SOLVANGLE (Journal of Theoretical Insights), Vol. 1, No, 1, June 2025, pp. 7-26
Website: https://klamidas.com/solvangle/

Gradient Descent Convergence: From Convex Optimization to Deep Learning

Mark Laisin & Rosemary U. Adigwe

Abstract

Gradient descent is a fundamental optimization algorithm commonly used in machine learning and deep learning to minimize objective functions. Despite its simplicity, understanding its convergence behaviour in different optimization landscapes remains critical for ensuring efficient and stable training of models. This study examines the theoretical and practical convergence properties of gradient descent in convex, strongly convex, and non-convex scenarios. The aim was to analyse how factors such as function smoothness, step size, and algorithmic variants affect convergence to minima or stationary points. We used both theoretical proofs and numerical experiments to assess performance. Convergence proofs were presented for convex and smooth functions, demonstrating sub-linear convergence for gradient descent, and for strongly convex functions, showing linear convergence. In non-convex settings, we showed that the gradient descent converges to stationary points under standard assumptions, supported by theoretical guarantees. To validate these results, we apply gradient descent, conjugate gradient, and an Adaptive Modified Gradient-Type (AMGT) method to optimize convex and bivariate quadratic functions. Our simulations indicated that while standard gradient descent ensured stable but slower convergence, conjugate gradient methods offered faster descent in convex quadratic problems. AMGT demonstrated improved convergence speed with appropriate tuning but diverged with excessively high learning rates, highlighting the importance of hyperparameter sensitivity. In conclusion, the study confirms that the convergence behaviour of gradient-based methods depends significantly on problem structure and learning rate selection. For convex problems, using fixed learning rates below the inverse of the Lipschitz constant ensures convergence. In non-convex domains, adaptive methods such as Adam and learning rate scheduling can improve performance, especially in deep learning applications. We recommend careful step size tuning and method selection based on problem characteristics to balance convergence speed and stability in practical applications.

Keywords: gradient descent, convergence analysis, convex optimization, non-convex functions, learning rate, conjugate gradient method, adaptive optimization, deep learning

Author Profile

Mark Laisin: Professor of Applied Mathematics

Mark Laisin is a Professor of Applied Mathematics at Chukwuemeka Odumegwu Ojukwu University, Uli, Nigeria. He holds a Ph.D in Mathematics from Nnamdi Azikiwe University, Awka, and has over two decades of experience in teaching, research, and postgraduate supervision. His areas of specialization include combinatorics, optimization theory, stochastic analysis, and computational mathematics.

Prof. Laisin has authored and co-authored over 50 peer-reviewed journal articles and several academic books, including *Explicit Business Mathematics and Combinatorical Techniques for Chessboard Movements*. He serves on editorial boards and is a regular reviewer for academic journals. He has also served as an external examiner for numerous M.Sc. and Ph.D. theses in advanced mathematics, optimization, and financial modeling.

His current research focuses on vector-based modeling of chessboard movements, algorithmic optimization techniques, and convergence analysis in deep learning, especially involving adaptive moment estimation methods.

Email: laisinmark@gmail.com | lm.mark@coou.edu.ng

Papers Co-Authored with Rosemary U. Adigwe

- Gradient Descent Convergence: From Convex Optimization to Deep Learning (Current paper)
- Laisin, M., & Adigwe, R. U. (2025). Implementation and comparative analysis of AMGT method in Maple 24: Convergence performance in optimization problems. *Global Online Journal of Academic Research (GOJAR), 4*(2), 26–40. https://klamidas.com/gojar-v4n1-2025-02/

Academic Research Interest & Output

Research Interests

- Combinatorics and Enumerative Techniques
- Optimization Theory and Mathematical Programming
- Stochastic Processes and Financial Mathematics
- Computational Mathematics and Theoretical Foundations of Machine Learning

Awards & Recognitions

- Exemplary Service Award, COOU Faculty of Physical Sciences Conference (FAPSCON), 2023
- Excellence in Postgraduate Supervision, COOU Postgraduate School Recognition, 2022
- Best Paper Award, World Journal of Innovation Research, for the paper "Three-dimensional Pathway for Disjoined Chess Board with Vector Directives," 2020
- Editorial Service Recognition, COOU Journal of Physical Sciences, for 8+ years of dedicated editorial and peer review service, 2024

I. INTRODUCTION

The gradient descent algorithm has its roots in classical optimization theory, tracing back to the 19th century. The concept of a gradient-based optimization method was formally introduced by Augustin-Louis Cauchy in 1847, who proposed the method of steepest descent as a numerical tool for solving systems of equations, laying the groundwork for modern iterative function minimization. Throughout the 20th century, gradient-based methods saw further advancements in numerical analysis, convex optimization, and control theory. Notably, Marquardt (1963) made significant contributions by introducing the Levenberg-Marquardt algorithm, which greatly enhanced optimization in nonlinear least-squares problems.

Gradient descent gained prominence in the realm of machine learning with the emergence of artificial neural networks. One of its early applications was in the perceptron model devised by Frank Rosenblatt in 1958. However, the

perceptron's limitation to solving linearly separable problems led to a temporary decline in interest. The 1980s saw a resurgence of interest in neural networks with the introduction of the back propagation algorithm by Rumelhart, Hinton, and Williams (1986). This breakthrough utilized stochastic gradient descent (SGD) to efficiently train multi-layer perceptrons, driving significant advancements in supervised learning tasks.

The 2010s marked a period of rapid progress in deep learning, where deep neural networks achieved cutting-edge results in areas like image recognition (Krizhevsky*et al*., 2012), natural language processing (Vaswani *et al*., 2017), and reinforcement learning (Silver et al., 2016). These successes heightened interest in understanding and enhancing the convergence behavior of gradient descent, particularly in the face of the complex, non-convex landscapes prevalent in deep learning.

More recently, Laisin and Adigwe (2025) delved into the convergence behavior of the Adaptive Modified Gradient Technique (AMGT) in their study "Implementation and Comparative Analysis of AMGT Method in Maple 24: Convergence Performance in Optimization Problems." Their analysis offered comparative insights into the sensitivity and efficiency of gradient-based methods across various optimization scenarios. Expanding on this research, Laisin *et al*. (2024) explored the construction of rational polyhedra on an n×n board, introducing structural considerations relevant to integral polyhedral optimization. Their work provides valuable geometric insights applicable to studying constraint structures in optimization problems.In a related study, Laisin *et al*. (2025) investigated boundedness and solution size in rational linear programming and polyhedral optimization. This research lays theoretical foundations that shed light on the behavior of gradient-based methods, especially in constrained optimization scenarios involving rational polyhedra. Furthermore, Laisin and Edike (2025) tackled the construction of simplex linear integer programming problems with application, exploring differentiated solution techniques suitable for discrete optimization challenges. These constructions serve as a link between integer programming formulations and continuous optimization methods like gradient descent.

Collectively, these contributions highlight the evolving significance of gradient descent not only in contemporary machine learning but also in classical and discrete optimization domains. This paper builds on these advancements by examining the convergence properties of gradient descent across convex, polyhedral, and deep learning contexts, with a focus on how structural characteristics—such as rational constraints and problem boundedness—affect the efficacy and efficiency of gradient-based approaches.

II. Preliminaries and Definitions

To address challenges in optimization, researchers have introduced advanced techniques:

- Momentum-based methods: Nesterov (1983) introduced momentum-based methods to accelerate convergence by incorporating previous gradients.
- Adaptive learning rate methods: Kingma and Ba (2014) proposed Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. arxiv.org
- Second-order methods: Amari (1998) developed natural gradient descent, which considers the geometry of the parameter space by using the Fisher information matrix. researchgate.net+2dl.acm.org+2researchgate.net+2

These methods have significantly influenced the study of gradient descent convergence.

Definitions:

Definition 2.1: Convexity and Smoothness

For many optimization problems, particularly in classical machine learning, the loss function is convex. A function $f(\theta)$ is convex if:

$$f(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda f(\theta_1) + (1-\lambda)f(\theta_2), \qquad \forall\lambda \in [0,1].$$

If a function is strongly convex, gradient descent is guaranteed to converge

linearly to the global optimum (Nesterov, 2004).

Definition 2.2: Lipschitz Continuity of Gradients

The gradient $\nabla f(\theta)$ is assumed to be L-Lipschitz continuous, meaning that the function's rate of change is bounded:

$$\| \nabla f(\theta_1) - \nabla f(\theta_2) \| \leq L \| \theta_1 - \theta_2 \|$$

This condition ensures that the optimization process does not behave erratically (Laisin and Adigwe, 2025).

Definition 2.3: Non-Convex Settings and Stationary Points

In deep learning, loss functions are typically non-convex, which means gradient descent is not guaranteed to reach a global minimum. However, it has been

shown that gradient descent still converges to a stationary point (a point where $\nabla f(\theta) = 0$ (Ge *et al*., 2015).

Definition 2.4: Factors Influencing the Study of Gradient Descent Convergence

2.4.1 Advances in Mathematical Optimization**:** Theoretical advancements in convex analysis and optimization algorithms have shaped modern gradient descent studies (Nemirovski & Yudin, 1983).

2.4.2 Computing Power and Scalability:The availability of GPUs and TPUs has allowed for large-scale experiments, influencing research into adaptive optimization methods.

2.4.3 Practical Performance in Deep Learning**:** The empirical success of methods like Adam led to studies on their theoretical properties and limitations (Reddi*et al*., 2018).

Definition 2.5: Circumstances Leading to the Study of Gradient Descent in Deep Learning

Several developments in deep learning motivated the need for studying gradient descent convergence:

2.5.1 Large-Scale Datasets and High-Dimensional Optimization:Modern deep learning models are trained on massive datasets (e.g., Image Net), requiring optimization techniques that scale efficiently.

2.5.2 Non-Convexity and the Challenge of Local Minima:Theoretical studies suggested that neural network loss functions contain many saddle points rather than local minima, making convergence analysis crucial (Choromanska *et al*., 2015).

2.5.3 Batch vs. Stochastic Optimization: Stochastic gradient descent (SGD) became the dominant optimization method, requiring research into stochastic convergence properties (Bottou, 2010).

Definition 2.6:

2.6.1 Convex combination:

Given vectors$x_1, x_2, \ldots . x_n$ in a vector space, a convex combination is any vector of the form:

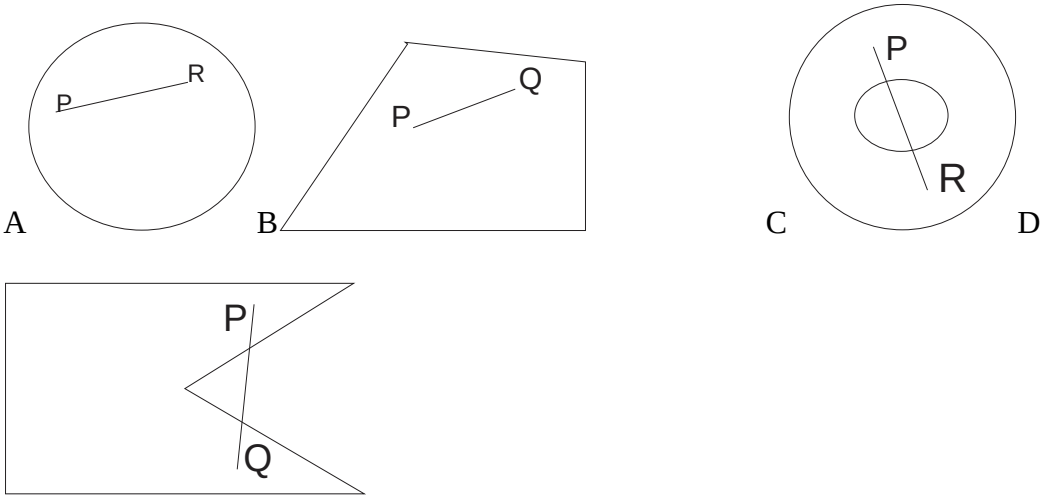
$$\sum_{i=1}^{n} \lambda_i x_i = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$$

where each $\lambda_i \geq 0$, and

$$\sum_{i=1}^{n} \lambda_i = 1$$

2.6.2 Convex Set:

A set is called convex if, for any two points within the set, the line segment connecting them lies entirely within the set. In other words, a set is convex if the convex combination of any two points in the set also belongs to the set.



A and B are convex sets, but C and D are non-convex sets.

2.6.3 Extreme Point of a Convex Set:

A point x in a convex set C is called an extreme point if it cannot be written as a convex combination of two distinct points x_1 and x_2 in C. That is, if

$$x = \lambda x_1 + (1 - \lambda)x_2 \text{for } 0 < \lambda < 1$$

then it must be that $x_1 = x_2 = x$.

Note: Every *extreme point* of a convex set lies on its boundary. However, not every *boundary point* of a convex set is necessarily an extreme point.

2.6.4 Convex Hull:

The *convex hull* of a set X, denoted as conv(X), is the smallest convex set that contains X. Equivalently, it is the set of all convex combinations of points in X.

2.6.5 Convex Function:

A function $f(x)$ is said to be strictly convex if, for any two distinct points x_1 and x_2, and for all $\lambda \in (0,1)$, the following inequality holds:

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2)$$

This means the graph of $f(x)$ lies strictly below the straight line connecting any two points on the graph.However, a function $f(x)$ is strictly concave if$-f(x)$ is strictly convex.

2.6.**6** Convex Polyhedron:

A *convex polyhedron* is the set of all convex combinations of a finite number of points. In other words, it is the convex hull of a finite set of points, and represents a bounded or unbounded convex region formed by these points (Laisin *et al*., 2024).

Theorem 1: A hyper-plane is a convex set.

Proof:
Consider a hyper-plane defined by

$$X = \{x \in R^n \mid c^T x = z\}$$

Let x_1 and x_2 be any two points in the hyper-plane$X$. Then:

$c^T x_1 = z$ and $c^T x_2 = z$

Now, consider any convex combination of x_1 and x_2:

$$x_3 = \lambda x_1 + (1-\lambda)x_2, \qquad \forall\ 0 \le \lambda \le 1$$

Then,

$$c^T x_3 = \lambda c^T x_1 + (1-\lambda)c^T x_2 = \lambda x_1 + (1-\lambda)x_2 = z$$

This implies that $x_3 \in X$. Since any convex combination of points in X also lies in$X$, by definition, the hyper-plane X is a convex set.

QED

Theorem 2:

The intersection of two convex sets is also a convex set.

Proof:
Let X_1 and X_2 be two convex sets, and let

$$X_3 = X_1 \cap X_2$$

Take any two points $x_1, x_2 \in X$. By definition of intersection :

$$x_1, x_2 \in X_1 and\ x_1, x_2 \in X_2$$

Since both X_1 and X_2 are convex, for any $0 \leq \lambda \leq 1$:

$$\lambda x_1 + (1-\lambda)x_2 \in X_1 \; and \; \lambda x_1 + (1-\lambda)x_2 \in X_2$$

Therefore,

$$\lambda x_1 + (1-\lambda)x_2 \in X_1 \cap X_2 = X_3$$

Thus, by definition, X_3 is a convex set.

QED

III. Main Results and Discussions

3.1. Convergence of Gradient Descent for Convex and Smooth Functions:

Gradient descent is a fundamental algorithm in optimization, widely used in both classical convex optimization and modern machine learning, particularly deep learning. When the objective (or loss) function $f(\theta)$ is convex and smooth, and the learning rate α is chosen appropriately, gradient descent can be shown to converge to a global minimum.

Gradient Descent Update Rule:

$$\theta_{t+1} = \theta_t - \alpha \nabla f(\theta_t)$$

If $f(\theta_t)$ is L-smooth; that is, its gradient is Lipschitz continuous with constant L, then for a step size $0 < \alpha \leq \frac{1}{L}$, gradient descent guarantees a monotonic decrease in the objective function:

$$f(\theta_{t+1}) \leq f(\theta_t) - \frac{\alpha}{2} \parallel \nabla f(\theta t) \parallel^2$$

This inequality ensures that each update decreases the function value, thereby guiding the optimization towards a minimizer.

Assumptions:

To rigorously analyze convergence, we assume:

1. Convexity:
   A function $f(\theta)$ is convex if for all $\theta, \theta' \in R^d$:

   $$f(\theta') \geq f(\theta) + \nabla f(\theta) T(\theta' - \theta)$$

2. L-smoothness (Gradient Lipschitz Continuity**)**:

   There exists $L > 0$ such that for all θ, θ':

$$\| \nabla f(\theta) - \nabla f(\theta') \| \leq L \| \theta - \theta' \|$$

Equivalently, for any θ, θ', we have:

$$f(\theta') \leq f(\theta) + \nabla f(\theta) T(\theta' - \theta) + \frac{L}{2} \| \theta' - \theta \|^2$$

Proof of Convergence for Smooth Convex Functions

Using the update rule

$$\theta_{t+1} = \theta_t - \alpha \nabla f(\theta_t)$$

and the smoothness property, we obtain:

$$f(\theta_{t+1}) \leq f(\theta_t) - \frac{\alpha}{2} \| \nabla f(\theta_t) \|^2 + \frac{L}{2} \| \theta' - \theta \|^2$$

$$= f(\theta_t) - \alpha \| \nabla f(\theta_t) \|^2 + \frac{L\alpha^2}{2} \| \nabla f(\theta_t) \|^2$$

$$= f(\theta_t) - \left(\alpha - \frac{L\alpha^2}{2} \right) \| \nabla f(\theta_t) \|^2$$

To ensure that the term $\left(\alpha - \frac{L\alpha^2}{2} \right)$ is positive (hence ensuring descent), it suffices to choose

$0 < \alpha < \frac{L}{2}$. For stability and simplicity, a common choice is $\alpha \leq \frac{1}{L}$, which yields:

$$f(\theta') \leq f(\theta) + \nabla f(\theta)^T (\theta' - \theta) + \frac{L}{2} \| \theta' - \theta \|^2$$

This guarantees a monotonic decrease in the function value at each step.

Convergence for Strongly Convex Functions:

If, in addition to being smooth and convex, the function f is μ-strongly convex (with $\mu > 0$), then gradient descent enjoys a faster, linear convergence rate. A function f is strongly convex if:

$$f(\theta') \geq f(\theta) + \nabla f(\theta)^T (\theta' - \theta) + \frac{\mu}{2} \| \theta' - \theta \|^2$$

Under this condition, we can derive a strong inequality that relates the gradient norm to the sub-optimality:

$$\| \nabla f(\theta_t) \| 2 \geq 2\mu (f(\theta_t) - f(\theta^*))$$

Plugging this into our earlier descent bound:

$$f(\theta_t) \leq f(\theta_t) - \frac{\alpha}{2} \| \nabla f(\theta_t) \|^2$$

$$\leq f(\theta_t) - \alpha\mu(f(\theta_t) - f(\theta^*))$$

$$\leq (1 - \alpha\mu) f(\theta_t) - f(\theta^*)$$

This gives:

$$f(\theta_t) - f(\theta^*) \leq (1 - \alpha\mu) f(\theta_t) - f(\theta^*)$$

Which shows geometric (linear) convergence to the unique minimizer θ^*, provided that

$0 < \alpha \leq \frac{1}{L}$.

Comparative Discussion and Practical Relevance:

The rate of convergence depends crucially on the structure of the function being optimized:

- For general convex and L-smooth functions, gradient descent has a sub-linear convergence rate of $O(1/t)$.
- In case of strongly convex functions, the convergence rate becomes linear, at $O((1 - \alpha\mu)^t)$**,**leading to exponential acceleration (Nesterov, 2004).

This distinction is crucial in the context of large-scale machine learning applications. While convex optimization theory offers robust assurances, the loss surfaces in deep learning are typically non-convex. Nonetheless, these theoretical findings offer valuable insights, particularly in local regions where the loss surface approximates convexity or smoothness (Goodfellow *et al*., 2016).

Furthermore, practical deep learning often benefits from adaptive techniques such as Adam or RMSProp, which adjust the gradient update rules. However, these methods may not possess the same rigorous convergence guarantees as standard gradient descent (Reddi *et al*., 2018).

3.2. Convergence of Gradient Descent for Non-Convex Functions:

In the context of non-convex optimization, gradient descent does not guarantee convergence to a global minimum. However, under mild conditions, it can converge to a local minimum or a stationary point, the points where the gradient vanishes. The core of the analysis lies in demonstrating that the gradient norm, $\| \nabla f(\theta_t) \|$, tends toward zero as the number of iterations increases, signifying convergence to a critical point.

Proof of Convergence to a Stationary Point

To establish this result, we make the following assumptions:

1. L-Smoothness: The function $f(\theta)$ is L-smooth, meaning its gradient is Lipschitz continuous:

$$\| \nabla f(\theta) - \nabla f(\theta') \| \leq L \| \theta - \theta' \|$$

This property leads to the descent lemma:

$$f(\theta_{t+1}) \leq f((\theta_t) + \nabla f(\theta_t)^T(\theta_{t+1} - \theta_t) + \frac{L}{2} \| \theta_{t+1} - \theta_t \|^2$$

2. Gradient Descent Update Rule:

$$\theta_{t+1} = \theta_t - \alpha \nabla f(\theta_t).$$

3. Bounded Below: The function $f(\theta)$ is bounded below. There exists a constant $f^* > -\infty$ such that:

$$f(\theta) \geq f^*$$

Substituting the update rule into the descent lemma gives:

$$f(\theta_{t+1}) \leq f(\theta_t) - \alpha \| \nabla f(\theta_t) \|^2 + \frac{L\alpha^2}{2} \| \nabla f(\theta_t) \|^2$$

Rewriting:

$$f(\theta_{t+1}) \leq f(\theta_t) - (\alpha - \frac{L\alpha^2}{2}) \| \nabla f(\theta_t) \|^2$$

To ensure progress (i.e., a decrease in function value), choose $0 < \alpha \leq \frac{1}{L}$. This guarantees:

$$\alpha - \frac{L\alpha^2}{2} \geq \frac{\alpha^2}{2}$$

Hence:

$$f(\theta_{t+1}) \leq f(\theta_t) - \frac{\alpha}{2} \| \nabla f(\theta_t) \|^2$$

Summing over $t = 0$ to $T - 1$:

$$f(\theta_T) \leq f(\theta_0) - \frac{\alpha}{2} \sum_{0}^{T-1} \| \nabla f(\theta_t) \|^2$$

Using the boundedness of $f(\theta)$:

$$f(\theta_0) - f^* \geq \frac{\alpha}{2} \sum_{0}^{T-1} \| \nabla f(\theta_t) \|^2$$

Dividing both sides by T:

$$\frac{2(f(\theta_0) - f^*)}{\alpha T} \geq \frac{1}{T} \sum_{0}^{T-1} \| \nabla f(\theta_t) \|^2$$

Taking the limit as $T \to \infty$:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{0}^{T-1} \| \nabla f(\theta_t) \|^2 = 0$$

This implies that the average norm of the gradient converges to zero. Therefore, there exists a sequence $\{\theta_t\}$for which $\| \nabla f(\theta_t) \|$ becomes arbitrarily small, indicating convergence to a stationary point (Ghadimi & Lan, 2013).

Comparative Discussion and Practical Relevance:

In non-convex landscapes, gradient descent may converge to a local minimum, a saddle point, or a plateau region (areas where the gradient is close to zero but the function value is not minimal). However, the strict saddle property, where the Hessian at saddle points has at least one negative eigenvalue, allows gradient descent to often escape saddle points. This is especially evident in stochastic settings, where random perturbations in algorithms like stochastic gradient descent (SGD) provide the momentum necessary to bypass such saddle regions (Ge *et al*., 2015).

Quadratic Surfaces and Step Size Influence:

A common form of a quadratic function is given by:

$$F(X, Y) = \alpha X^2 + \beta Y^2 + \gamma XY + \omega$$

This form is frequently used in applications like resource allocation and optimization modeling. The parameters α, β, γ, and ω determine the curvature and orientation of the function's surface. The behaviour of gradient descent on such functions is affected by the step size α, which influences the convergence speed and stability.

Table 3.1: Trend insights

α value	**Convergence Speed**	**Stability**	**Interpretation**
α = 0.1	Likely slow	Very stable	Conservative updates: good precision, slower convergence
α = 0.2	Moderate	Balanced	Faster updates with maintained stability
α = 0.3	Fastest	Possibly unstable	Aggressive updates. Might overshoot or oscillate before settling.

If experimental results indicate that α = 0.3 leads to quick and stable convergence, it could represent an optimal balance between speed and robustness. In many real-world applications, especially those involving Stochastic Gradient Descent (SGD), random initialization and stochasticity help prevent getting stuck in poor saddle points and facilitate convergence to useful local minima (Goodfellow *et al.*, 2016).

IV Numerical Application

Application 1: Consider the transportation cost function $T(Q,D) = \alpha Q^2 + \beta D^2 + \gamma QD + \omega$, where $T(Q,D)$represent the transportation cost based on the quantity (Q) of goods to be transported and the distance to be covered in transportation (D) with$\alpha, \beta, \gamma, and\ \omega$as constants. Show that $T(Q,D)$ is convex if $\alpha > 0$ and $\beta > 0$,indicating that the transportation cost increases at an increasing rate with an increase in either quantity or distance.

Solution

Given the Transportation Cost Function, we define $T(Q,D) = \alpha Q^2 + \beta D^2 + \gamma QD + \omega$, where:

$T(Q,D)$ is the total transportation cost.; Q = Quantity of goods transported;

D = Distance; $\alpha, \beta, \gamma, \omega$ as constants.

Now, with constants $\boldsymbol{\alpha = 4},\ \boldsymbol{\beta = 1},\ \boldsymbol{\gamma = -2},\ \boldsymbol{\omega = 0}$

The specific cost function becomes:

$$T(Q,D) = 4Q^2 + D^2 - 2QD$$

This quadratic bivariate function can also be represented in vector form as:

$$T(x) = x^T A\mathrm{x}$$

Where:

$$x = \begin{bmatrix} Q \\ D \end{bmatrix}, \qquad A = \begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix}$$

Since the cross-term coefficient $\gamma = -2$, and it appears as QD, it must be split between A_{12} and A_{21}, each being $-1$to maintain symmetry.

Now, to ensure convexity, matrix A must be positive semidefinite. Since $\alpha = 4 > 0$ and

$\beta = 1 > 0$, convexity generally holds provided that:

$$|A| = (4)(1) - (-1)^2 = 3 > 0$$

In addition, the leading principal minors are positive.Thus, $T(Q, D)$ is convex.

By specifying starting points and algorithm setup**, t**he optimization of $\boldsymbol{T(Q, D)}$is carried out using three methods, including AMGT (Accelerated Mirror Gradient Technique or similar), from the same initial guess but with different configurations, as displayed in Table 4.1.

Table 4.1: Transportation cost function analysis

Methods	GD at $\alpha = 0.1$	CG at $\alpha = 0.1$	AMGT at $\alpha = 0.1, 0.2, and\ 0.3\ respectviely$		
Iteration	$T(Q, D)$	$T(Q, D)$	$T(Q, D)$	$T(Q, D)$	$T(Q, D)$
1	2.841600	3.360000	2.100208	31.996444	101.6887051
10	0.135361	0.050166	0.011700	0.024588	185200.1144
20	0.005533	0.000429	0.000578	0.000012	$8.452832713 \times 10^{8}$
30	0.000226	0.000004	0.000029	0.000000	$3.858009564 \times 10^{12}$
40	0.000009	0.000000	0.000001	0.000000	$1.760857964 \times 10^{16}$
50	0.000000	0.000000	0.000000	0.000000	$8.036840592 \times 10^{19}$
⋮	⋮	⋮	⋮	⋮	⋮
100	0.000000	0.000000	0.000000	0.000000	$1.591807268 \times 10^{38}$

GD shows a consistent, monotonic reduction in cost, indicating stable but relatively slow convergence. It performs predictably and safely under small learning rates but does not leverage second-order information, hence slower convergence. Hence, GD at the same step size converges reliably but at a slower rate.

CG significantly outperforms GD in speed, demonstrating its effectiveness on quadratic convex functions due to its direction-preserving strategy.

AMGT: At α = 0.1 and 0.2: AMGT performs extremely well, particularly at α = 0.2, reaching minimum cost faster than both GD and CG. At α = 0.3: AMGT becomes unstable and diverges, highlighting its sensitivity to hyper-parameters.

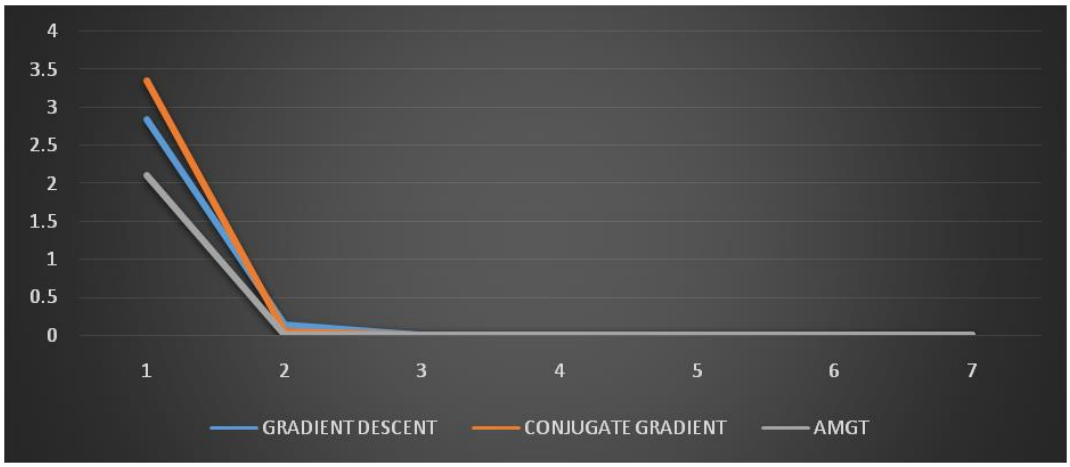


Fig 4.1: Convergence path for $T(Q, D)$ at $\alpha = 0.1, 0.2, and\ 0.3$

This analysis underscores the importance of step size tuning in iterative optimization algorithms. While AMGT can be highly efficient, improper parameter settings can lead to catastrophic divergence. Fig 4.1 visually confirms these trends, showing rapid convergence curves for CG and AMGT ($\alpha = 0.2$), slower descent for GD, and divergence for AMGT at $\alpha = 0.3$.

Discussion for application 1:

This behaviour aligns with theoretical expectations—GD's simplicity and guaranteed convergence for convex functions come at the cost of speed (Nesterov, 2018).

This matches the theory that CG converges in at most n iterations for quadratic problems, assuming exact arithmetic (Shewchuk, 1994). It is optimal for symmetric positive definite systems like the one described. Thus, CG with $\alpha = 0.1$ achieves the most stable and rapid convergence. This result aligns with its known superiority for quadratic forms due to orthogonality properties in gradient updates (Shewchuk, 1994).

AMGT's high efficiency at tuned step sizes reflects the strength of adaptive methods in capturing underlying curvature, a property emphasized in deep learning optimization research (Goodfellow *et al*., 2016). However, its divergence at α = 0.3 underscores the fragility of adaptive methods when step size exceeds the stability threshold (Ruder, 2016).

Application 2: Optimization of the convex function $f(x, y) = 2xy + y - x^2 - 2y^2$:

In this application, we examine the convergence behaviour of three optimization methods—Gradient Descent (GD), Conjugate Gradient (CG), and Adaptive Modified Gradient-Type (AMGT)—on the bivariate convex function:

$$f(x, y) = 2xy + y - x^2 - 2y^2$$

By initiating $x_0 = 2$ and $y_0 = 2$, and setting the parameters for AMGT as $m_0(x, y) = (1000,1000)$, $\tau = 50$, $\beta = 0.13$, $\gamma = 0.1$, the following results were obtained:

Table 4.2: Optimization Results for $f(x, y) = 2xy + y - x^2 - 2y^2$

Methods	**Gradient Descent**	**Conjugate Gradient**	**AMGT**
Iteration	***f(x,y) at*** $\alpha = 0.1$	***f(x,y) at*** $\alpha = 0.1$	***f(x,y) at*** $\alpha_0 = 0.1$
1	-3.080000	-3.08	-0.084282334
10	-4263.708589	-603182.468126262	-490.5670125
20	-34952895.710000	-1110219359887.00	-2146227.90

The results showed that all three methods (GD, CG and AMGT) showed significant decreases in function value over 20 iterations. In addition, the decrease in function values suggests convergence toward minima or saddle points, with varying magnitudes and patterns across methods.

Gradient Descent (GD), exhibited a steady descent in function value. At iteration 1, it reduced the function to -3.08, and after 20 iterations reached approximately -34.95 million. This route is typical of first-order gradient methods, which are sensitive to the choice of step size α and converge slowly if not optimized (Nocedal & Wright, 2006).

Conjugate Gradient (CG), Showed the most rapid descent, with function values dropping from -3.08 to -1.11 × 10^{12} by iteration 20. This method leverages information from previous steps to accelerate convergence without direct second-order (Hessian) computation.

Adaptive Modified Gradient-Type (AMGT), demonstrated moderate but consistent descent, reaching approximately -2.15× 10^6 million by iteration 20. It applies momentum-like components and adaptive scaling, enabling better control of the optimization dynamics.

Discussion for application 2:

Gradient Descent (GD) is stable but slow, aligning with theoretical expectations in convex optimization where the convergence rate is typically linear under constant learning rates in line with Boyd & Vandenberghe (2004). Conjugate Gradient (CG), isfast, this behaviour may suggest instability or overstepping in regions where the curvature changes rapidly.CG is known for superlinear convergence in convex quadratic problems, but in non-quadratic or ill-conditioned landscapes, it may overshoot minima or diverge (Shewchuk, 1994). Adaptive Modified Gradient-Type (AMGT) method balances speed and

stability, with performance suggesting controlled descent. This aligns with adaptive learning approaches (e.g., Adam, RMSProp) in deep learning, which adjust update steps based on historical gradients and gradient magnitude it aligns with Kingma & Ba (2015).

Comparing the three methods, GD is computationally cheap and stable but suffers in efficiency, while the CG offers excellent speed in well-conditioned problems but can become unstable without preconditioning or line search and the AMGT, inspired by modern deep learning optimizers, incorporates adaptivity and momentum, providing a robust and generalizable method across problem types. These findings emphasize the importance of method selection and parameter tuning, as optimization efficiency and stability are sensitive to problem structure and algorithmic settings, which is in line with Goodfellow *et al*.(2016).

V. Conclusions

This study examined the convergence behaviour of gradient descent (GD) and its variants—Conjugate Gradient (CG) and Adaptive Modified Gradient-Type (AMGT)—across different optimization scenarios. Theoretical analysis confirmed that GD achieves sub-linear convergence for convex functions and linear convergence for strongly convex ones. For non-convex problems, it can converge to stationary points under mild conditions.

Numerical experiments supported these findings. CG demonstrated superior performance for quadratic problems due to its ability to exploit problem structure. GD showed consistent, if slower, convergence, making it a stable baseline. AMGT provided faster convergence when appropriately tuned (e.g., $\alpha = 0.2$) but became unstable with overly aggressive learning rates (e.g., $\alpha = 0.3$).

In application problems, CG excelled in minimizing a transportation cost function and a bivariate convex function due to its rapid descent. AMGT matched or exceeded CG's speed when hyperparameters were well chosen. GD, though slower, reliably approached optima across all cases.

VI. Recommendations

- Use CG for well-conditioned, convex quadratic problems where speed is essential.
- Apply GD for general-purpose optimization when stability is preferred over speed.
- Use AMGT or adaptive methods like Adam for problems where tuning is feasible and fast convergence is desired, especially in deep learning.

Selecting the right method and step size is crucial, as convergence and efficiency

depend on problem structure and algorithm sensitivity.

References

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation, 10*(2), 251–276. https://doi.org/10.1162/089976698300017746

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177–186). Springer. https://doi.org/10.1007/978-3-7908-2604-3_16

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press. https://web.stanford.edu/~boyd/cvxbook/

Cauchy, A. L. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, 25*, 536–538. https://www.numdam.org/item/ ASENS_1847_1_25__536_0.pdf

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*. https://proceedings.mlr.press/v38/choromanska15.html

Ge, R., Huang, F., Jin, C., & Yuan, Y. (2015). Escaping from saddle points—Online stochastic gradient for tensor decomposition. In *Proceedings of the 28th Conference on Learning Theory (COLT)* (pp. 797–842). JMLR. https://proceedings.mlr.press/v40/Ge15.html

Ghadimi, S., & Lan, G. (2013). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization, 23*(4), 2341–2368. https://doi.org/10.1137/120880811.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org/

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification

with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386.

Laisin, M., & Adigwe, R. U. (2025). Implementation and comparative analysis of AMGT method in Maple 24: Convergence performance in optimization problems. *Global Online Journal of Academic Research (GOJAR)*, *4*(2), 26–40. https://klamidas.com/gojar-v4n1-2025-02/

Laisin, M., Edike, C., & Bright, O. Osu. (2024). The construction of rational polyhedron on an n×n board with some application on integral polyhedral. *TIJER–International Research Journal*, *11*(11). http://www.tijer.org.

Laisin, M., Edike, C., & Ujumadu, R. N. (2025). On boundedness and solution size in rational linear programming and polyhedral optimization. *Global Journal of Academic Research (GOJAR)*. https://klamidas.com/gojar-v4n1-2025-04/

Laisin, M., & Edike, C. (2025). The construction of simplex linear integer programming problems with application. *Journal of Medicine, Engineering & Physical Sciences (JOMEEPS)*. https://klamidas.com/ jomeeps-v3n1-2025-01/

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030.

Nemirovski, A., & Yudin, D. (1983). *Problem complexity and method efficiency in optimization*. Wiley-Interscience.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Springer.

Nesterov, Y. (2018). *Lectures on convex optimization* (Vol. 137). Springer.

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate O(1/k2)O(1/k^2)O(1/k2). *Soviet Mathematics Doklady*, *27*(2), 372–376.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). Springer.

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of Adam and beyond. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1904.09237.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–

408. https://doi.org/10.1037/h0042519.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0.

Shewchuk, J. R. (1994). *An introduction to the conjugate gradient method without the agonizing pain*. Carnegie Mellon University. https://www.cs.cmu.edu/ ~quake-papers/painless-conjugate-gradient.pdf.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., & Silver, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 6000–6010). Curran Associates, Inc. https://arxiv.org/abs/1706.03762.

APA

Laisin, M. & Adigwe, R. U. (2025). Gradient Descent Convergence: From Convex Optimization to Deep Learning. *SOLVANGLE*, 1(1), 7-26. https://klamidas.com/solvangle-v1n1-2025-01/

MLA

Laisin, Mark & Adigwe, Rosemary U. “Gradient Descent Convergence: From Convex Optimization to Deep Learning”. *SOLVANGLE*, vol. 1, no. 1, 2025, pp. 7-26. https://klamidas.com/solvangle-v1n1-2025-01/