**Electronic Transactions on Numerical Analysis.**
**Volume 65, pp. 93–109, 2026.**
**© 2026, the Author(s), licensed via CC BY 4.0.**
**ISSN 1068–9613.**
**DOI: 10.1553/etna_vol65s93**

**ETNA**
Kent State University and
Johann Radon Institute (RICAM)

# BIASED SUPERIORIZATION OF STEEPEST DESCENT: REDEFINING THE RECONSTRUCTION TARGET IN NOISY INVERSE PROBLEMS*

TOURAJ NIKAZAD† AND MOKHTAR ABBASI‡

**Abstract.** Reconstruction in inverse problems is commonly performed by minimizing a data-fidelity term, such as the least-squares objective. However, when the data are noisy, this minimizer may not represent the true solution. The superiorization methodology attempts to improve the structure of the solution by perturbing iterative schemes, but such methods still target the noisy data-fit minimizer. This work proposes a paradigm shift: rather than improving convergence to a flawed target, we redefine the reconstruction goal itself. We introduce the Superiorized Biased Steepest Descent (S-BSD-LS) algorithm, which combines two key components: (i) a deliberately biased residual update that decouples the method from the noisy least-squares solution and (ii) a total variation (TV) perturbation that promotes a desirable structure. We rigorously analyze convergence of the algorithm using an auxiliary sequence and characterize the systematic bias induced by both the residual update and the data noise. This approach offers a new perspective on solving ill-posed noisy problems. We demonstrate the performance of our proposed method, S-BSD-LS, for tomographic imaging examples using total variation regularization as a specific instance of the general framework and compare it with state-of-the-art methods, including the standard steepest descent method, the conjugate gradient method for least-squares (CGLS) , the superiorized conjugate gradient method with conjugate descent (S-CG-CD), and FISTA, a well-known proximal gradient algorithm. In each iteration, S-BSD-LS requires two matrix-vector multiplications—similar to CGLS—and one additional function evaluation compared to CGLS while still maintaining competitive performance. Additional experiments confirm that the superiorization mechanism is primarily responsible for the improved stability and a continuous error reduction while the biased residual update mainly enhances computational efficiency.

**Key words.** inverse problems, superiorized, biased steepest descent, tomographic imaging, total variation

**AMS subject classifications.** 47J25, 49M20, 90C25

## 1. Introduction.

In many inverse problems, particularly in imaging applications, the reconstruction from projection data is modeled by a large-scale linear system,

$$Ax = b^{\delta}, \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the unknown image, $A \in \mathbb{R}^{m \times n}$ is the forward operator, and $b^{\delta} \in \mathbb{R}^m$ contains the measured data. Due to model inaccuracies and measurement errors, $b^{\delta}$ is typically contaminated by noise, that is, $b^{\delta} = b + \delta b$, where $\delta b$ represents the noise in the data. The system (1.1) can be quite ill-conditioned, making reconstructions particularly challenging. A common approach is to solve the least-squares (LS) problem,

$$\min_x \tfrac{1}{2}\|Ax - b^{\delta}\|^2, \tag{1.2}$$

often by using gradient-based iterative methods such as Steepest Descent (SD). However, due to the presence of noise and ill-conditioning, these methods are susceptible to the well-known phenomenon of *semi-convergence* [34, p. 89], where early iterates improve the solution but later iterations amplify noise. Traditionally, this issue is addressed using stopping rules such as the discrepancy principle, the L-curve method, or generalized cross-validation (GCV).

*Received July 20, 2025. Accepted December 19, 2025. Published online on February 13, 2026. Recommended by Stefan Kindermann. The first author gratefully acknowledges partial financial support from the Isaac Newton Institute for Mathematical Sciences and the London Mathematical Society.

†Applied Mathematics Department, School of Mathematics and Computer Sciences, Iran University of Science and Technology, Tehran, 16846-13114, Tehran, Iran.

‡Corresponding author. Department of Mathematics, University of Qom, Qom, 37161-46611, Qom, Iran (m.abbasi@qom.ac.ir).

This work is published by ETNA and licensed under the Creative Commons license CC BY 4.0.

**ETNA**
**Kent State University and**
**Johann Radon Institute (RICAM)**



However, these rules come with significant limitations. For instance, some require prior knowledge of the noise level [26], which is rarely available in practical settings. Others, like the L-curve, can select poor stopping points when the underlying solution is too smooth or when the singular spectrum exhibits atypical behavior [24]. Furthermore, determining the optimal stopping point often incurs additional computational cost, especially in large-scale problems [29]. In particular, estimating the noise level—often required by methods such as the discrepancy principle—can be costly. For instance, [30] uses Golub–Kahan iterative bidiagonalization to approximate the noise level. Even conceptually, stopping too early leads to underfitting while stopping too late allows noise to distort the result [3, 23].

These challenges motivate the development of methods that inherently *control* semi-convergence, reducing or eliminating the reliance on fragile stopping criteria. For example, embedding averaging within the iterations has been shown to effectively dampen noise accumulation without external stopping [22]. Superiorization techniques similarly guide iterates toward regularized solutions through built-in perturbations [21]. Our proposed method, S-BSD-LS is designed according to the following principle: it integrates a biased residual update with a regularization objective (such as total variation) to steer the iterates toward high-quality solutions. We note that classical total variation promotes piecewise-constant reconstructions; it is used here as a canonical and well-understood example while other regularizers (e.g., TGV) could be incorporated within the same framework. Each iteration of S-BSD-LS requires two matrix-vector multiplications—as in CGLS—and one additional function evaluation, making it computationally efficient while enhancing the solution quality.

To mitigate the above-mentioned limitations due to semi-convergence, the incorporation of *a priori* information has become essential in improving both the quality and the stability of the reconstructed solutions [8, 43]. In many applications, known physical or statistical properties of the underlying signal—such as non-negativity [20, 31], sparsity [19, 42], or piecewise smoothness [41]—are imposed as constraints or regularizers. Among these, total variation (TV) regularization has proven effective in promoting piecewise-smooth structures that are typical for natural and medical images.

While this paper presents the Superiorized Biased Steepest Descent algorithm in a general framework that can accommodate any convex regularization functional, we focus on total variation in our numerical experiments for several reasons. TV regularization is well-established in imaging applications, promotes desirable piecewise-constant structures, and serves as a standard benchmark. The perturbation vectors $v_k$ of the algorithm are chosen as (sub)gradients of the TV functional in Section 5, but the theoretical development in Sections 2–4 applies to any bounded perturbation sequence $v_k$ satisfying the stated assumptions. This generality allows the method to be adapted to other regularization functionals (e.g., $\ell_1$-norm, TGV) as needed for specific applications.

The superiorization methodology provides a computationally efficient way to enforce such priors by perturbing the iterates of an existing algorithm (e.g., SD) to reduce a secondary objective (like TV) without significantly compromising the primary data-fidelity reduction. The data-fidelity term, typically expressed as $\frac{1}{2}\|b^\delta - Ax\|^2$, quantifies how well a candidate solution $x$ fits the observed (possibly noisy) measurements $b^\delta$. Minimizing this term aims to ensure consistency with the measured data. Understanding how such perturbations, both explicit (like perturbing the iterates) and implicit (like noise in $b^\delta$), affect the convergence and accuracy of these algorithms is crucial for their reliable application.

However, a subtle but important limitation persists: superiorization and traditional regularization methods still aim to minimize the original data-fidelity term, $\frac{1}{2}\|Ax - b^\delta\|^2$, which is built on noisy data. Since $b^\delta$ includes noise, its minimizer may not correspond to the true solution $x_{\text{true}}$, meaning the algorithm—even if superiorized—is still guided toward a flawed

ETNA
Kent State University and
Johann Radon Institute (RICAM)



target. This work argues that instead of improving the trajectory toward such a minimizer, it is more effective to redefine the destination itself.

To this end, we propose the Superiorized Biased Steepest Descent algorithm, which embodies the perspective that redefining the destination is more effective than merely improving the trajectory toward a fixed minimizer. Departing from strict fidelity minimization, the algorithm integrates two key components into a unified iterative framework: (i) a computationally inexpensive, intentionally inaccurate residual update that introduces a controlled bias and (ii) a TV-based perturbation that promotes a desirable structure. This paper focuses on a specific variant of the steepest descent algorithm that incorporates a deliberate, summable perturbation and explicitly analyzes its behavior in the presence of noisy data.

Our analysis reveals that although convergence to a fixed point is achieved, the combination of the explicit perturbation and the implicit noise in the data introduces a systematic bias, leading the algorithm to converge to a point that is generally not the true least-squares solution for the noise-free problem. A central theme of our investigation is to gain understanding of the specific residual update rule—deliberately differing from the true residual. We formalize our algorithm, provide a rigorous convergence proof using an auxiliary sequence, and clarify the origin and implications of the resulting bias.

The paper is organized as follows. Section 2 provides a brief review of the superiorization methodology and reintroduces the Steepest Descent (SD) method along with its convergence properties. Our main algorithm, the Superiorized Biased Steepest Descent algorithm, is presented in Section 3. Section 4 contains the convergence analysis of the main algorithm, including Theorem 4.2. In this section, we also provide motivation for our biased residual control strategy and its connection to semi-convergence. In Section 5, we demonstrate the effectiveness of our approach through examples from tomographic imaging and conduct comparative evaluations with other relevant methods.

## 2. Preliminaries.

In this section, we first briefly review the superiorization methodology and then reintroduce the Steepest Descent (SD) method along with its convergence properties.

### 2.1. Superiorization methodology.

The superiorization methodology combines optimization techniques (such as minimizing a penalty function) with the process of finding a feasible point (for example, a solution to a linear system of equations). A feasible point is simply a candidate solution that satisfies all the constraints of the problem.

To explain superiorization, suppose we have a fixed-point iterative method of the form $x_{k+1} = \Upsilon(x_k)$ for solving a linear system. In addition, assume there is a cost function $h$, and our goal is to find a feasible solution that also reduces the value of $h$. The key idea behind superiorization is to modify the iteration by adding a term involving the negative (sub)gradient of the cost function. This leads to update rules such as

$$x_{k+1} = \Upsilon(x_k) - \beta_k \nabla h(\Upsilon(x_k)),$$

which aims to maintain convergence toward feasibility while also improving the value of the objective function.

Superiorization is a clever strategy that blends ideas from optimization into feasibility-seeking algorithms. It can play a role similar to regularization methods. The first practical implementations of this approach appeared in [9, 17], though its theoretical foundations were introduced earlier in [10, 11] and related works. Since then, the methodology has been extended in several directions such as in [1, 15, 35, 36, 40, 44]. For an extensive and continuously updated bibliography of works on the superiorization methodology, the reader is referred to [13].

To describe this technique more formally, suppose we are given a sequence of non-negative scalars $\{\beta_k\}$ and a sequence of vectors $\{v_k\}$. An algorithmic operator $\Upsilon : \mathbb{R}^n \to \mathbb{R}^n$ is said to be *bounded perturbations resilient* (BPR) if the following holds:

If the unperturbed iteration $y_{k+1} = \Upsilon(y_k)$ converges to a fixed point of $\Upsilon$ for any initial point $y_0$, then the perturbed sequence $x_{k+1} = \Upsilon(x_k + \beta_k v_k)$ also converges to a fixed point of $\Upsilon$ for any initial point $x_0$, provided that $\sum_k \beta_k < \infty$ and the set $\{v_k\}$ is bounded.

This definition is consistent with the formal statement given in [32, Definition 4]. We denote the set of fixed points of an operator $T$ by $\operatorname{Fix} T$. The terms "bounded" and "resilient" refer to the boundedness of the perturbations $\{v_k\}$ and the robustness with respect to the convergence of the method, respectively.

It is important to note that the goal of superiorization is not to solve the constrained minimization problem $\min_x h(x)$ subject to $x \in \operatorname{Fix} \Upsilon$ exactly. Instead, the perturbations are designed to steer the iterates toward points in $\operatorname{Fix} \Upsilon$ that have reduced (but not necessarily minimal) values of $h$.

In practice, the perturbation vectors $\{v_k\}$ are often chosen as negative (sub)gradients of a given cost function. This guides the iterates toward solutions that are not only feasible but also, in most cases, improved with respect to the objective, in comparison to setting $\beta_k = 0$, which would effectively ignore the optimization aspect. The aim is to achieve a "better" or "more optimal" solution in terms of the objective function.

The superiorization approach can be categorized into two types: *weak* and *strong*, as discussed in [14, 15]. To illustrate the difference, consider a constrained minimization problem. In weak superiorization, as illustrated in works like [9, 17, 35, 36, 44] and [16, Section 4.1], the constraints are assumed to be consistent, meaning that a point exists that satisfies all of them. The approach proceeds in two steps: first, an existing algorithm is shown to be resilient to bounded perturbations. Then, the allowed perturbations are used to guide the iterates toward lower values of the objective while still converging to a feasible point.

In contrast, strong superiorization does not require the constraints to be consistent. Instead, it uses a proximity function to measure how far a point is from satisfying the constraints. The goal becomes achieving $\varepsilon$-compatibility—getting sufficiently close to feasibility—while applying stronger types of perturbation resilience. For more details, see [14, 18, 29, 39, 40] and [16, Section 4.2]. The method we propose in this paper can, in a sense, be regarded as a form of strong superiorization since we do not assume that the linear system (1.1) is consistent.

REMARK 2.1. While most applications of the superiorization methodology are based on feasibility-seeking algorithms, our approach represents a rarer case where superiorization is applied within a constrained optimization framework. Similar extensions can be found in [33, 44], where superiorization principles are adapted to optimization-driven algorithms rather than feasibility problems.

**2.2. The least-squares problem and Steepest Descent.** The minimizers of (1.2) satisfy the normal equations $A^T(Ax - b^\delta) = 0$. The Steepest Descent (SD) method generates a sequence $\{x_k\}$ as described in Algorithm 1 for solving the least-squares problem. We refer to this algorithm as SD-LS. A straightforward calculation shows that this algorithm is equivalent to the Landweber method, as noted in [37, with $s = 1$ in equation (3.8)], and each iteration requires two matrix-vector multiplications involving $A$ and $A^T$.

It is shown in [37, Lemma 3.3] that $\lambda_k \geq 1/\sigma_1^2$, where $\sigma_1$ denotes the largest singular value of the matrix $A$. A general convergence analysis of gradient-based methods, such as steepest descent, can be found in [5], while the specific case of the linear least-squares problem (1.2) is addressed in [37, Theorem 5.4]. Consequently, the sequence generated by Algorithm 1 converges to a least-squares solution of (1.2). Moreover, if $x_0 \in \mathcal{R}(A^T)$, then $x_k$

**ETNA**
**Kent State University and**
**Johann Radon Institute (RICAM)**



**Algorithm 1** Steepest Descent for Least-Squares Problem (SD-LS).

```
Given: Matrix A, vector b^δ.
Initialize: Choose x_0 ∈ ℝ^n.
r_0 := b^δ − A x_0.
for k = 0, 1, 2, ... do
    u_k := A^T r_k.
    if u_k = 0 then stop.
    end if
    s_k := A u_k.
    λ_k := ||u_k||^2 / ||s_k||^2 (optimal step-size).
    Update iterate: x_{k+1} := x_k + λ_k u_k.
    Update residual: r_{k+1} := r_k − λ_k s_k.
end for
```

converges to the minimum-norm solution. For a detailed analysis of the convergence properties of classical Landweber iteration, see [12, Chapters 4 and 5].

**3. The proposed Superiorized Biased Steepest Descent algorithm.** Let $\{\beta_k\}$ be a sequence of non-negative scalars such that $\beta_k \geq 0$ and $\sum_{k=0}^{\infty} \beta_k < \infty$, and let $\{v_k\} \subset \mathbb{R}^n$ be a sequence of bounded vectors satisfying $\|v_k\| \leq \mathcal{M}$ for some constant $\mathcal{M} > 0$. The Superiorized Biased Steepest Descent method for solving the least-squares problem is described in Algorithm 2. We refer to this algorithm as S-BSD-LS. The key step is line 12, where the residual update intentionally omits the effect of the perturbation $\beta_k v_k$.

**Algorithm 2** S-BSD-LS.

```
1:  Given: Matrix A, vector b^δ.
2:  Initialize: Choose x_0 ∈ ℝ^n.
3:  r_0 := b^δ − A x_0.
4:  for k = 0, 1, 2, ... do
5:      u_k := A^T r_k.
6:      if u_k = 0 then stop.
7:      end if
8:      s_k := A u_k.
9:      λ_k := ||u_k||^2 / ||s_k||^2.
10:     Compute parameter β_k and direction v_k.
11:     Update iterate: x_{k+1} := x_k + λ_k u_k + β_k v_k.
12:     Update biased residual: r_{k+1} := r_k − λ_k s_k.
13: end for
```

NOTE 3.1. *Note that while Algorithm 2 is presented in a general form, in our numerical implementation (Section 5) we specifically choose $v_k$ as a normalized subgradient of the total variation functional to promote piecewise-constant reconstructions.*

To understand the combined effect of explicit perturbations and noise, it is helpful to first examine the case in which explicit perturbations are absent. When $\beta_k = 0$ for all $k$, the algorithm reduces to the standard Steepest Descent (SD-LS) method. This classical method is known to converge to the minimizer $x^*_{\text{noisy}}$ of $\frac{1}{2}\|b^\delta - Ax\|^2$, which satisfies the first-order optimality condition $A^T(b^\delta - Ax^*_{\text{noisy}}) = 0$. This condition implies that the residual $b^\delta - Ax^*_{\text{noisy}}$ is orthogonal to the column space of $A$.

It is important to note that $x^*_{\text{noisy}}$ is the least-squares solution corresponding to the noisy right-hand side $b^\delta$ and generally differs from the ideal solution associated with the noise-free vector $b$. Therefore, the algorithm does not converge to the true minimizer of $\frac{1}{2}\|b - Ax\|$. Analogously, we will show that the sequence generated by Algorithm 2 converges to a biased point due to the presence of noise. In fact, when explicit perturbations are added, the conditions $\sum_k \beta_k < \infty$ and the boundedness of $\{v_k\}$ ensure convergence, but the resulting solution remains biased and does not recover the true least-squares solution.

REMARK 3.2. For a symmetric positive definite (SPD) matrix $M$, one may consider the preconditioned system $M^{1/2}Ax = M^{1/2}b$ instead of (1.1), where $M^{1/2}$ denotes the square root of $M$. We note that considering such a system does not affect our analysis, and all results can be readily rewritten in the new setup. For various choices of $M$ and their impact on the iterative performance, see [37].

## 4. Convergence analysis.

The key feature in Algorithm 2 lies in the update rule for $x_{k+1}$, which includes the explicit perturbation term $\beta_k v_k$:

$$x_{k+1} = x_k + \lambda_k u_k + \beta_k v_k.$$

However, a critical step is the update rule of the algorithm for the residual $r_{k+1}$:

$$r_{k+1} = r_k - \lambda_k s_k.$$

This residual update does not account for the perturbation introduced in $x_{k+1}$. The true residual at step $k+1$, based on the actual $x_{k+1}$, would be

$$\begin{aligned}\tilde{r}_{k+1} &= b^\delta - Ax_{k+1} = b^\delta - A(x_k + \lambda_k u_k + \beta_k v_k) = (b^\delta - Ax_k) - \lambda_k Au_k - \beta_k Av_k \\ &= r_k - \lambda_k s_k - \beta_k Av_k = r_{k+1} - \beta_k Av_k.\end{aligned}$$

Thus, for $k > 0$, the computed residual $r_k$ in the algorithm deviates from the true residual by an accumulation of the terms $-\beta_i Av_i$.

REMARK 4.1. The SD-LS algorithm (Algorithm 1) is known to satisfy the BPR property (see [32] or [2, Lemma 5, Proposition 4, and Theorem 3]), which makes it applicable as basic algorithm within the superiorization framework. However, the proposed Algorithm 2 (S-BSD-LS) is not a direct superiorization of Algorithm 1 because the underlying operator uses a biased residual, in contrast to Algorithm 1. In fact, a superiorized version of Algorithm 1 is Algorithm 3 in Section 5.2. Theorem 4.2 establishes convergence of Algorithm 2, demonstrating a form of stability analogous to that guaranteed by the BPR property.

THEOREM 4.2. *Let $\{x_k\}$ be the sequence generated by Algorithm 2. Assume that the sequence of vectors $\{v_k\}$ is bounded and that the sequence of scalars $\{\beta_k\}$ is non-negative and summable ($\sum_{k=0}^{\infty} \beta_k < \infty$). Then $\{x_k\}$ converges to a limit $x^*$ given by*

$$x^* := y^* + S^*,$$

*where $y^*$ is a specific least-squares solution determined by $x_0$ and $S^* := \sum_{k=0}^{\infty} \beta_k v_k$.*

*Proof.* To analyze the convergence of $x_k$, we introduce an auxiliary sequence $y_k$, defined as

$$y_k = x_k - \sum_{i=0}^{k-1} \beta_i v_i,$$

**ETNA**
**Kent State University and**
**Johann Radon Institute (RICAM)**



where $y_0 = x_0$ and $k \geq 1$. The iterative update for $y_{k+1}$ can be rewritten as

$$\begin{aligned} y_{k+1} &= x_{k+1} - \sum_{i=0}^{k} \beta_i v_i = (x_k + \lambda_k u_k + \beta_k v_k) - \left( \sum_{i=0}^{k-1} \beta_i v_i + \beta_k v_k \right) \\ &= x_k - \sum_{i=0}^{k-1} \beta_i v_i + \lambda_k u_k = y_k + \lambda_k u_k \\ &= y_k + \lambda_k A^T r_k. \end{aligned} \tag{4.1}$$

We use mathematical induction to show that in Algorithm 2, the residuals are computed as

$$r_k = b - A y_k. \tag{4.2}$$

For $k = 0$, since $y_0 = x_0$, the relation clearly holds. Suppose that for $k = p$, the relation $r_p = b - A y_p$ holds. We must then show that it also holds for $k = p + 1$.

Using line 12 of Algorithm 2, we get

$$\begin{aligned} r_{p+1} &= r_p - \lambda_p s_p = (b - A y_p) - \lambda_p s_p && \text{(by the inductive hypothesis)} \\ &= (b - A y_p) - \lambda_p A u_p = (b - A y_p) - \lambda_p A (A^T r_p) \\ &= b - A \left( y_p + \lambda_p A^T r_p \right) = b - A y_{p+1} && \text{(by 4.1).} \end{aligned}$$

This completes the induction. Therefore, using (4.2) and (4.1), the sequence $\{y_k\}$ then evolves as

$$y_{k+1} = y_k + \lambda_k A^T (b^\delta - A y_k), \tag{4.3}$$

and we have

$$\lambda_k = \frac{\|u_k\|^2}{\|s_k\|^2} = \frac{\|A^T r_k\|^2}{\|A A^T r_k\|^2}. \tag{4.4}$$

Interestingly, the iterative method (4.3) with relaxation parameter (4.4) coincides with the standard steepest descent method applied to minimize $\frac{1}{2}\|b^\delta - Ay\|^2$. As such, it is known that the sequence $y_k$ converges to a point $y^*$ satisfying the optimality condition for the unperturbed problem with noisy data, namely $A^T(b^\delta - Ay^*) = 0$. This $y^*$ is the least-squares solution of the inconsistent system $Ay = b^\delta$ (see Remark 4.3). From the definition of $y_k$, we have

$$x_k = y_k + \sum_{i=0}^{k-1} \beta_i v_i. \tag{4.5}$$

We are given that $\|v_k\| \leq \mathcal{M}$, for some constant $\mathcal{M} > 0$, and that $\sum_{k=0}^{\infty} \beta_k < \infty$. These conditions ensure that the series $\sum_{i=0}^{\infty} \beta_i v_i$ converges absolutely

$$\sum_{k=0}^{\infty} \|\beta_k v_k\| \leq \mathcal{M} \sum_{k=0}^{\infty} \beta_k < \infty.$$

Therefore, since any absolutely convergent series in $\mathbb{R}^n$ converges, we conclude that $\sum_{k=0}^{\infty} \beta_k$

converges. Given that the sequence $y_k$ converges to $y^*$ and the sum $\sum_{i=0}^{k-1} \beta_i v_i$ converges to $S^*$ as $k \to \infty$, using (4.5), it follows that

$$\lim_{k\to\infty} x_k = y^* + S^* = x^*.$$

Thus, the perturbed algorithm does converge to a specific point $x^*$. ∎

REMARK 4.3. The iterative method (4.3) with relaxation parameters (4.4) is studied in [37, Algorithm 1, p. 37, with $M = I$ and relaxation parameters (3.8), $s = 1$, p. 40]. It is shown in [38, Theorem 6] that the sequence generated by (4.3) with the relaxation parameters (4.4) converges to $y^* = x^* + P_{N(A)}(x_0)$, where $P_{N(A)}(x_0)$ is the orthogonal projection of $x_0$ onto the null space of $A$ and $x^*$ is the unique solution with minimal Euclidean norm of the least-squares equation (1.2). Therefore, if $x_0 \in \mathcal{R}(A^T)$, then the sequence $\{y_k\}$ converges to the unique minimum-norm solution of (1.2). For all methods in our numerical experiments, the initial vector is set to $x_0 = 0$. In practical applications, however, a more suitable starting point may be chosen if a priori information is available.

REMARK 4.4 (The bias term $S^*$). The cumulative explicit perturbation $S^* = \sum_{k=0}^{\infty} \beta_k v_k$ in Theorem 4.2 is precisely the *bias term* introduced by the algorithm. This term represents the systematic deviation from the least-squares solution $y^*$ of the noisy problem. The conditions $\|v_k\| \leq \mathcal{M}$ and $\sum \beta_k < \infty$ ensure that $S^*$ is well defined and finite.

### 4.1. Semi-convergence and the bias of $x^*$.

In the context of iterative methods, semi-convergence refers to the phenomenon that an algorithm initially converges toward a meaningful solution—such as the minimum-norm solution or the true minimizer of the original, noise-free problem—but ultimately deviates due to the influence of noise.

#### 4.1.1. Influence of the biased residual update on semi-convergence.

In standard iterative methods for least-squares problems, such as SD-LS, the phenomenon of semi-convergence is an inherent limitation when dealing with noisy data. These methods are designed to minimize the data-fidelity term $\frac{1}{2}\|b^\delta - Ax\|^2$. While initial iterations typically move the solution closer to the true, noise-free solution, later iterations begin to fit the noise present in the measurements $b^\delta$. Consequently, after reaching an initial optimum, the solution quality degrades as the algorithm amplifies noise, making it a poor choice for ill-posed inverse problems. The proposed Superiorized Biased Steepest Descent (S-BSD-LS) algorithm counters this problem with a fundamental paradigm shift: instead of attempting to converge to a flawed target (the noisy minimizer), it redefines the destination itself. The key to this strategy lies in the residual update rule:

$$r_{k+1} = r_k - \lambda_k A u_k.$$

This update intentionally ignores the effect of the perturbation term $\beta_k v_k$, which is concurrently added to the iterate $x_k$. This deliberate decoupling is the cornerstone of controlling semi-convergence and has profound implications for the stability and behavior of the algorithm:

- Establishing a stable underlying process: As demonstrated in the proof of Theorem 4.2, this biased update forces the auxiliary sequence $y_k$ to follow a standard, unperturbed steepest descent process for the noisy least-squares problem,

  $$y_{k+1} = y_k + \lambda_k A^T(b^\delta - Ay_k).$$

  Because this standard steepest descent process is guaranteed to converge to a least-squares solution $y^*$, our algorithm maintains a predictable and stable underlying trajectory. Unlike traditional methods that are directly susceptible to the instabilities of the noisy objective, our algorithm is anchored to this well-behaved dynamic.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



- Enabling predictable bias accumulation: By forgoing the goal of exact residual minimization at each step, the algorithm avoids chasing noise effects. Instead, the full iterate sequence $x_k$ converges to a well-defined, biased limit $x^* = y^* + S^*$. This bias is not erratic or random; it is the predictable, cumulative effect of the perturbation terms $S^* = \sum_{k=0}^{\infty} \beta_k v_k$, which are themselves chosen to promote desirable solution structures (e.g., lower total variation).

In essence, S-BSD-LS sidesteps the trap of semi-convergence by consciously abandoning the pursuit of the noisy minimizer. It does not ask, “How can we get to a flawed target faster?” but rather, “Can we define a more stable and structurally meaningful target?”. By converging to a biased yet well-structured solution, the algorithm achieves stability instead of being affected by noise, thus making it a robust tool for noisy inverse problems. The bias in the final solution is not a flaw but a designed feature to achieve robustness.

**4.1.2. Why $x^*$ is biased.** For $\bar{x}$ to be a true minimizer of $\frac{1}{2}\|b^\delta - Ax\|^2$, it must satisfy the first-order optimality condition $A^T(b^\delta - A\bar{x}) = 0$. Let us evaluate this condition for our limit point $x^* = y^* + S^*$, that is,

$$A^T(b^\delta - Ax^*) = A^T(b^\delta - A(y^* + S^*)) = A^T(b^\delta - Ay^*) - A^T AS^*.$$

As established, $y^*$ is the minimizer of the least-squares problem $\frac{1}{2}\|b^\delta - Ay\|^2$, which means that $A^T(b^\delta - Ay^*) = 0$. Substituting this into the expression for $A^T(b^\delta - Ax^*)$ gives

$$A^T(b^\delta - Ax^*) = -A^T AS^*. \tag{4.6}$$

For $x^*$ to be a true minimizer of $\frac{1}{2}\|b^\delta - Ax\|^2$, we would require that $A^T AS^* = 0$. This condition generally holds if and only if $S^*$ lies in the null space of $A^T A$, which is equivalent to $S^* \in \text{Null}(A)$. However, since $S^* = \sum_{i=0}^{\infty} \beta_i v_i$ represents a general sum of potentially nonzero vectors $v_i$, scaled by nonzero coefficients $\beta_i$, it is highly unlikely that $S^*$ lies in the null space of $A$.

Therefore, the limit point $x^*$ of the perturbed steepest descent algorithm is generally not a true minimizer of $\frac{1}{2}\|b^\delta - Ax\|^2$ if $S^* \notin \text{Null}(A)$. Instead, it is a biased solution, and the bias is directly related to the accumulated explicit perturbation $S^*$. The algorithm reaches a final result (a fixed point), but this result is not the same as the ideal solution we would get if there were no noise and no added perturbation. The difference comes from two sources. First, the method itself includes a built-in change, represented by $\beta_k v_k$, which causes some bias. Second, the data $b^\delta$ include noise, so the least-squares solution $y^*$ is already affected by that noise. In the end, the final output $x^*$ includes both the effect of the bias of the method and the noise in the data.

REMARK 4.5. We note that the optimality condition in (4.6) can be rewritten as the normal equations for a least-squares problem with a modified right-hand side,

$$A^T Ax^* = A^T(b^\delta + AS^*).$$

While this provides an elegant *a posteriori* interpretation of the limit point $x^*$, it is crucial to recognize that this modified problem cannot be formulated and solved directly. The reason is fundamental: the accumulated perturbation $S^* = \sum_{k=0}^{\infty} \beta_k v_k$ is not a predetermined quantity but is rather an emergent product of the entire iterative history generated by Algorithm 2. Specifically, each vector $v_k$ is determined dynamically from the current iterate $x_k$. For instance, in our numerical experiments, $v_k$ is the negative subgradient of the total variation functional at $x_k$. Consequently, $S^*$ is unknown until after the algorithm has converged, making it impossible to construct the modified right-hand side from the outset. This highlights the dynamic and path-dependent nature of our proposed method.

**5. Numerical tests.** In this section, we implement the proposed S-BSD-LS algorithm using total variation (TV) regularization as a concrete example. While the algorithm can be applied with various regularization functionals, TV serves as a well-established benchmark for imaging problems. We present two examples from tomographic image reconstruction, a field focused on generating images from projection data. Our primary goal here is to demonstrate the performance of the algorithm when dealing with noisy data. To maintain uniform experimental conditions, we add $5\%$ white Gaussian noise to all data sets. For all presented figures and tables, the relative error is calculated as $\|x^{\text{true}} - x_k\|/\|x^{\text{true}}\|$, where $x^{\text{true}}$ represents the phantom (exact solution) and $x_k$ is the solution vector at iteration $k$. CPU time refers to the total computational time after 300 iterations for each method. In all experiments, the initial vector is chosen as the zero vector.

We compare the performance of our proposed algorithm, S-BSD-LS (Algorithm 2), with several state-of-the-art methods: SD-LS, CGLS, S-CG-CD, and FISTA. Below, we briefly describe each of these methods and detail the parameters used in our numerical tests. The SD-LS method is a variant of S-BSD-LS that excludes the superiorization term, allowing us to isolate the impact of superiorization, which serves as a baseline comparison. The CGLS method is the standard conjugate gradient method for least-squares.

The S-CG-CD method, a recently introduced algorithm [44], is also included for comparison. For our tests, we specifically use [44, Algorithm 9] as it has shown the lowest error among all algorithms reported in the original publication. The technique employs the negative and normalized subgradient of the total variation as the perturbation term $v_k$, using a coefficient $\beta_k = \omega a^k$, where $a = 0.99$. The parameter $\omega = 40$ was manually selected, as recommended by the authors, to achieve optimal performance, specifically to attain the minimum relative error within the first 300 iterations. For the proposed S-BSD-LS method, we adopt the same form $\beta_k = \omega a^k$, with $a = 0.99$ and $\omega = 1$, and we also use the negative and normalized subgradient of the total variation as the perturbation term $v_k$.

Finally, we include FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [7], a well-known proximal gradient method, to provide a benchmark from a different class of optimization algorithms. The FISTA algorithm is designed to solve optimization problems of the form

$$\psi(x) = f(x) + \phi(x),$$

where $f$ is a smooth convex function and $\phi$ is a convex (usually non-smooth) function. Various versions of this algorithm are presented in [27]. In image reconstruction problems, it is common to set

$$f(x) = \tfrac{1}{2}\|b - Ax\|^2 \qquad \text{and} \qquad \phi(x) = \eta\, TV(x),$$

where $\eta$ is a suitable scalar [27] and $TV(x)$ denotes the total variation (TV) functional.

The challenging part of FISTA is to compute

$$\operatorname{prox}(x) = \arg\min_{y \in C} \left\{ \phi(y) + \tfrac{1}{2}\|y - x\|^2 \right\},$$

where $C$ is a convex set in $\mathbb{R}^n$. If $\phi(x) = \eta\, TV(x)$, then we have

$$\begin{aligned} \operatorname{prox}(x) &= \arg\min_{y \in C} \left\{ \eta\, TV(y) + \tfrac{1}{2}\|y - x\|^2 \right\} \\ &= \arg\min_{y \in C} \left\{ 2\eta\, TV(y) + \|y - x\|^2 \right\}. \end{aligned} \tag{5.1}$$

In [6], a method called FGP is presented for solving the optimization problem (5.1). In fact, the FGP method solves the problem

$$\min_{x \in C} \|x - z\|^2 + 2\eta TV(x), \tag{5.2}$$

where $z$ is an arbitrary vector. It is clear that any algorithm solving problem (5.2) also solves problem (5.1). To solve problem (5.1), FISTA in this section employs 10 iterations of the FGP method. In our implementation, the set $C$ is chosen as the box constraint $0 \leq x_i \leq 1$. The parameter $\eta$ was chosen empirically by a coarse search over several orders of magnitude; we set $\eta = 100$ as it gave the best relative error within 300 iterations.

In both methods, namely S-BSD-LS and S-CG-CD, the vector $v_k$ is selected according to the principles of the gradient descent algorithm, which updates the iterates in the direction opposite to the (sub)gradient of the objective function. In our case, the objective is the total variation functional. We also normalize each vector $v_k$ so that it remains bounded.

A vector $t^x \in \mathbb{R}^n$ is called a *subgradient* of a convex function $g : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ if

$$\langle z - x, t^x \rangle \leq g(z) - g(x) \qquad \text{for all } z \in \mathbb{R}^n.$$

The set of all subgradients of $g$ at $x$ is called the *subdifferential* of $g$ at $x$ and is denoted by $\partial g(x)$. It is well known that $\partial g(x) \neq \emptyset$ for any convex function $g$ and that $\partial g(x) = \{\nabla g(x)\}$ if $g$ is differentiable at $x$; see [4] for a general discussion.

Before proceeding, we recall the definition of the (discrete) total variation. For an image $y \in \mathbb{R}^{G \times H}$, where $y_{g,h}$ denotes the pixel value at position $(g, h)$, the total variation is defined as

$$TV(y) = \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \sqrt{(y_{g+1,h} - y_{g,h})^2 + (y_{g,h+1} - y_{g,h})^2}.$$

This definition is standard; see [28, p. 1567, (4.5)].

The $(g, h)$-component of a subgradient of $TV$ at $y$, denoted by $\ell_y^{g,h}$, is given by

$$\begin{aligned} \ell_y^{g,h} = & \frac{2y_{g,h} - y_{g+1,h} - y_{g,h+1}}{\sqrt{(y_{g+1,h} - y_{g,h})^2 + (y_{g,h+1} - y_{g,h})^2}} \\ & + \frac{y_{g,h} - y_{g-1,h}}{\sqrt{(y_{g,h} - y_{g-1,h})^2 + (y_{g-1,h+1} - y_{g-1,h})^2}} \\ & + \frac{y_{g,h} - y_{g,h-1}}{\sqrt{(y_{g+1,h-1} - y_{g,h+1})^2 + (y_{g,h} - y_{g,h-1})^2}}. \end{aligned}$$

When any of the denominators in the above expression evaluates to zero, the corresponding term is assigned a value of zero to avoid division by zero. This convention ensures the well-definedness and continuity of the subgradient expression.

In both methods, namely S-BSD-LS and S-CG-CD, the vector $v_k$ is selected as a negative subgradient of the TV functional at the current point $x_k$. Specifically, we compute a subgradient $g_k \in \partial TV(x_k)$ using the componentwise formula given above, with the convention that any term with a zero denominator is set to zero. We then set $v_k = -g_k/\|g_k\|$ if $g_k \neq 0$, and $v_k = 0$ otherwise. This ensures that $\|v_k\| \leq 1$, satisfying the boundedness condition. The negative sign indicates that we aim to reduce the value of the TV functional. Note that at the initial point $x_0 = 0$, we have $g_0 = 0$, so $v_0 = 0$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



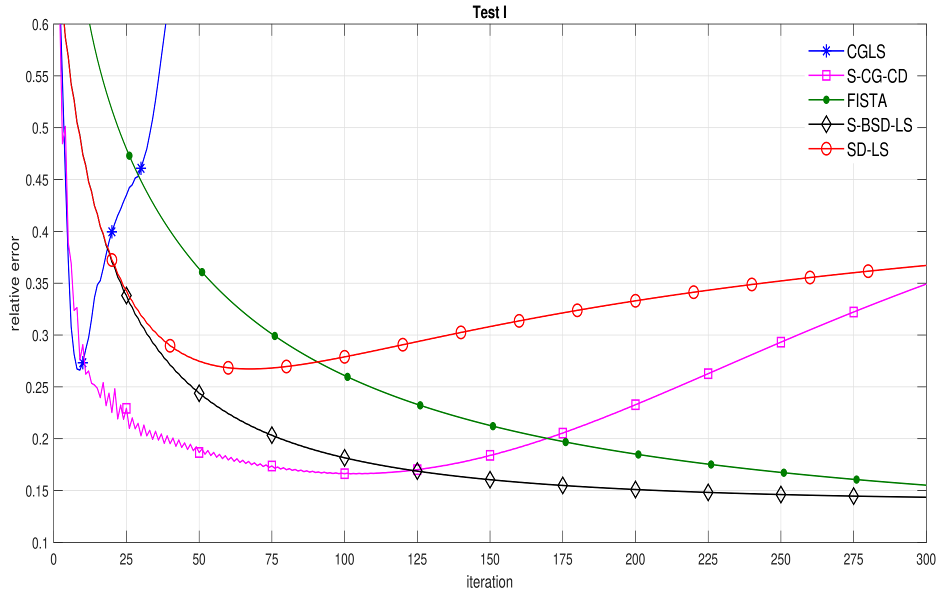


FIG. 5.1. *Histories of the relative errors in S-BSD-LS, SD-LS, CGLS, S-CG-CD, and FISTA. Test 1.*

**5.1. Experimental setup: data generation.** In all iterative methods—S-BSD-LS, SD-LS, CGLS, and S-CG-CD—we use zero as the starting point. For our experiments, we generated all phantom images, system matrices, and right-hand side vectors using the AIR Tools (Matlab-based) package [25]. The Shepp-Logan phantom, a standard synthetic image in medical imaging (especially CT), serves as our ground truth (exact solution). We designed two distinct test cases to evaluate the performance of our algorithm in comparison with existing methods:

**Test 1:** Underdetermined systems. This scenario models limited-angle tomography applications such as breast X-ray tomography and situations with few-projection measurements, where limiting the X-ray dose is critical. In this test, the phantom is discretized into $512 \times 512$ pixels. We collect $179$ projections, evenly distributed between $0$ and $179$ degrees, with $724$ rays per projection. This setup yields a system matrix $A$ of dimension $130320 \times 262144$, resulting in an underdetermined system of equations.

**Test 2:** Overdetermined systems. For this test, we again discretize the phantom into $512 \times 512$ pixels. However, we acquire a larger data-set: $500$ projections evenly distributed between $0$ and $179$ degrees, with $800$ rays per projection. Consequently, the resulting system matrix $A$ has a dimension of $400000 \times 262144$, forming an overdetermined system of equations.

We next analyze the performance of the proposed S-BSD-LS algorithm against established methods: SD-LS, CGLS, S-CG-CD, and FISTA. The evaluation is based on two distinct tomographic reconstruction scenarios, Tests 1 and 2. Performance is assessed using the relative error histories (Figures 5.1 and 5.2) and a summary table (Table 5.1) comparing the CPU time, the minimum relative error, and the iteration at which this minimum error is achieved.

In Test 1, modeling an underdetermined system (limited-angle tomography) with noisy data, CGLS and SD-LS exhibited fast initial convergence, reaching their lowest relative

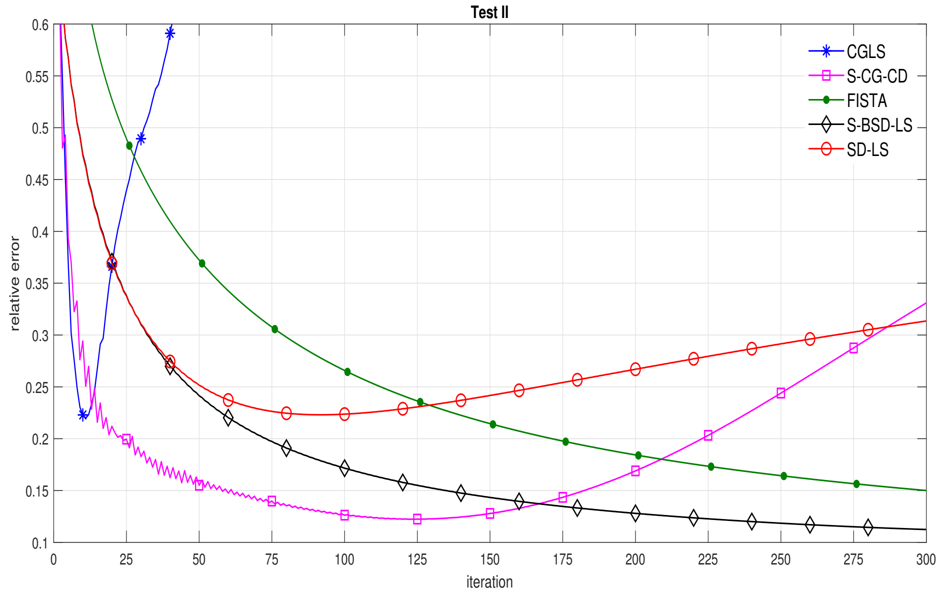


FIG. 5.2. *Histories of the relative errors in S-BSD-LS, SD-LS, CGLS, S-CG-CD, and FISTA. Test 2.*

TABLE 5.1
*Comparison of the methods in terms of the CPU time, the minimum relative error, and the corresponding iteration, Top: Test 1, Bottom: Test 2.*

| Method | CPU Time (s) | Min Rel. Err. | Iter. at Min Rel. Err. |
|---|---|---|---|
| S-BSD-LS | 22.65 | 0.1435 | 300 |
| SD-LS | 19.87 | 0.2673 | 68 |
| CGLS | 20.39 | 0.2662 | 10 |
| S-CG-CD | 301.97 | 0.1661 | 103 |
| FISTA | 344.45 | 0.1552 | 300 |
| S-BSD-LS | 59.97 | 0.1124 | 300 |
| SD-LS | 55.12 | 0.2231 | 93 |
| CGLS | 55.78 | 0.2198 | 12 |
| S-CG-CD | 3675.50 | 0.1223 | 125 |
| FISTA | 611.410 | 0.1500 | 300 |

error early (at iterations 10 and 68, respectively). However, both suffered from severe semi-convergence, with their relative errors increasing significantly after reaching a minimum, as shown in Figure 5.1. This behavior highlights their vulnerability to noise and ill-posedness in underdetermined scenarios.

By contrast, S-BSD-LS maintained a monotonic decrease in the relative error, ultimately achieving the lowest minimum error ($0.1435$) at iteration $300$, with a modest computational cost. This behavior is a direct result of the biased residual update and the TV-based superiorization, which guide the iterates toward structured and stable solutions while avoiding overfitting to noisy data. The method demonstrates strong robustness against semi-convergence while keeping the CPU time ($22.65$ seconds) competitive with SD-LS and CGLS.

S-CG-CD achieved a competitive error ($0.1661$) but at a significantly higher computational cost (over $300$ seconds). Similarly, FISTA reached an error of $0.1552$, but again required

ETNA
Kent State University and
Johann Radon Institute (RICAM)



substantially more time. These results suggest that S-BSD-LS provides a better trade-off between computational efficiency and accuracy in underdetermined, noisy systems.

The performance trends observed in Test 2, characterized by an overdetermined system with noisy data where data redundancy is higher, were broadly consistent with Test 1 but even more pronounced. Both CGLS and SD-LS again converged quickly at first, only to suffer from semi-convergence, which led to a worsening of the error in later iterations (see Figure 5.2). Their minimum errors were moderate ($0.2198$ and $0.2231$), and again occurred early (iterations 12 and 93).

S-BSD-LS, on the other hand, continued to exhibit gradual and stable improvement, ultimately reaching the best overall accuracy with a minimum relative error of $0.1124$ at iteration $300$. This demonstrates the effectiveness of the method even in data-rich contexts. Despite the larger system size, its CPU time ($59.97$ seconds) remained comparable to that of CGLS and SD-LS.

Once again, S-CG-CD and FISTA delivered relatively good accuracy ($0.1223$ and $0.1500$, respectively) but at a significantly higher computational cost ($3675$ and $611$ seconds, respectively). The high computational demand of S-CG-CD in particular limits its practicality, especially when high iteration counts are necessary.

The experiments clearly show that S-BSD-LS consistently delivers the lowest minimum relative error in both underdetermined and overdetermined noisy inverse problems. Compared to traditional methods such as SD-LS and CGLS, it exhibits greater robustness to noise, suppresses semi-convergence, and maintains a reasonable computational cost. Moreover, when compared to S-CG-CD and FISTA, S-BSD-LS achieves comparable or superior accuracy with much lower CPU time, making it an effective and efficient method for stable reconstruction in ill-posed settings.

**5.2. Additional experiments: superiorized SD-LS with exact residual update.** To further analyze the respective effects of superiorization and residual bias, Algorithm 3 introduces the superiorized steepest descent algorithm with exact residual update, denoted by S-SD-LS. Using [2, Lemma 5, Proposition 4, and Theorem 3], one can easily obtain that Algorithm 1 is bounded perturbation resilient (BPR), and therefore, the sequence $\{x_k\}$ generated by Algorithm 3 (S-SD-LS) converges to a least-squares solution of the system (1.1). Unlike S-BSD-LS, this algorithm preserves the true residual at each iteration and thus requires one additional matrix-vector multiplication (compare Step 12 of Algorithm 3 with Algorithm 2). In fact, the most time-consuming part of Algorithms 1, 2, and 3 is the matrix-vector multiplication. Algorithms 1 and 2 each require two matrix-vector multiplications, whereas Algorithm 3 requires three, making it approximately 50% more time-consuming than Algorithm 2. The results for Tests 1 and 2 are summarized in Table 5.2, showing that S-SD-LS achieves nearly the same stability and relative error behavior as S-BSD-LS, while incurring a higher computational cost. This indicates that the stability enhancement primarily originates from the superiorization mechanism and the use of the TV functional for smoothing the solution, whereas the biased residual update contributes mainly to computational efficiency.

**Conclusion.** We proposed the Superiorized Biased Steepest Descent for Least Squares (S-BSD-LS) algorithm, a novel approach that redefines the reconstruction target in noisy inverse problems. By decoupling the residual update from the explicit perturbation and integrating a perturbation step of the form $\beta_k v_k$, the method achieves robust, stable convergence to a structured, biased solution. Numerical experiments in tomographic imaging demonstrate that S-BSD-LS consistently outperforms standard least-squares solvers—including SD-LS and CGLS—as well as advanced regularization methods like S-CG-CD and FISTA, in terms of both accuracy and resistance to semi-convergence. Despite its improved performance, the computational cost per iteration of S-BSD-LS remains comparable to CGLS, requiring

**Algorithm 3** Superiorized Steepest Descent Algorithm for Least-Squares problem (S-SD-LS).

**Given:** Matrix $A$, vector $b^\delta$.
**Initialize:** Choose $x_0 \in \mathbb{R}^n$.
$r_0 \coloneqq b^\delta - Ax_0$.
**for** $k = 0, 1, 2, \ldots$ **do**
  $u_k \coloneqq A^T r_k$.
  **if** $u_k = 0$ **then stop**.
  **end if**
  $s_k \coloneqq Au_k$.
  $\lambda_k \coloneqq \|u_k\|^2/\|s_k\|^2$.
  Compute parameter $\beta_k$ and direction $v_k$.
  Update iterate: $x_{k+1} \coloneqq x_k + \lambda_k u_k + \beta_k v_k$.
  Update biased residual: $r_{k+1} \coloneqq r_k - \lambda_k s_k - \beta_k A v_k$.
**end for**

TABLE 5.2
*Comparison of Algorithms 1, 2, and 3 in terms of the CPU time, the minimum relative error, and the corresponding iteration. Top: Test 1; Bottom: Test 2. The new method S-SD-LS (superiorized SD-LS with exact residual update) is included to illustrate the effect of exact versus biased residual computation.*

| Method | CPU Time (s) | Min Rel. Err. | Iter. at Min Rel. Err. |
|---|---|---|---|
| S-BSD-LS | 22.65 | 0.1435 | 300 |
| SD-LS | 19.87 | 0.2673 | 68 |
| S-SD-LS | 36.81 | 0.1601 | 300 |
| S-BSD-LS | 59.97 | 0.1124 | 300 |
| SD-LS | 55.12 | 0.2231 | 93 |
| S-SD-LS | 111.62 | 0.1254 | 300 |

only two matrix-vector multiplications and a single additional function evaluation of the total variation term, whose subgradient and norm can be computed efficiently with minimal computational cost.

Additional experiments with the superiorized SD-LS algorithm (S-SD-LS)—which employs exact residual updates—further confirmed that the superiorization mechanism itself is the principal source of the improved stability and continuous error reduction, while the biased residual update mainly enhances computational efficiency. These findings reinforce the effectiveness and practicality of the proposed S-BSD-LS method as a reliable and computationally efficient alternative for stable reconstructions in ill-posed, noisy scenarios.

**Acknowledgments.** We sincerely thank the anonymous referees for their valuable comments and insightful suggestions, which have significantly improved the clarity and quality of this manuscript.

REFERENCES

[1] M. ABBASI, M. AHMADINIA, AND A. AHMADINIA, *A new step size selection strategy for the superiorization methodology using subgradient vectors and its application for solving convex constrained optimization problems*, IMA J. Numer. Anal., 44 (2024), pp. 2997–3027.

[2] M. ABBASI AND T. NIKAZAD, *Bounded perturbations resilient iterative methods for linear systems and least squares problems: operator-based approaches, analysis, and performance evaluation*, BIT Numer. Math., 64 (2024), Paper No. 15, 29 pages.

[3] F. BAUER AND M. A. LUKAS, *Comparing parameter choice methods for regularization of ill-posed problems*, Math. Comput. Simulation, 81 (2011), pp. 1795–1841.
[4] H. H. BAUSCHKE AND J. M. BORWEIN, *On projection algorithms for solving convex feasibility problems*, SIAM Rev., 38 (1996), pp. 367–426.
[5] A. BECK, *First-Order Methods in Optimization*, SIAM, Philadelphia, 2017.
[6] A. BECK AND M. TEBOULLE, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.
[7] ———, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.
[8] M. BERTERO AND P. BOCCACCI, *Introduction to Inverse Problems in Imaging*, IOP Publishing, Bristol, 1998.
[9] D. BUTNARIU, R. DAVIDI, G. T. HERMAN, AND I. G. KAZANTSEV, *Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems*, Sel. Topics Signal Proc., 1 (2007), pp. 540–547.
[10] D. BUTNARIU, S. REICH, AND A. J. ZASLAVSKI, *Convergence to fixed points of inexact orbits of Bregman-monotone and of nonexpansive operators in Banach spaces*, in Fixed Point Theory and Its Applications, H. Fetter Nathansky, B. Gamboa de Buen, K. Goebel, W. A. Kirk, and B Sims, eds., Yokohama Publ., Yokohama, 2006, pp. 11–32.
[11] ———, *Stable convergence theorems for infinite products and powers of nonexpansive mappings*, Numer. Funct. Anal. Optim., 29 (2008), pp. 304–323.
[12] A. CEGIELSKI, *Iterative Methods for Fixed Point Problems in Hilbert Spaces*, Springer, Heidelberg, 2012.
[13] Y. CENSOR, *Superiorization and perturbation resilience of algorithms: a bibliography compiled and continuously updated*, Website.
http://math.haifa.ac.il/yair/bib-superiorization-censor.html#top
[14] ———, *Weak and strong superiorization: between feasibility-seeking and minimization*, An. Ştiinţ. Univ. "Ovidius" Constanţa Ser. Mat., 23 (2015), pp. 41–54.
[15] ———, *Can linear superiorization be useful for linear optimization problems?*, Inverse Problems, 33 (2017), Paper No. 044006, 22 pages.
[16] ———, *Superiorization: The asymmetric roles of feasibility-seeking and objective function reduction applied set-valued analysis and optimization*, Appl. Set-Valued Anal. Optim., 5 (2023), pp. 325–346.
[17] Y. CENSOR, R. DAVIDI, AND G. T. HERMAN, *Perturbation resilience and superiorization of iterative algorithms*, Inverse Problems, 26 (2010), Paper No. 065008, 12 pages.
[18] Y. CENSOR, R. DAVIDI, G. T. HERMAN, R. W. SCHULTE, AND L. TETRUASHVILI, *Projected subgradient minimization versus superiorization*, J. Optim. Theory Appl., 160 (2014), pp. 730–747.
[19] S. S. CHEN, D. L. DONOHO, AND M. A. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM Rev., 43 (2001), pp. 129–159.
[20] J. A. FESSLER AND A. O. HERO, *Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms*, IEEE Trans. Imag Proc., 10 (1995), pp. 1417–1429.
[21] A. GIBALI AND M. HALTMEIER, *Superiorized regularization of inverse problems*, J. Appl. Numer. Opti., 2 (2020), pp. 63–70.
[22] M. HALTMEIER, A. LEITÃO, AND O. SCHERZER, *Kaczmarz methods for regularizing nonlinear ill-posed equations. I. Convergence analysis*, Inverse Probl. Imaging, 1 (2007), pp. 289–298.
[23] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.
[24] ———, *The L-curve and its use in the numerical treatment of inverse problems* in Computational Inverse Problems in Electrocardiology, P. Johnston, ed., WIT Press, Southampton, 2001, pp. 119–142.
[25] P. C. HANSEN AND J. S. JØRGENSEN, *AIR Tools II: algebraic iterative reconstruction methods, improved implementation*, Numer. Algorithms, 79 (2018), pp. 107–137.
[26] P. C. HANSEN, J. S. JØRGENSEN, AND P. W. RASMUSSEN, *Stopping rules for algebraic iterative reconstruction methods in computed tomography*, in 2021 21st International Conference on Computational Science and Its Applications (ICCSA), IEEE Conference Proceedings, Los Alamitos, 2021, pp. 60–70.
[27] E. S. HELOU, M. V. ZIBETTI, AND G. T. HERMAN, *Fast proximal gradient methods for nonsmooth convex optimization for tomographic image reconstruction*, Sensing Imag., 21 (2020), Paper No. 45, 31 pages.
[28] E. S. HELOU NETO AND A. R. DE PIERRO, *Incremental subgradients for constrained convex optimization: a unified framework and new methods*, SIAM J. Optim., 20 (2009), pp. 1547–1572.
[29] G. T. HERMAN, E. GARDUÑO, R. DAVIDI, AND Y. CENSOR, *Superiorization: an optimization heuristic for medical physics*, Medical Phys., 39 (2012), pp. 5532–5546.
[30] I. HNĚTYNKOVÁ, M. PLEŠINGER, AND Z. STRAKOŠ, *The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data*, BIT Numer. Math., 49 (2009), pp. 669–696.
[31] R. H. HUESMAN, *A new fast algorithm for the evaluation of regions of interest and statistical uncertainty computed tomography*, Phys. Med. Biol., 29 (1984), pp. 542–552.
[32] W. JIN, Y. CENSOR, AND M. JIANG, *Bounded perturbation resilience of projected scaled gradient methods*, Comput. Optim. Appl., 63 (2016), pp. 365–392.

[33] S. LUO AND T. ZHOU, *Superiorization of EM algorithm and its application in Single-Photon Emission Computed Tomography (SPECT)*, Inverse Probl. Imaging, 8 (2014), pp. 223–246.
[34] F. NATTERER, *The Mathematics of Computerized Tomography*, SIAM, Philadelphia, 2001.
[35] T. NIKAZAD AND M. ABBASI, *Perturbation-resilient iterative methods with an infinite pool of mappings*, SIAM J. Numer. Anal., 53 (2015), pp. 390–404.
[36] ———, *A unified treatment of some perturbed fixed point iterative methods with an infinite pool of operators*, Inverse Problems, 33 (2017), Paper No. 044002, 27 pages.
[37] T. NIKAZAD, M. ABBASI, AND T. ELFVING, *Error minimizing relaxation strategies in Landweber and Kaczmarz type iterations*, J. Inverse Ill-Posed Probl., 25 (2017), pp. 35–56.
[38] T. NIKAZAD, M. KARIMPOUR, AND M. ABBASI, *Notes on flexible sequential block iterative methods*, Comput. Math. Appl., 76 (2018), pp. 1321–1332.
[39] S. N. PENFOLD, R. W. SCHULTE, Y. CENSOR, AND A. B. ROSENFELD, *Total variation superiorization schemes in proton computed tomography image reconstruction*, Medical Phys., 37 (2010), pp. 5887–5895.
[40] D. REEM AND A. DE PIERRO, *A new convergence analysis and perturbation resilience of some accelerated proximal forward-backward algorithms with errors*, Inverse Problems, 33 (2017), Paper No. 044001, 28 pages.
[41] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[42] R. TIBSHIRANI, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[43] C. R. VOGEL, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
[44] M. V. ZIBETTI, C. LIN, AND G. T. HERMAN, *Total variation superiorized conjugate gradient method for image reconstruction*, Inverse Problems, 34 (2018), Paper No. 034001, 26 pages.