni, and P. Hurley, “A functional framework for ultrasound imaging,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1837–1841, IEEE, 2018.

[31] E. Frandi, R. Nanculef, S. Lodi, C. Sartori, and J. A. K. Suykens, “An algorithm for quadratic programming,” Journal of the Royal Statistical Society. Series B (Methodological), vol. 58, no. 1, pp. 267–288, 1996.

[32] B. Efron, T. Hastie, J. Johnstone, and R. Tibshirani, “Least angle regression,” The Annals of Statistics, vol. 32, pp. 407–499, Apr. 2004. Publisher: Institute of Mathematical Statistics.

[33] B. A. Turlach, “On algorithms for solving least squares problems under an l1 penalty or an l1 constraint,” in 2004 Proceedings of the American Statistical Association, Statistical Computing Section [CD-ROM], pp. 2572–2577, Citeseer, 2005.

[34] Z. Harchaoui, A. Juditsky, and A. Nemirovski, “Conditional gradient algorithms for machine learning,” in NIPS Workshop on Optimization for ML, vol. 3, pp. 3–2, Citeseer, 2012.

[35] S. Roquette, M. Simeoni, and P. Hurley, “A functional framework for ultrasound imaging,” in 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1837–1841, IEEE, 2018.

[36] Y. Zhao and X. Huo, “A homotopic method to solve the lasso problems under an l1 penalty or an l1 constraint,” in Advances in Neural Information Processing Systems 32 (Nips 2019), vol. 32, no. CONF, 2019.

[37] A. Badoual, J. Fageot, and M. Unser, “Periodic splines and gaussian processes for the resolution of linear inverse problems,” IEEE Transactions on Signal Processing, vol. 66, no. 22, pp. 6047–6061, 2018.

[38] M. Simeoni, “Functional penalised basis pursuit on spheres,” Applied and Computational Harmonic Analysis, New York, NY: Springer New York, 2013.