International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 10, Issue 3, pp 145-155, March-2026
https://doi.org/10.47001/IRJIET/2026.103020

Comparison between the Classical Gradient Algorithm and Nesterov's Accelerated Algorithm under an Inexact Oracle *(δ,L)*

[1]Abdulrahman Al-Younes, [2]Iyad Al-Hammada

[1]Postgraduate Student (Master's) in the Department of Mathematics, Faculty of Sciences, University of Aleppo, Syria
[2]Lecturer Doctor in the Department of Mathematics, Faculty of Sciences, University of Aleppo, Syria

***Abstract* - This study investigates the impact of an inexact oracle (δ, L) on the convergence of the classical gradient algorithm and Nesterov's accelerated algorithm for smooth convex optimization problems[1]. We show that the classical gradient method maintains a convergence rate of $O\left(\frac{1}{k}\right)$ with an accuracy floor determined by (δ), without error accumulation[2]. In contrast, Nesterov's algorithm achieves an accelerated rate of $O\left(\frac{1}{k^2}\right)$ at the expense of an error accumulation of order $O(k\delta)$[1, 2]. Through a numerical example, we confirm that the acceleration provides an advantage in the early iterations, while the classical method exhibits greater stability as the error increases, demonstrating that the choice of algorithm is directly linked to the quality of the oracle.**

Keywords: First-order methods, inexact oracle (δ, L), convex optimization, Lipschitz constant, classical gradient method, Nesterov's accelerated method, error accumulation, convergence rate.

I. Introduction

Convex optimization constitutes a fundamental pillar in applied mathematics, providing the theoretical framework for modeling a wide spectrum of problems in engineering, economics, data science, and machine learning[4,5]. With the growth of large-scale numerical applications, first-order methods have emerged as effective tools due to their reliance solely on first-order information, their computational simplicity, and practical efficiency[5].

In the ideal case, accurate information about the function and its gradient is assumed to be available, achieving the well-known convergence rates of $O\left(\frac{1}{k}\right)$ for the classical gradient algorithm and $O\left(\frac{1}{k^2}\right)$ for Nesterov's accelerated algorithm[2,5]. However, this assumption rarely holds in practice, as calculations are affected by measurement noise and numerical rounding errors, necessitating dealing with approximate information.

Within this context, the concept of the inexact oracle (δ, L) emerged as a generalization of the exact oracle $(0, L)$, where a controlled error δ is allowed while maintaining the convexity of the function and the Lipschitz condition for its gradient with constant L[1]. This framework has enabled the study of first-order methods in environments characterized by uncertainty and the analysis of how error enters into convergence inequalities[1,7].

This study addresses a comparison between the classical gradient algorithm and Nesterov's accelerated algorithm under a oracle (δ, L), focusing on the structure of error accumulation and the convergence rate. We show that the classical algorithm maintains its stability with an accuracy floor determined by δ, while Nesterov's accelerated algorithm achieves faster convergence in the early stages at the cost of a gradual amplification of the error effect[1,2].

The general convex optimization problem is formulated as follows:

$$\min_{x \in Q} f(x) \quad ; \quad f: Q \to R \quad (1.1)$$

where f is a convex function, and Q is a closed convex set in a finite-dimensional space[4]. We assume that problem (1.1) is solvable, and we denote its optimal solution by x_*.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

First-order methods for solving (1.1) rely on information from the function's gradient, the most famous of which is the classical gradient algorithm, which updates the sequence according to:

$$x_{k+1} = x_k - h_k \nabla f(x_k), \quad k = 0,1,2, \ldots$$

where $x_* \in Q$ is a starting point, and h_$k > 0$ is a step size[7]. This update assumes the availability of the true gradient $\nabla f(x)$ accurately.

In practical applications, accurate values of $f(x)$ or $\nabla f(x)$ may not be available due to measurement noise or the use of internal numerical approximations to compute the gradient. Therefore, accurate information is replaced by a first-order oracle that provides approximate values of the function and its gradient within a controlled error bound[1].

The main question thus arises: What is the impact of the inexact oracle(δ, L) on the convergence of the classical gradient algorithm and Nesterov's accelerated algorithm in optimization, and how does the trade-off between speed and stability influence the choice of the appropriate algorithm for the available computational environment?

II. Basic Definitions

Definition 2.1: (The Class $F_L^{1,1}(Q)$)[4]

Let $Q \in \mathbb{R}^n$be a convex and closed set. The class $F_L^{1,1}(Q)$ is defined as the class of convex differentiable functions on Q whose gradient is continuous and satisfies the Lipschitz condition with constant L>0; i.e.:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2 \ , \ \forall x, y \in Q. \quad (2.1)$$

This condition is equivalent to the following inequality:

$$0 \leq f(x) - (f(y) + \langle \nabla f(y), x - y \rangle) \leq \frac{L}{2}\|x - y\|_2^2 \ , \forall x, y \in Q \ \ (2.2) \ .$$

Definition 2.2: (Exact Oracle$(0, L)$) [1]

Let $f \in F_L^{1,1}(Q)$ The function f is said to be equipped with an exact oracle $(0, L)$ if at every point $y \in Q$, a pair of values$(f_L(y), g_L(y)) \in \mathbb{R} \times \mathbb{R}^n$ can be determined such that they satisfy the two relations:$f_L(y) = f(y)$, $g_L(y) = \nabla f(y)$and satisfy the following inequality:

$$0 \leq f(x) - (f_L(y) + \langle g_L(y), x - y \rangle) \leq \frac{L}{2}\|x - y\|_2^2, \forall x \in Q \ (2.3)$$

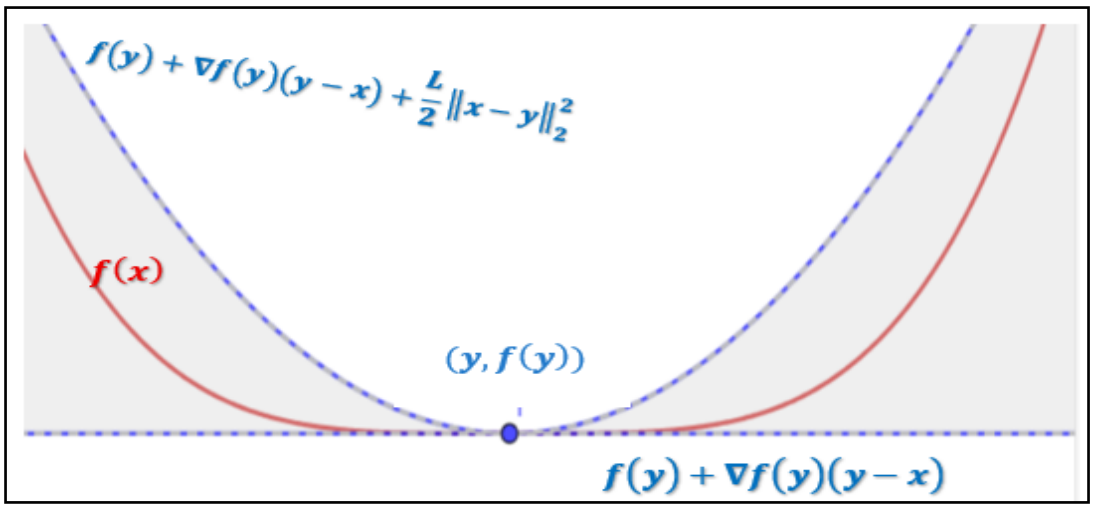

Figure 2.1: Illustration of Inequality (2.3)

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 3, pp 145-155, March-2026
https://doi.org/10.47001/IRJIET/2026.103020

Definition 2.3: (Inexact Oracle(δ, L)) [1]

Let $f \in F_L^{1,1}(Q)$ The function f is said to be equipped with an oracle (δ, L), where δ is called the oracle accuracy, if at every point $y \in Q$ a pair of values $\left(f_{\delta,L}(y), g_{\delta,L}(y)\right) \in \mathbb{R} \times \mathbb{R}^n$ can be determined that satisfy the following inequality:

$$0 \leq f(x) - (f_{\delta,L}(y) + \langle g_{\delta,L}(y), x - y\rangle) \leq \frac{L}{2}\|x - y\|_2^2 + \delta, \forall x \in Q \quad (2.4)$$

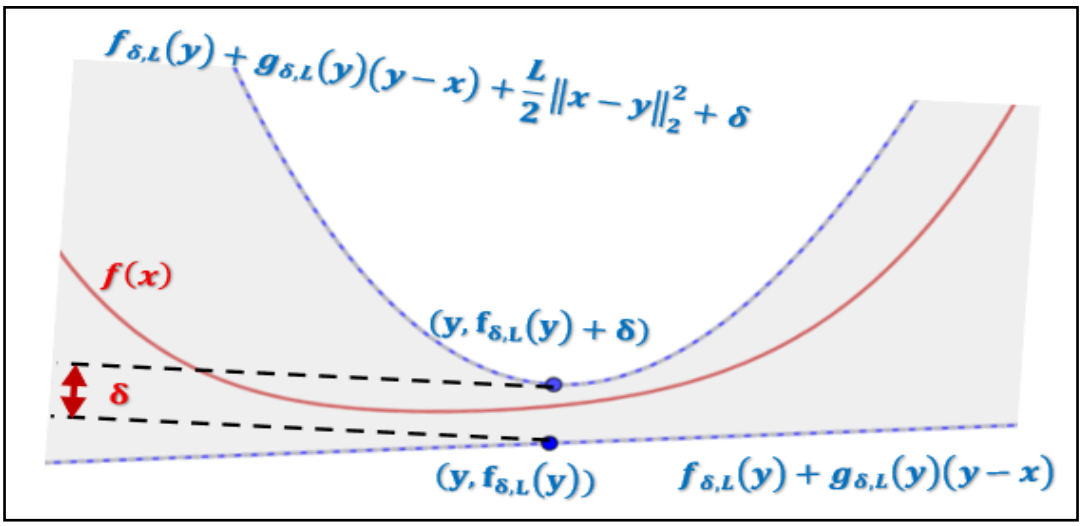

Figure 2.2: Illustration of Inequality (2.4)

Example 2.1: (Computations at Shifted Points) [1]

Let $f \in F_M^{1,1}(Q)$, and suppose that the function f is equipped with an oracle that provides, at every point $y \in Q$, the exact values of the function f and its gradient ∇f, but through an indirect algorithm. Specifically, these values are computed at a shifted point $\hat{y}$different from y. We aim to show the possibility of transforming this type of oracle into an oracle (δ, L), by determining the appropriate values for the constantδ and the Lipschitz constant L as follows:

$$\delta = M\|y - \hat{y}\|_2^2, \quad L = 2M$$

Since f is a convex function, it satisfies the following condition:

$$f(x) \geq f(\hat{y}) + \langle \nabla f(\hat{y}), x - \hat{y}\rangle \quad , \quad \forall x \in Q$$

We note that:$x - \hat{y} = (x - y) + (y - \hat{y})$

Therefore, the right-hand side can be rewritten as:

$$f(x) \geq f(\hat{y}) + \langle \nabla f(\hat{y}), y - \hat{y}\rangle + \langle \nabla f(\hat{y}), x - y\rangle$$

This defines the inexact oracle at point y as:

$$f_{\delta,L}(y) := f(\hat{y}) + \langle \nabla f(\hat{y}), y - \hat{y}\rangle \quad , \quad g_{\delta,L}(y) := \nabla f(\hat{y})$$

With this algorithm, the lower inequality in relation (2.4) is satisfied. On the other hand, since ∇fsatisfies the Lipschitz condition with constant M, according to (2.1) we have thefollowing:

$$f(x) \leq f(\hat{y}) + \langle \nabla f(\hat{y}), x - \hat{y}\rangle + \frac{M}{2}\|x - \hat{y}\|_2^2$$

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

Consequently:

$$f(x) \leq f(\hat{y}) + \langle \nabla f(\hat{y}), y - \hat{y} \rangle + \langle \nabla f(\hat{y}), x - y \rangle + \frac{M}{2}\|x - \hat{y}\|_2^2$$

Given that $\|.\|_2^2$is a convex function, we can estimate the last term using the following property:

$$\|x - \hat{y}\|_2^2 = \left\|\frac{1}{2}\big(2(x-y)\big) + \frac{1}{2}\big(2(y-\hat{y})\big)\right\|_2^2 \leq \frac{1}{2}\|2(x-y)\|_2^2 + \frac{1}{2}\|2(y-\hat{y})\|_2^2$$
$$= 2\|y - \hat{y}\|_2^2 + 2\|x - y\|_2^2$$

Thus, we obtain:$f(x) \leq f_{\delta,L}(y) + \langle g_{\delta,L}(y), x - y \rangle + M\|x - y\|_2^2 + M\|y - \hat{y}\|_2^2$

This demonstrates that the choicesL = 2M and $\delta = M\|y - \hat{y}\|_2^2$satisfy the definition of the oracle(δ, L). This example illustrates that the inexact oracle is not an exceptional case but arises naturally in the presence of computational errors. It also shows that the resulting error can be mathematically controlled within the (δ, L) framework, allowing the study of its impact on the convergence of first-order methods.

III. Classical Gradient Algorithm with Oracle $(\delta, L)$for Smooth Convex Functions

Based on the definition of the oracle(δ, L), the classical gradient algorithm can be adapted to fit this inexact framework[1]. The basic idea relies on replacing the true gradient $\nabla f(x_k)$ with the approximate estimate provided by the oracle $g_{\delta,l}(x_k)$.

The algorithm generates a sequence of points within the convex set Q according to a projection rule that ensures staying within it, while using the constant L to control the step size[2,4].

Algorithm 1.3

Require: initial point $\boldsymbol{x_0 \in Q}$ and $\boldsymbol{L} > 0$.

1: $\boldsymbol{for\ k = 0, 1, \ldots.\ do}$

2: $\boldsymbol{Obtain(f_{\delta,L}(x_k), g_{\delta,L}(x_k))}$

3: $\boldsymbol{Calculat\ x_{k+1} = T_L\left(x_k, g_{\delta,L}(x_k)\right) = \underset{x \in Q}{argmin}\left\{\langle g_{\delta,L}(x_k), x - x_k\rangle + \frac{L}{2}\|x - x_k\|_2^2\right\}.}$

4: $\boldsymbol{end\ for}$

Theorem 3.1: (Error Stability and Cumulative Gap Estimation)[1]

Let Q be a closed convex set and $f: Q \to \mathbb{R}$ a convex function equipped with an oracle (δ, L). When applying the classical gradient algorithm, the following inequality holds:

$$\sum_{i=1}^{k-1}(f(x_{i+1}) - f(x_*)) \leq \frac{L}{2}\|x_0 - x_*\|_2^2 + k\delta \quad , \forall k \geq 1 \quad (3.1)$$

This theorem represents the structural foundation of the convergence analysis. The right-hand side consists of a geometric constant term that depends on the initial distance to the optimal solution, reflecting the nature of the problem itself, and a linear term in k resulting from the oracle accuracyδ.

The significance of this theorem lies in the fact that the error does not enter the analysis through amplification coefficients or uncontrolled accumulation, but rather appears linearly and in a controllable manner. This indicates that the algorithm maintains its stability even in the presence of approximate information[1].

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

Theorem 3.2: (Convergence of the Averaged Sequence and Final Accuracy Bound)[1]

Let Q be a closed convex set and $f: Q \to \mathbb{R}$ a convex function equipped with an oracle (δ, L). Suppose there exists a constant $R > 0$ such that $\|x_0 - x_*\|_2 \leq R$. Then the sequence $\{y_k\}_{k\geq 1}$ defined by $y_k = \frac{1}{k}\sum_{i=1}^{k} x_i$ and generated by the classical gradient algorithm satisfies the following inequality:

$$f(y_k) - f(x_*) \leq \frac{LR^2}{k} + \delta = O\left(\frac{1}{k}\right) + \delta \ , \quad \forall k \geq 1 \quad (3.2)$$

This result demonstrates that transitioning to the arithmetic mean allows extracting an explicit convergence rate. The term $\frac{LR^2}{k}$ represents the algorithm's behavior in the exact case, while δ appears as a convergence floor. Consequently, the $O\left(\frac{1}{k}\right)$ convergence rate is preserved, and the oracle's influence does not alter the fundamental structure of the rate. The final attainable bound is determined by the quality of the available information[1].

Corollary: (Iteration Complexity for Achieving ε-Accuracy)[1]

To achieve $f(y_k) - f(x_*) \leq \varepsilon$, it suffices that $\frac{LR^2}{k} + \delta \leq \varepsilon$. Assuming $\delta \leq \varepsilon$, we obtain

$$k \geq \frac{LR^2}{(\varepsilon - \delta)}$$

This relationship reveals a direct interconnection between the required accuracy ε, the oracle accuracy δ, and the number of iterations k. When ε approaches δ, the required number of iterations becomes large. Moreover, it is theoretically impossible to achieve accuracy better than δ.

Remark: (Final Accuracy Floor)[1]

From relation (3.1), we conclude that: $\lim_{k\to\infty} sup\big(f(y_k) - f(x_*)\big) \leq \delta$

This result confirms that the errors resulting from the inexact oracle do not accumulate explosively, but rather remain bounded within a constant limit. The algorithm maintains its convergence rate, exhibits no error amplification, and remains stable.

IV. Fast Gradient Algorithm with Oracle(δ, L)

Let $\{\alpha_k\}_{k=0}^{\infty}$ be a sequence of real numbers satisfying $\alpha_0 \in (0,1]$, and define the cumulative sum: $A_k := \sum_{i=0}^{k} \alpha_i \quad , \quad k \geq 0$

such that the condition $\alpha_k^2 \leq A_k$ holds.

We also define the mixing coefficients: $\tau_k = \frac{\alpha_{k+1}}{A_{k+1}}$

Let $d(x)$ be an auxiliary convex function on Q, and define the starting point:

$$x_0 = argmin_{x\in Q} d(x)$$

We define the constrained gradient step operator: $T_L(x, g) := P_Q\left(x - \frac{1}{L}g\right)$.

Algorithm 3.1[2]

1: $\boldsymbol{Choose\ \alpha_0 \in (0,1]\ and\ x_0 = argmin_{x\in Q} d(x)}$

2: $\boldsymbol{for\ k = 0, 1, \ldots.\ do}$

3: $\boldsymbol{Obtain\big(f_{\delta,L}(x_k), g_{\delta,L}(x_k)\big)}$

4: $\boldsymbol{Compute\ y_k = T_L(x_k, g_{\delta,L}(x_k))}$

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

5: $\boldsymbol{Compute\ z_k = argmin_{x\in Q}\{Ld(x) + \sum_{i=0}^{k} \alpha_i \langle g_{\delta,L}(x_i), x - x_i\rangle\}}$.

6: $\boldsymbol{Compute\ x_{k+1} = \tau_k z_k + (1-\tau_k) y_k}$

7: $\boldsymbol{end\ for}$

Theorem 2.4: (General Structural Estimate and Error Accumulation)[1,2]

Let $Q \subset \mathbb{R}^n$ be a closed and convex set, and let $f: Q \to \mathbb{R}$ be a smooth convex function equipped with an oracle (δ, L). When applying Nesterov's algorithm with this oracle, for every $k \geq 0$ the following inequality holds: $A_k f(y_k) \leq \Psi_k^* + E_k$, where:

$A_k = \sum_{i=0}^{k} \alpha_i$ are the cumulative gradient coefficients associated with the algorithm steps.

Ψ_k^* is the minimum value of the auxiliary function at step k.

$E_k = \sum_{i=0}^{k} A_i \delta$ is the sum of accumulated errors resulting from using the oracle (δ, L).

y_k is the update point at iteration k.

This inequality reveals that the error in the accelerated algorithm does not enter as a constant term as in the classical gradient algorithm, but rather becomes amplified through cumulative coefficients A_i. That is, the error undergoes a cumulative amplification structure linked to the acceleration mechanism itself. This is where the fundamental difference lies: the acceleration mechanism generates the term $A_k \sim O(k^2)$, which leads to the amplification of the error effect[1,2].

Theorem 4.4: (Convergence Rate and Error Accumulation)[1,2]

Let $Q \subset \mathbb{R}^n$ be a closed and convex set, and let $f: Q \to \mathbb{R}$ be a smooth convex function equipped with an oracle (δ, L). If the sequence y_k is generated by Nesterov's accelerated algorithm, then the following inequality holds:

$$f(y_k) - f(x_*) \leq \frac{2LR^2}{(k+1)^2} + \frac{1}{3}(k+3)\delta$$

where:

$R = \sqrt{2d(x_*)} > 0$ is the approximate radius of the distance between the starting point and the optimal point.

The right-hand side consists of two distinct terms: an accelerated geometric term $\frac{2LR^2}{(k+1)^2}$ of order $O\left(\frac{1}{k^2}\right)$ representing the effect of the acceleration mechanism, and a cumulative linear term $\frac{1}{3}(k+3)\delta$ of order $O(k\delta)$ representing error accumulation.

This result demonstrates that Nesterov's accelerated algorithm achieves a substantial acceleration in the convergence rate compared to the classical gradient algorithm. However, this acceleration is accompanied by a linear amplification mechanism of the approximate error resulting from the oracle.

Consequently, the structural acceleration $O\left(\frac{1}{k^2}\right)$ dominates in the early stages of iterations, while the error term $O(k\delta)$ begins to dominate as the number of iterations increases. Thus, the algorithm maintains its superiority in the early iterations before the error term begins to affect the final achievable accuracy.

V. Numerical Application

Numerical Example 5.1: (Example in $\mathbb{R}^{100}$ under a Constant Additive Error Model in the Gradient)

Consider the constrained smooth convex optimization problem:

$$\min_{x\in Q} f(x) \quad ; \quad f: Q \to R$$

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 3, pp 145-155, March-2026
https://doi.org/10.47001/IRJIET/2026.103020

where:

$$Q = \{x \in \mathbb{R}^{100} \ : \ \|x\|_2 \leq 6\}$$

The objective function is defined as:

$$f(x) = \frac{1}{4}\sum_{j=1}^{100} x_j^4$$

We have:

$$\nabla f(x) = (x_1^3 , \dots \dots , x_{100}^3)$$

The Hessian matrix of the function f is:

$$\nabla^2 f(x) = diag(3x_1^2 , \dots \dots , 3x_{100}^2)$$

Since $3x_j^2 \geq 0$, we have:

$\nabla^2 f(x) \geq 0 \Rightarrow f$ is a convex function

The Lipschitz constant for the gradient ∇fon the set Q is obtained as follows:

$$\left\|\nabla^2 f(x)\right\|_2 = \max_{1 \leq j \leq 100} (3x_j^2)$$

Since $|x_j| \leq \|x\|_2 \leq 6$,we have $x_j^2 \leq 36$and thus $\left\|\nabla^2 f(x)\right\|_2 \leq 108$; that is$L = 108$.

The step size is:$h = \frac{1}{L} = \frac{1}{108}$

Clearly, the optimal solution is:$x_* = 0 \ , \ \ f^* = 0$

To represent the non-ideal case, we assume that the gradient is not $\nabla f(x_k)$but rather an approximation of the form:$g_{\delta,L}(x_k) = \nabla f(x_k) + \xi$

where the error vector is constant across all iterations and satisfies:$\|\xi\|_2 = \delta$

To achieve this, we choose a constant vector:$u = \frac{1}{10}(1,1, \dots ,1) \in \mathbb{R}^{100} \ , \ \|u\|_2 = 1$

Then we define the error vector as:$\xi = \delta$ u

Let us choose the starting point:$x_0 = \frac{1}{\sqrt{2}}(1,1,0, \dots \dots ,0) \in \mathbb{R}^{100}$

Classical Gradient Algorithm under a Constant Additive Error Model in the Gradient[7]

The update formula is:$x_{k+1} = P_Q\left(x_k - \frac{1}{L} g_{\delta,L}(x_k)\right)$

Nesterov's Accelerated Algorithm under a Constant Additive Error Model in the Gradient[2]

- Projected gradient step:$y_k = P_Q\left(x_k - \frac{1}{L} g_{\delta,L}(x_k)\right)$
- Acceleration coefficients:$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \ \ \beta_k = \frac{t_k - 1}{t_{k+1}}$

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

- The update formula is: $x_{k+1} = P_Q(y_k + \beta_k(y_k - y_{k-1}))$

where $t_0 = 1$, $y_{-1} = x_0$.

Both algorithms were implemented programmatically using the Python programming language for each of the cases: $\delta = 0.01$, $\delta = 0.1$, $\delta = 0.9$

with the number of iterations chosen as $k = 150$(a sufficient number of iterations to illustrate the comparison between the three cases for each of the two algorithms).

When $\delta = 0.01$, the graph is:

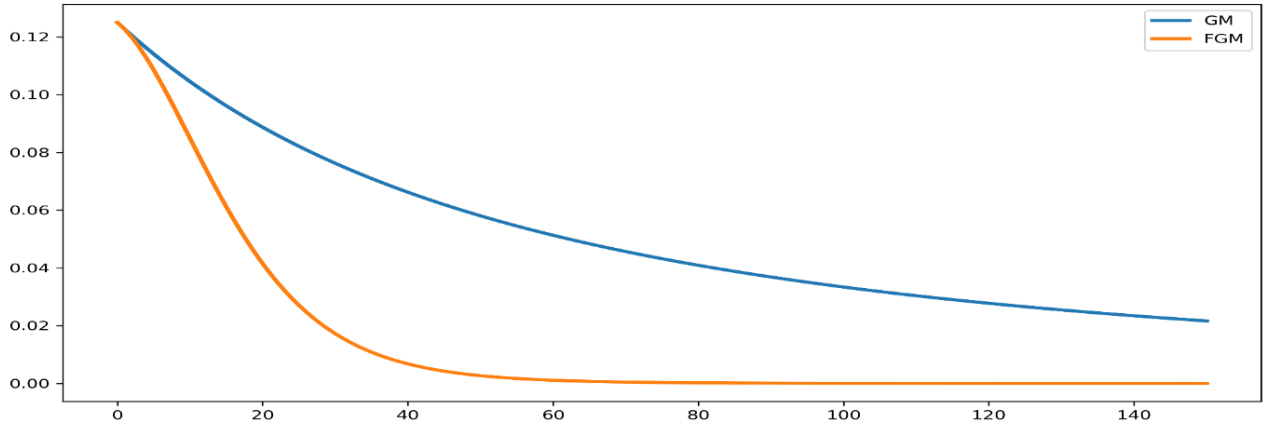

Figure 5.1 Graphical representation of the two algorithms for δ = 0.01 (Python)

In this case, the amount of error added to the gradient is small. The graphical representation shows that both algorithms achieve convergence toward the optimal solution, but the difference appears in the convergence speed. Nesterov's accelerated algorithm (FGM) reaches the vicinity of the solution faster than the classical algorithm (GM). This behavior confirms what we presented theoretically: when δ is small, the impact of the error remains limited, and accumulation does not appear in a way that hinders acceleration, allowing the FGM algorithm to fully benefit from its accelerated structure, while GM remains slower but more stable. This means that FGM outperforms GM in the case of small δ.

When $\delta = 0.1$, the graph is:

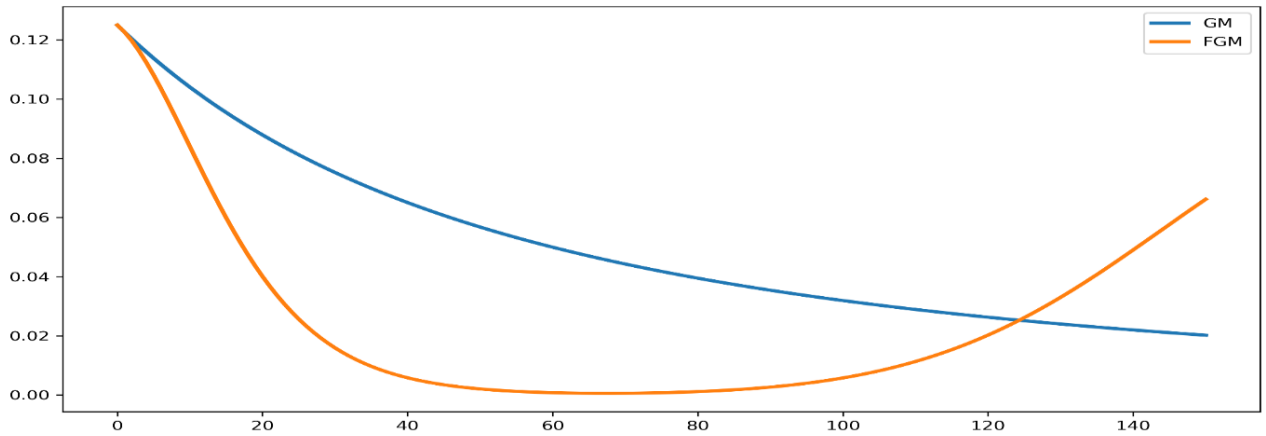

Figure 5.2 Graphical representation of the two algorithms for δ = 0.1 (Python)

When δ is increased to 0.1, the effect of the inexact oracle begins to appear. GM continues its stable behavior, with its curve slowly but steadily approaching the value of f^*. As for FGM, it starts with fast convergence in the early iterations, then its

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 3, pp 145-155, March-2026

https://doi.org/10.47001/IRJIET/2026.103020

trajectory begins to move away from the solution before approaching again. We conclude that when δ is moderate, GM remains stable, while FGM deviates from the solution as a result of error accumulation.

When $\delta = 0.9$, the graph is:

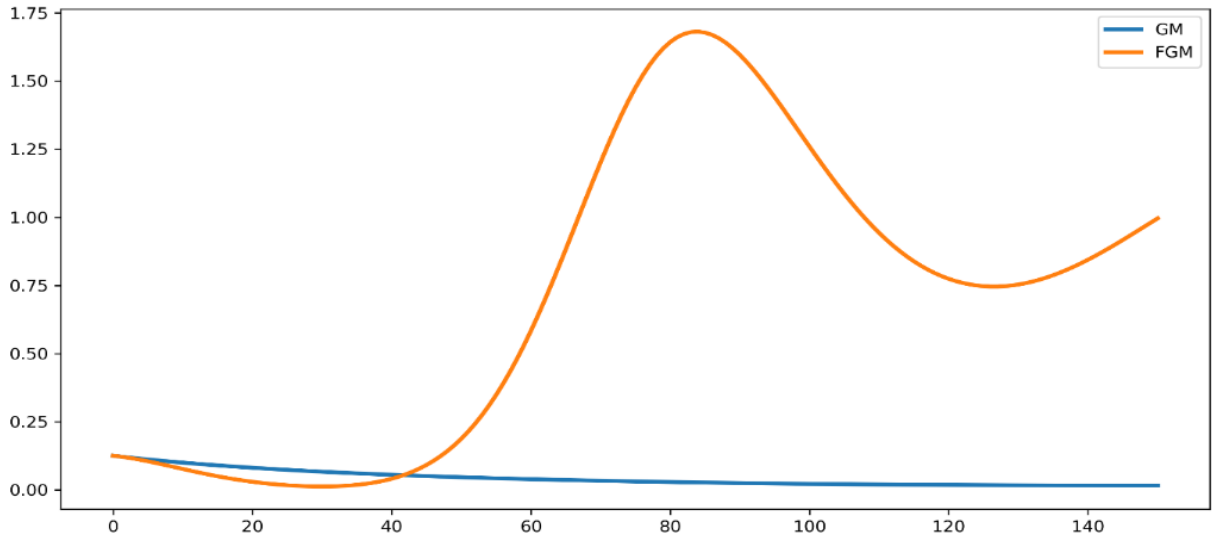

Figure 5.3 Graphical representation of the two algorithms for $\delta = 0.9$ (Python)

With δ increased to 0.9, GM remains coherent, while FGM, after a very fast initial convergence phase, experiences an explosion in the value of $f(x_k) - f^*$leading to divergence from the solution. This divergence reflects the dominance of the error term $O(k\delta)$ over the acceleration term $O\left(\frac{1}{k^2}\right)$. In this case, error accumulation becomes dominant, and FGM fails to maintain any convergence. Here, the difference between the two algorithms becomes evident, leading to the conclusion that when δ is large, GM remains stable, while FGM diverges as a result of error accumulation.

Table 5.1: Comparison of Numerical Results between GM and FGM in the Numerical Example:

δ	GM Behavior	Behavior FGM	Comparative Result
0.01	Regular behavior and stable convergence with gradual decrease in optimality gap	Faster decrease in early stages with clear preservation of acceleration effect	FGM outperforms in terms of speed
0.1	Continues to show good stability and limited sensitivity to error	Acceleration appears at the beginning, then begins to diverge due to error accumulation	FGM is faster than GM initially and less stable after a number of iterations
0.9	Remains relatively more stable despite slow improvement	Becomes more sensitive to error and exhibits clearer oscillations	GM outperforms in terms of stability

Table 1: Numerical comparison of GM and FGM performance under varying δ values

Table 5.2: Analytical comparison of the error effect between classical gradient algorithm and Nesterov's accelerated algorithm:

Comparison Criterion	Classical Gradient Algorithm	Nesterov's Accelerated Algorithm
Initial rate of decrease	Slower	Faster

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 3, pp 145-155, March-2026
https://doi.org/10.47001/IRJIET/2026.103020

Stability with small error	Very good	Very good
Stability with moderate error	Good	Moderate with possibility of divergence
Stability with large error	Relatively better	Relatively weaker
Sensitivity to error accumulation	Limited	Relatively high
Relatively high	When stability is more important than speed	When oracle accuracy is sufficient and faster convergence is desired

Table 2: Analytical comparison of error impact on classical gradient method versus Nesterov's accelerated method

Table 5.3: Theoretical comparison between classical gradient algorithm and Nesterov's accelerated algorithm:

Element	Classical Gradient Algorithm	Nesterov's Accelerated Algorithm
Type of error effect	Appears as a constant or controlled linear additional term	Appears as a clearer error accumulation through acceleration coefficients
Convergence estimate	$f(y_k) - f(x_*) \leq \frac{LR^2}{2k} + \delta$	$f(y_k) - f(x_*) \leq \frac{2LR^2}{(k+1)^2} + \frac{1}{3}(k+3)\delta$
Convergence rate	$O\left(\frac{1}{k}\right)$	$O\left(\frac{1}{k^2}\right)$
Error-related part	δ	$O(k\,\delta)$
Interpretation of effect	Error imposes a final accuracy floor but does not destabilize	Acceleration speeds up convergence but increases the method's sensitivity to error
Theoretical outcome	Slower but more stable	Faster in exact or near-exact cases, and more sensitive in inexact cases

Table 3: Theoretical comparison of error impact between classical gradient and Nesterov's accelerated algorithms

The results presented in the previous tables show that the inexact $oracle\ (\delta, L)$ is not limited in its effect to the theoretical formulation of the optimization problem, but is directly reflected in the convergence behavior of numerical methods. While the classical gradient method maintains stability in the presence of error, Nesterov's accelerated algorithm benefits from its theoretical speed when the oracle accuracy is high, but becomes more sensitive as the error magnitude increases. Consequently, the comparison between the two algorithms is not related solely to convergence speed, but is also influenced by the nature of the oracle and the level of inaccuracy in the available information about the function.

On this basis, using the classical gradient algorithm is an appropriate choice when the δ value is relatively large, due to its greater numerical stability. In contrast, Nesterov's accelerated algorithm remains more effective when the δ value is small. Thus, the two algorithms can be viewed as complementary to each other.

REFERENCES

[1] Devolder, O., Glineur, F., & Nesterov, Y. (2014). First-order methods with inexact oracle: The strongly convex case. *Mathematical Programming*, 147(1), 193–228.

ISSN (online): 2581-3048
Volume 10, Issue 3, pp 145-155, March-2026
https://doi.org/10.47001/IRJIET/2026.103020

[2] Nesterov, Y. (2018). Lectures on convex optimization (2nd ed.). *Springer*.
[3] Lan, G. (2020). First-order and stochastic optimization methods for machine learning. *Springer*.
[4] Boyd, S., & Vandenberghe, L. (2004). Convex optimization. *Cambridge University Press*.
[5] Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, Volume 8, Issue (3-4), 231–357.
[6] Garrigos, G., & Gower, R. M. (2024). Handbook of convergence theorems for (stochastic) gradient methods. *Cornell University*, Volume 3.
[7] Polyak, B. T. (1987). Introduction to optimization. *Optimization Software*.

Citation of this Article:

Abdulrahman Al-Younes, & Iyad Al-Hammada. (2026). Comparison between the Classical Gradient Algorithm and Nesterov's Accelerated Algorithm under an Inexact Oracle *(δ,L)*. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(3), 145-155. Article DOI https://doi.org/10.47001/IRJIET/2026.103020

****

