Comparing FISTA and ADMM for ℓ_1-Regularized Optimization: A LASSO-Centric Study

Xingtong Liu

December 18, 2025

Abstract

The project provides an empirical comparison of FISTA and ADMM for ℓ_1-regularized optimization, focusing primarily on the LASSO problem with supplementary analysis of logistic regression and generalized LASSO. Through controlled experiments on synthetic data, we examine convergence behavior, computational efficiency, and parameter sensitivity. We find that while ADMM achieves faster convergence in iteration count and earlier support identification, FISTA maintains significantly lower per-iteration cost, making it more efficient in wall-clock time for moderate-sized problems. Our results verify that local linear convergence occurs when algorithms correctly identify the solution's active support set, confirming theoretical predictions about strict complementarity conditions.

AI Assistance: AI was used for code debugging, literature review, and manuscript editing. All algorithm design, experiments, and interpretations were conducted by the author.
Deviation from Proposal: While the original proposal focused on ℓ_1-regularized logistic regression, the final project emphasizes a detailed analysis of the LASSO problem (Section 3). Logistic regression is therefore discussed only briefly in Section 4.

1 Introduction

ℓ_1-regularized optimization has become a cornerstone of modern statistical learning and signal processing. Problems of the form

$$\min_{x\in\mathbb{R}^n} \quad f(x) + \lambda\|x\|_1, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is smooth and convex, arise in applications such as sparse regression, signal and image processing, and compressed sensing. The ℓ_1 penalty promotes sparsity, enabling interpretable models and efficient representations in high-dimensional settings. The computational challenge lies in the composite structure of (1): while f is smooth and amenable to gradient-based methods, the ℓ_1 term is non-smooth, requiring specialized algorithms. Two main approaches have emerged:

- **FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)** [1]: An accelerated proximal gradient method with $\mathcal{O}(1/k^2)$ theoretical convergence, building on Nesterov's momentum techniques.
- **ADMM (Alternating Direction Method of Multipliers)** [2, 3]: A splitting method with with $\mathcal{O}(1/k)$ theoretical convergence that decouples f and the ℓ_1 penalty through augmented Lagrangian relaxation, offering flexibility at the cost of introducing a penalty parameter ρ.

While both FISTA and ADMM are popular first-order methods capable of solving this problem, they are built upon distinct theoretical frameworks—proximal gradients versus variable splitting. Despite extensive theoretical studies on each individually, there is a lack of systematic empirical comparison in existing literature regarding their relative performance under controlled conditions. This motivates our investigation.

This work provides a comprehensive experimental study comparing FISTA and ADMM on ℓ_1-regularized problems, with a primary focus on the LASSO problem:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \tag{2}$$

We also briefly illustrate the applicability of the methods to ℓ_1 regularized logistic regression

$$\min_x \sum_{i=1}^{m} \log(1 + \exp(-y_i a_i^\top x)) + \lambda\|x\|_1 \tag{3}$$

and the feasibility of two methods to the generalized LASSO.

Through carefully designed experiments, we examine convergence behavior, parameter sensitivity, and computational efficiency. Our findings provide practical guidelines for algorithm selection and parameter tuning.

2 Preliminaries

2.1 Proximal Gradient Methods (ISTA and FISTA)

The ISTA is the standard first-order method for solving (1). It iterates by performing a gradient descent step on the smooth term f followed by the proximal operator of the non-smooth term:

$$x_{k+1} = \mathcal{S}_{\lambda/L}\left(x_k - \frac{1}{L}\nabla f(x_k)\right). \tag{4}$$

where $S_\kappa(a) = \operatorname{sgn}(a)\max(|a| - \kappa, 0)$ is the soft-thresholding operator. While simple, ISTA converges slowly with a global rate of $\mathcal{O}(1/k)$. To accelerate this, FISTA [1] employs a Nesterov momentum. Let $x_0 = y_0$ and $t_1 = 1$. The update modifies the ISTA step to be applied on an extrapolated point y_k:

$$x_k = \mathcal{S}_{\lambda/L}\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right), \tag{5}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{6}$$

$$y_k = x_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_k - x_{k-1}). \tag{7}$$

This modification improves the global convergence rate to $\mathcal{O}(1/k^2)$ while maintaining the same computational complexity per iteration.

2.2 Alternating Direction Method of Multipliers (ADMM)

ADMM [2] solves (1) via variable splitting: $\min f(x) + \lambda\|z\|_1$ s.t. $x - z = 0$. Using the scaled Augmented Lagrangian $\mathcal{L}_\rho(x, z, u) = f(x) + \lambda\|z\|_1 + \frac{\rho}{2}\|x - z + u\|_2^2$, the iterations are:

$$x_{k+1} := \operatorname*{argmin}_x \left(f(x) + \frac{\rho}{2}\|x - z_k + u_k\|_2^2\right), \tag{8}$$

$$z_{k+1} := \mathcal{S}_{\lambda/\rho}(x_{k+1} + u_k), \tag{9}$$

$$u_{k+1} := u_k + x_{k+1} - z_{k+1}. \tag{10}$$

For the LASSO case, the x-update reduces to a linear system $(A^\top A + \rho I)x_{k+1} = A^\top b + \rho(z_k - u_k)$, where the matrix factorization can be precomputed. The effective threshold for the z-update is λ/ρ.

2.3 Optimality Conditions (KKT)

A point x^* is globally optimal if zero lies in the subdifferential of the objective: $0 \in \nabla f(x^*) + \lambda \partial \|x^*\|_1$. Element-wise, this necessitates:

$$\begin{cases} \nabla f(x_i^*) + \lambda \operatorname{sign}(x_i^*) = 0 & \text{if } x_i^* \neq 0, \\ |\nabla f(x_i^*)| \leq \lambda & \text{if } x_i^* = 0. \end{cases} \tag{11}$$

2.4 Local Linear Convergence Conditions

Although FISTA and ADMM have global sublinear rates, they often exhibit asymptotic linear convergence. Tao et al. [4, 5] proved this occurs if the solution x^* satisfies two key conditions:

1. **Uniqueness:** The optimal solution x^* is unique.
2. **Strict Complementarity:** For all inactive indices $j \notin S = \operatorname{supp}(x^*)$, the gradient is strictly bounded:
$$|(A^\top (Ax^* - b))_j| < \lambda. \tag{12}$$

Under these conditions, the algorithms identify the support S in finite time, effectively reducing the non-smooth problem to smooth optimization on a subspace, thereby triggering local linear convergence.

3 LASSO

3.1 Problem Setup and Experimental Design

In this experiment, we focus on the standard LASSO problem as defined in (2). We conduct numerical simulations on synthetic data to evaluate the convergence behavior of FISTA and ADMM under a controlled environment where the ground truth is known.

Data Generation We generated a synthetic dataset with $m = 50$ observations and $n = 100$ features. The sensing matrix $A \in \mathbb{R}^{m \times n}$ had entries drawn i.i.d. from $N(0, 1)$ and scaled by $1/\sqrt{m}$. The ground truth vector $x_{\text{true}} \in \mathbb{R}^n$ was $K = 10$ sparse, with non-zero entries chosen uniformly at random and drawn from $N(0, 1)$. The observation vector was computed as $b = Ax_{\text{true}}$ (noiseless setting).

Parameter Settings The regularization parameter was set to $\lambda = 0.1$. Both algorithms were initialized with a zero vector $x_0 = \mathbf{0}$. The maximum number of iterations was set to 500 for all trials.

Algorithm Configuration We use the algorithms established in 2.1 and 2.2 to implement the experiments.

- **FISTA:** The Lipschitz constant was computed as $L = \lambda_{\max}(A^\top A) \approx 5.482$. Consequently, we used a constant step size of $\alpha = 1/L \approx 0.182$.
- **ADMM:** We utilized the scaled form of ADMM. The penalty parameter was set to $\rho = 1.0$. The x-update step involves solving a linear system with the matrix $(A^\top A + \rho I)$.

A summary of the experimental parameters is provided in Table 1.

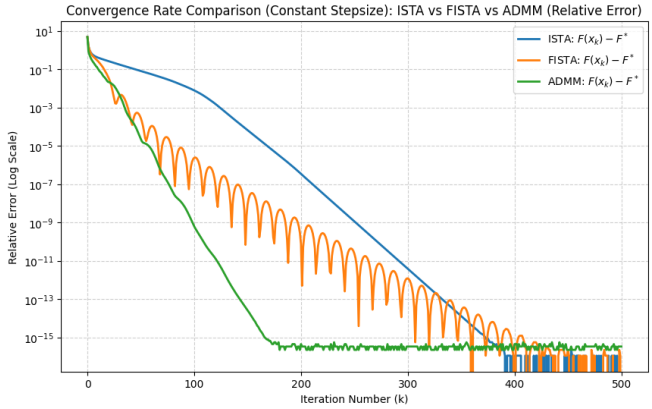
Table 1: *Summary of LASSO Experimental Parameters*

Parameter	Symbol	Value
Number of features	n	100
Number of observations	m	50
Sparsity	K	10
Regularization	λ	0.1
Max Iterations	k_{max}	500
Lipschitz Constant	L	5.482
FISTA(ISTA) Stepsize	α	0.182
ADMM Penalty	ρ	1.0

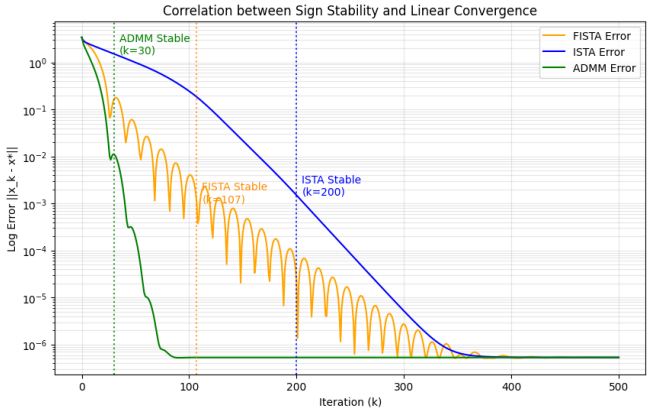
3.2 Comments and Findings

3.2.1 Convergence Analysis

Interestingly, unlike the theoretical global convergence discussed in 2.1 and 2.2, all three algorithms achieved linear convergence, as shown in Figure 1a.



(a) *Convergence Comparison using Constant Stepsize for ISTA and FISTA*



(b) *Support Identification & Stability*

Figure 1: *Comparison of convergence behaviors. (a) The objective function value gap $F(x_k) - F(x^*)$ versus iterations. (b) The solution gap versus iteration for different algorithms*

Local Linear Convergence As Figure 1a shows, FISTA demonstrates a clear acceleration over ISTA, characterized by the oscillatory "ripples" typical of momentum-based methods. ADMM ($\rho = 1.0$) outperforms both gradient-based methods significantly, reaching high precision much earlier.

Verification of Theoretical Conditions This favorable behavior is not guaranteed for all LASSO problems but depends on the solution's properties. As discussed in 2.4, local linear convergence requires *uniqueness* and *strict complementarity*. We numerically verified these conditions for our specific problem instance:

1. **Uniqueness:** We verified that the sub-matrix A_S corresponding to the support set S (with $|S| = 11$)has full column rank, ensuring a unique solution.

2. **Strict Complementarity:** We computed the dual certificate and confirmed that for all zero entries, the gradient magnitude is strictly bounded away from the threshold λ. Our check returned `True` for strict complementarity with a safe margin (max $|\nu| \approx 0.996$ relative to the scaled condition).

These verifications explain why all three tested algorithms consistently attain the local linear convergence rate.

The Onset of Linear Convergence A critical observation from our experiment is the transition to linear convergence(indicated by the straight lines on the semi-log plot). Theory in [4]suggests this transition is triggered when the algorithm correctly identifies the non-zero support of the solution (finite identification of the active manifold).

To verify this, we monitored the *sign stability*, defined as the iteration count k where the sign pattern of the iterate x_k matches the optimal solution x^* (i.e., $\text{sign}(x_k) = \text{sign}(x^*)$). These events are marked by vertical dotted lines in Figure 1b:

- **ADMM** identifies the support earliest at iteration $k = 30$.
- **FISTA** stabilizes the signs at $k = 107$.
- **ISTA** takes the longest, stabilizing at $k = 200$.

ADMM tends to identify the correct support in early iterations due to the explicit proximal update on the ℓ_1 term, which enforces sparsity via soft-thresholding. In contrast, although FISTA has an accelerated global convergence rate, the presence of Nesterov momentum may induce oscillations near the optimal manifold, delaying exact support identification.

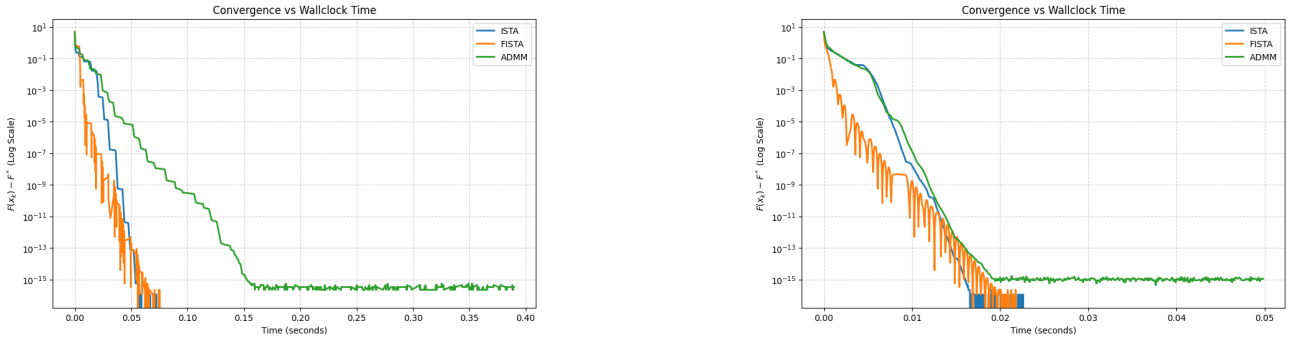
3.2.2 Computational Cost

Apart from the analysis in terms of iteration count, an evaluation of physical runtime (wall-clock time) reveals a crucial trade-off between convergence rate and per-iteration computational complexity. Figure 2a and Figure 2b plot the objective function gap against the CPU execution time. Unlike the iteration-based plot, in Figure 2a, ISTA and FISTA significantly outperform ADMM in terms of raw speed for this problem instance.

Complexity Analysis This reversal can be attributed to the computational cost per iteration:

- **Gradient Methods (ISTA/FISTA):** The dominant operations are matrix-vector multiplications ($A^\top(Ax-b)$), which have a complexity of $\mathcal{O}(mn)$. For our problem size ($m = 50, n = 100$), these operations are extremely lightweight and highly optimized in standard numerical libraries.
- **ADMM:** With naive implementation, it requires $\mathcal{O}(n^3)$ of $(A^\top A+\rho I)$ and $\mathcal{O}(n^2)$ per iteration. Using Matrix Inversion Lemma to conduct an optimized ADMM mentioned in [2], the cost per iteration is reduced to $\mathcal{O}(mn)$.

This suggests that FISTA may be more favorable when $m \ll n$, as it avoids the cubic preprocessing cost entirely.



(a) *Naive ADMM* (b) *Optimized ADMM*

Figure 2: *Comparison of wallclock time between ISTA, FISTA, and ADMM: (a) Naive implementation using $O(n^2)$ updates; (b) Optimized implementation using the Matrix Inversion Lemma.*

3.2.3 Parameter Sensitivity

While the previous section established the convergence properties under fixed, ideal settings, practical implementations must account for hyperparameter selection. We now investigate how sensitive the algorithms are to their key parameters.

ADMM: Sensitivity to Penalty Parameter ρ Unlike FISTA, whose step size is bounded by the Lipschitz constant, ADMM relies on the penalty parameter ρ to balance the convergence of primal and dual residuals. We conducted an empirical sensitivity analysis by varying ρ across a logarithmic scale from 10^{-2} to 10^{2} and measuring the asymptotic convergence rate τ (estimated from the slope of the linear phase).

As illustrated in Figure 3, the performance of ADMM exhibits a characteristic dependence on ρ:

- Small ρ ($\rho < 0.1$): The penalty for constraint violation is too weak, leading to slow dual convergence and oscillatory behavior in the updates.
- Large ρ ($\rho > 10$): The penalty is too strong, forcing the primal variable to stay too close to the auxiliary variable z, which reduces the effective change per iteration and slows overall progress.
- Optimal Region: The algorithm achieves its fastest convergence (lowest rate $\tau \approx 0.75$) in the vicinity of $\rho \approx 1.0 \sim 2.0$.

This result validates our choice of $\rho = 1.0$ in the main experiment. It suggests that ADMM is relatively robust within the order of magnitude of the optimal parameter, but performance degrades significantly at extremes.

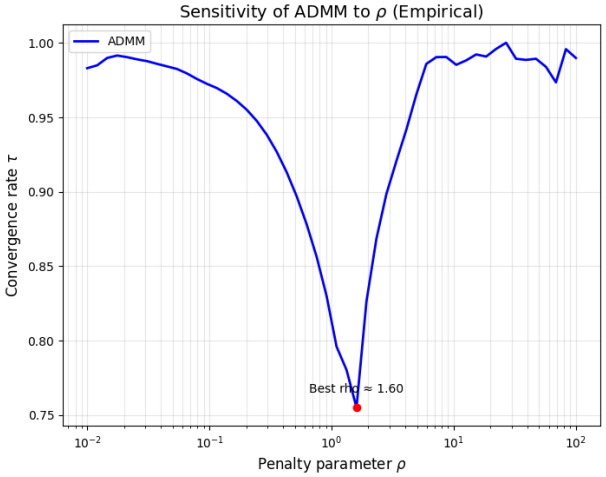


Figure 3: *Empirical sensitivity of ADMM convergence rate to the penalty parameter ρ. The red dot marks the optimal observed $\rho \approx 1.60$.*

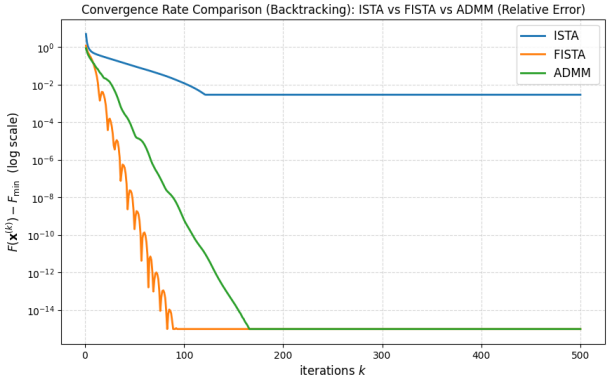


Figure 4: *Convergence comparison using Backtracking Line Search. FISTA maintains accelerated linear convergence without pre-computing L.*

FISTA with Backtracking In the previous experiments, we utilized a constant step size $\alpha = 1/L$, which requires the explicit computation of the Lipschitz constant $L = \lambda_{\max}(A^\top A)$. In large-scale scenarios, computing this spectral norm is computationally prohibitive.

To evaluate the algorithm's robustness when L is unknown or estimated, we implemented FISTA in the same experiment setting with a backtracking line search. This strategy adaptively estimates the local curvature, removing the dependency on a pre-computed L.

Figure 4 compares the convergence trajectories using this adaptive scheme.

- Adaptivity: FISTA with backtracking successfully recovers the linear convergence rate, reaching machine precision ($\sim 10^{-15}$). This demonstrates that the acceleration mechanism is robust and does not rigidly require a fixed step size $1/L$.

- Comparison: The adaptive FISTA matches the asymptotic performance of ADMM, confirming that lack of knowledge about the global Lipschitz constant does not hinder the algorithm's ability to achieve optimal sparsity recovery.

4 Discussion Beyond LASSO

4.1 ℓ_1 logistic regression

In the experiment of ℓ_1-regularized logistic regression (3), we generate synthetic datasets with the following settings: the number of samples $M = 500$, the number of features $N = 200$, and the regularization parameter $\lambda = 0.1$. The true weight vector $\boldsymbol{W}_{\text{true}}$ is set to have a sparsity of 0.5 (i.e., 50% of the elements are zero), and the dataset $(\boldsymbol{X}, \boldsymbol{Y})$ is generated based on $\boldsymbol{W}_{\text{true}}$

In our experiments (see in Figure 5), FISTA consistently achieves lower objective values in less wall-clock time than ADMM. The logistic loss lacks a closed-form solution in the ADMM x-update, necessitating inner iterations (e.g., Newton method) which increase the computational cost per iteration.

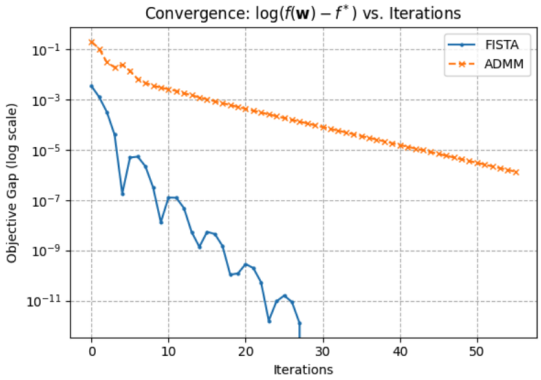


Figure 5: *Convergence of FISTA and ADMM in ℓ_1 logistic regression*

4.2 Generalized LASSO

Consider the problem

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \lambda\|Fx\|_1 \tag{13}$$

where F is a linear transformation. We compare the feasibility of FISTA and ADMM for solving this specific problem.

For FISTA, the iteration step for FISTA is given by:

$$x^{k+1} = \text{prox}_{t\lambda\|F(\cdot)\|_1}\left(y^k - tA^T(Ay^k - b)\right) \tag{14}$$

where the proximal operator is defined as:

$$\text{prox}_{t\lambda\|F(\cdot)\|_1}(y) = \arg\min_x \left(\lambda\|Fx\|_1 + \frac{1}{2t}\|x - y\|^2\right) \tag{15}$$

For $F = I$, the problem is the standard LASSO which we discussed in Section 3. For a general matrix F, this proximal operator does not have a closed-form solution. Computing this step would require a nested iterative solver, which significantly increases the computational overhead and makes FISTA less efficient for Generalized LASSO problems.

In contrast, ADMM introduces an auxiliary variable $z = Fx$, the problem is reformulated as:

$$\min_{x,z} \frac{1}{2}\|Ax - b\|^2 + \lambda\|z\|_1 \quad \text{subject to } Fx = z \tag{16}$$

The ADMM update steps are:

$$x^{k+1} := (A^T A + \rho F^T F)^{-1}(A^T b + \rho F^T (z^k - u^k)) \tag{17}$$

$$z^{k+1} := S_{\lambda/\rho}(Fx^{k+1} + u^k) \tag{18}$$

$$u^{k+1} := u^k + Fx^{k+1} - z^{k+1} \tag{19}$$

ADMM is highly feasible because it decouples the linear operator F from the ℓ_1 norm. The z-update becomes a simple element-wise soft-thresholding, and the x-update is a linear system. Since the matrix $(A^T A + \rho F^T F)$ remains constant across iterations, its factorization (e.g., Cholesky) can be pre-computed, making each iteration extremely fast.

In summary, while FISTA is efficient for standard LASSO ($F = I$), it becomes computationally prohibitive for Generalized LASSO due to the intractable proximal mapping. ADMM is the preferred choice as it simplifies the complex objective into efficient, closed-form sub-problems.

5 Conclusion

5.1 Main Findings

In this project, we observed that in the LASSO problem, ISTA, FISTA, and ADMM all exhibit a transition to a faster local convergence rate after the underlying support of the solution is identified. In the experiment in 3, ADMM typically enters this local linear convergence regime more rapidly. While FISTA is accelerated, the Nesterov momentum induces oscillations that can delay exact convergence to the manifold. In terms of computational cost, FISTA maintains a lower per-iteration cost due to its efficient soft-thresholding updates. In contrast, ADMM's requirement for solving a linear system makes each iteration significantly more expensive. We found that ADMM is highly sensitive to the penalty parameter ρ, which governs the balance between primal and dual feasibility, while FISTA with adaptive backtracking provides a practical alternative when the Lipschitz constant is unknown. FISTA is preferable when the problem difficulty lies in the smooth term, like ℓ_1 logistic regression, while the nonsmooth term remains separable with a cheap proximal operator. ADMM is more suitable when the difficulty lies in the structure of the nonsmooth term, like the Generalized LASSO, which destroys coordinate-wise separability and can be handled via variable splitting.

5.2 Limitations and Future Work

While this project provides a comparative analysis of FISTA and ADMM specifically for the LASSO problem, several limitations remain to be addressed in future research:

1. Comprehensive Numerical Evaluation: Although ℓ_1 logistic regression and generalized LASSO were briefly discussed, more extensive numerical experiments are required to evaluate the algorithms' stability and convergence rates across a broader range of loss functions and penalty structures.

2. Scalability to Large-Scale Data: The current implementation primarily focuses on moderate-sized datasets. Extending the framework to large-scale settings, where memory efficiency and parallel computation become critical, remains a significant challenge.

3. Algorithmic Hybridization: Future work could explore a hybrid paradigm that leverages the respective strengths of ADMM and FISTA. For instance, integrating the fast convergence of FISTA with the decoupling flexibility of ADMM could potentially yield a more robust solver for structured sparsity problems.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[3] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

[4] Shaozhe Tao, Daniel Boley, and Shuzhong Zhang. Convergence of common proximal methods for l1-regularized least squares. In *IJCAI*, pages 3849–3855, 2015.

[5] Shaozhe Tao, Daniel Boley, and Shuzhong Zhang. Local linear convergence of ista and fista on the lasso problem. *SIAM Journal on Optimization*, 26(1):313–336, 2016.