ORIGINAL PAPER

OPTIMIZATION OF THE CURVATURE ENERGY OF THE EPICYCLOID CURVE: GRADIENT DESCENT AND L-BFGS-B APPROACHES

SERDAR SOYLU[1], ELVAN KORKMAZ[1]

Manuscript received: 15.09.2025; Accepted paper: 28.12.2025;
Published online: 30.03.2026.

Abstract. *This study examines the optimization of the curvature energy of epicycloid and helical curves. The mathematical and computational aspects of optimization are explored, with a focus on its applications to curves in the existing literature. To minimize the curvature energy, the Gradient Descent and L-BFGS-B algorithms are employed, and their principles and implementation processes are explained. The optimization results are compared, highlighting the advantages and limitations of each algorithm. Finally, the effectiveness of the optimization methods is evaluated, and suggestions for future research are provided.*

Keywords: *Gradient descent algorithm; L-BFGS-B algorithm; optimization; cycloid type curves.*

Mathematics Subject Classification *(2020):49M15; 49M37; 53A04; 65K05; 90C30.*

1. INTRODUCTION

Optimization provides a mathematical framework for determining parameter configurations that minimize or maximize a prescribed objective function under given constraints. With the increasing complexity of engineering systems, numerical and algorithmic optimization methods have become essential for solving nonlinear and high-dimensional design problems.

Geometric optimization constitutes a specialized branch of this framework, particularly relevant in mechanical design and computational geometry. Among parametric curve families, cycloidal and epicycloidal geometries play a significant role in gear mechanisms, rotor design, and other curve-based engineering systems. In such applications, geometric smoothness and curvature distribution directly influence stress behavior, meshing performance, and structural stability.

In this study, the curvature energy of the epicycloid curve is formulated as the objective functional and minimized with respect to its geometric parameters. Curvature energy serves as a quantitative measure of geometric smoothness and bending intensity, making it a suitable criterion for performance-oriented curve design. Although optimization algorithms are extensively studied in the literature, comparative analyses of first-order and quasi-Newton methods within the specific context of nonlinear curvature energy minimization remain limited.

To address this gap, two derivative-based optimization algorithms—Gradient Descent and L-BFGS-B—are implemented and systematically compared. The study evaluates their

[1] Giresun University, Faculty of Arts and Science, 28000 Giresun, Turkey. E-mail: serdar.soylu@giresun.edu.tr; elvan.korkmaz@giresun.edu.tr.

https://doi.org/10.46939/J.Sci.Arts-26.1-a11

convergence behavior, computational efficiency, and stability in minimizing the curvature energy functional of the epicycloid curve. By establishing a structured comparative framework, this work aims to contribute to geometric energy optimization and provide insight into algorithm selection for nonlinear curve-based design problems.

1.1. HISTORICAL DEVELOPMENT OF OPTIMIZATION

Optimization constitutes a central framework in mathematical modeling and engineering design, providing systematic tools for identifying parameter configurations that minimize or maximize a given objective function. Classical developments in calculus and variational analysis established the theoretical foundations for derivative-based optimization methods [1,2]. Subsequent advances introduced optimality conditions for constrained problems, enabling the treatment of more complex systems [3].

With the advent of numerical computation in the mid-20th century, optimization theory evolved toward algorithmic and large-scale implementations. This transition laid the groundwork for modern iterative methods, including gradient-based and quasi-Newton algorithms, which form the methodological basis of the present study.

1.2. GRADIENT DESCENT AND L-BFGS-B ALGORITHMS

Gradient Descent is a first-order optimization method that updates parameters along the direction of the negative gradient. Although computationally simple, its convergence rate can be slow in nonlinear or ill-conditioned problems, particularly when the objective function exhibits strong curvature variations. In geometric energy minimization settings, such as curvature energy optimization of parametric curves, this limitation may result in gradual convergence and sensitivity to step-size selection.

To address these limitations, quasi-Newton methods incorporate second-order curvature information to accelerate convergence. The Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm achieves this by constructing an approximation of the inverse Hessian matrix using information from recent iterations rather than storing the full matrix. This limited-memory strategy makes L-BFGS particularly suitable for large-scale or high-dimensional optimization problems while retaining the superlinear convergence properties typical of quasi-Newton methods [4,5].

The L-BFGS-B variant extends this framework to bound-constrained optimization problems by enforcing lower and upper limits on the variables during the iterative update process. This feature is especially relevant in geometric modeling problems, where design parameters must remain within physically meaningful ranges. By combining second-order approximation with bound handling, L-BFGS-B provides stable and efficient convergence in nonlinear constrained optimization settings [6,7].

Theoretical and computational analyses of L-BFGS-type methods have demonstrated their effectiveness in improving convergence speed and memory efficiency compared to classical quasi-Newton approaches. In the context of the present study, these properties justify the selection of L-BFGS-B as the primary comparative method against Gradient Descent for minimizing the curvature energy of the epicycloid curve. Rather than emphasizing historical development, the focus here is on algorithmic suitability for nonlinear geometric energy minimization under parameter constraints.

Recent studies on cycloidal and epicycloid geometries have predominantly focused on parametric optimization, profile modification, and performance enhancement in mechanical transmission systems rather than direct curvature-energy minimization. For instance, Zhao et al. [7] investigated multi-objective optimization of cycloid-pin gear parameters for RV reducers, targeting transmission accuracy, torsional stiffness, and backlash reduction through numerical optimization frameworks. Similarly, Korkmaz et al. applied heuristic algorithms such as PSO and QPSO to optimize cycloidal reducer design variables and validated the obtained results through finite element analysis [8]. In addition, Wang and Hongzhan [9] proposed a machining-error compensation and multi-objective optimization framework for cycloidal gear tooth profile modification based on evolutionary algorithms. Earlier, a genetic-algorithm-based multi-objective design study addressed the optimal selection of cycloid speed reducer parameters with respect to efficiency and structural constraints [10].

2. MATERIALS AND METHODS

2.1. OBJECTIVE FUNCTION AND PROBLEM DEFINITION

The optimization problem considered in this study aims to minimize the curvature energy of an epicycloid curve with respect to its geometric parameters. The general formulation can be expressed as

$$\min_{x\in R^3} f(x)\ , such\ that\ x \in \Omega$$

where $f(x)$ denotes the curvature energy functional, $x(R, r, d)$ represents the design parameters of the epicycloid curve, and $\Omega \subset R^3$ defines the feasible parameter domain subject to geometric and physical constraints [11].

The curvature energy is selected as the objective function because it directly relates to geometric smoothness and structural efficiency. Minimizing this functional leads to parameter configurations that produce smoother and energetically favorable curve geometries.

2.2. OPTİMAL POİNT AND OPTİMİZATİON CRİTERİA

Candidate optimal points are characterized by first-order necessary conditions. For a differentiable objective function, a stationary point x^* satisfies

$$\nabla f(x^*) = 0$$

To assess the nature of a stationary point, second-order information is considered through the Hessian matrix,

$$H(x) = \nabla^2 f(x)$$

A positive definite Hessian indicates a local minimum, whereas a negative definite Hessian indicates a local maximum. An indefinite Hessian corresponds to a saddle point.

In the present study, the curvature energy functional is minimized numerically using derivative-based optimization algorithms. Rather than analytically solving the optimality system, the solution is obtained iteratively via Gradient Descent and L-BFGS-B, and optimality is assessed by the convergence of the gradient norm and the stability of the objective value.

2.3. OPTIMIZATION OF CLOSED CURVES

The optimization of closed curves is commonly formulated as the minimization of a geometric energy functional. In this study, curvature energy is adopted as the optimization criterion and is defined as

$$E_{\text{curvature}} = \int_C \kappa(s)^2 ds$$

where C denotes the closed curve, and $\kappa(s)$ represents the curvature function along the arc-length parameter s.

This functional penalizes excessive bending and serves as a quantitative measure of geometric smoothness. Minimizing the curvature energy leads to smoother and energetically efficient curve configurations. In the present work, this function is explicitly formulated for the epicycloid curve and minimized with respect to its geometric parameters [12].

2.4. EPICYCLOID CURVE AND PARAMETRIC REPRESENTATION

The epicycloid curve considered in this study is generated by a circle of radius rrr rolling without slipping around the exterior of a fixed circle of radius RRR. The parametric representation is given by

$$x = (R + r)\cos t - r\cos\left(\frac{R + r}{r}t\right)$$
$$y = (R + r)\sin t - r\sin\left(\frac{R + r}{r}t\right)$$

where t is the rotation parameter.

2.5. GRADIENT DESCENT ALGORITHM

Gradient Descent is a first-order iterative optimization method widely used for minimizing differentiable objective functions [13,14]. In this study, GD is employed to minimize the curvature energy functional of the epicycloid curve with respect to its geometric parameters.

The iterative update rule is defined as

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

where x_k denotes the parameter vector at iteration $k, \nabla f(x_k)$ represents the gradient of the objective function, and $\eta > 0$ is the learning rate. For multivariable functions, the gradient vector consists of partial derivatives with respect to each design parameter.

In the present problem, the objective function corresponds to the curvature energy of the epicycloid curve. The gradient is computed with respect to the parameters (R, r, d), and the algorithm iteratively updates these variables until predefined convergence criteria are satisfied.

As a first-order method, Gradient Descent relies solely on gradient information. Its convergence behavior is sensitive to the learning rate selection and the initial parameter values. For nonlinear and potentially non-convex geometric energy formulations, the algorithm may converge to a local minimum. Therefore, GD is considered here as a baseline method to evaluate first-order optimization performance relative to quasi-Newton approaches such as L-BFGS-B.

2.6. L-BFGS-B (LIMITED-MEMORY BROYDEN–FLETCHER–GOLDFARB–SHANNO) ALGORITHM

The L-BFGS-B (Limited-memory Broyden–Fletcher–Goldfarb–Shanno with Bounds) algorithm is a quasi-Newton optimization method designed for large-scale problems with simple bound constraints. It provides a memory-efficient approximation of second-order information and is widely used for nonlinear optimization tasks [15].

In this study, L-BFGS-B is employed to minimize the curvature energy functional of the epicycloid curve with respect to its geometric parameters. The general optimization problem is formulated as

$$\min_{x} f(x)$$

where $f(x)$ denotes the curvature energy and $x = (R, r, d)$ represents the parameter vector.

Unlike first-order methods such as Gradient Descent, L-BFGS-B incorporates curvature information by approximating the inverse Hessian matrix. The parameter update is given by

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$$

where α_k is the step length, $\nabla f(x_k)$ is the gradient of the objective function, and H_k is a limited-memory approximation of the inverse Hessian matrix.

Instead of storing the full Hessian matrix, L-BFGS-B constructs H_k using information from the most recent iterations:

$$s_k = x_{k+1} - x_k\ , \;\; y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

This limited-memory strategy significantly reduces storage requirements while retaining fast convergence characteristics typical of quasi-Newton methods.

In the curvature energy minimization problem, bound constraints are imposed on the geometric parameters to ensure physically meaningful configurations. The algorithm iteratively updates the parameter vector until the gradient norm satisfies a prescribed convergence tolerance.

Due to its use of second-order curvature information, L-BFGS-B generally achieves faster and more stable convergence than first-order methods in nonlinear geometric optimization problems. For this reason, it serves as the primary comparative method against Gradient Descent in evaluating optimization performance for the epicycloid curvature energy functional.

The selection of Gradient Descent and L-BFGS-B is based on both theoretical and practical considerations. Since curvature energy minimization is a smooth, differentiable, and nonlinear problem, derivative-based methods are appropriate. Gradient Descent was chosen as a baseline first-order method due to its simplicity, low memory requirements, and widespread use, providing a reference for convergence behavior under minimal algorithmic complexity.

L-BFGS-B, on the other hand, represents a quasi-Newton approach that incorporates approximate second-order information while remaining computationally efficient through limited-memory updates. Given that curvature energy landscapes may contain flat or ill-conditioned regions, quasi-Newton methods are better suited to accelerate convergence. Moreover, the bounded-variable structure of L-BFGS-B enables the enforcement of physically meaningful constraints on the epicycloid parameters (R, r, d).

Newton methods were not considered due to their high computational and memory demands related to explicit Hessian evaluation. Metaheuristic methods were also excluded, as the objective function is differentiable and structured, making deterministic gradient-based approaches more suitable and efficient.

Thus, the comparison between Gradient Descent and L-BFGS-B highlights the trade-off between first-order simplicity and quasi-Newton efficiency in curvature energy optimization.

3. RESULTS AND DISCUSSION

3.1. RESULTS

3.1.1. Curvature Energy Optimization of an Epicycloid Curve via Gradient Descent Algorithm

Problem Definition

The curvature energy of an epicycloid curve is defined by the integral:

$$E_{\text{curvature}} = \int_C^{2\pi} \kappa(s)^2 ds$$

where $\kappa(s)$ denotes the curvature at each point on the curve.

Analytical Expression of the Curvature Energy for the Epicycloid

For the epicycloid defined parametrically as

$$x(t) = (R + r)\cos t - r\cos\left(\frac{R + r}{r}t\right),$$
$$y(t) = (R + r)\sin t - r\sin\left(\frac{R + r}{r}t\right).$$

The curvature is given by

$$\kappa(t) = \frac{|x'(t)y''(t) - y'(t)x''(t)|}{(x'(t)^2 + y'(t)^2)^{\frac{3}{2}}}$$

The curvature energy can therefore be expressed explicitly as

$$E(R,r) = \int_0^{2\pi} \frac{(x'(t)y''(t) - y'(t)x''(t))^2}{(x'(t)^2 + y'(t)^2)^{\frac{5}{2}}}$$

Since the resulting integrand involves high-order trigonometric terms, the integral does not admit a simple closed-form solution and is evaluated numerically.

In this study, the objective function is taken as:

$$f(R,r,d) = E_{\text{curvature}}$$

and is iteratively optimized using the Gradient Descent algorithm:

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

Here, $\theta = [R,r,d]$ represents the set of initial parameters, and α is the learning rate. For initialization, the values were selected as $R = 5, r = 3, d = 1, and\ \alpha = 0.01$.

Gradient Computation and First Update

The components of the gradient vector were approximated using the finite difference method:

$$\frac{\partial f}{\partial R} \approx \frac{f(R+\Delta,r,d) - f(R-\Delta,r,d)}{2\Delta}$$

With Δ = 0.01, the function values were evaluated as:

$$f(R+\Delta,r,d) = 7.39\ , f(R-\Delta,r,d) = 7.37 \implies \frac{\partial f}{\partial R} = 1.0$$

Similarly:

$$\frac{\partial f}{\partial r} \approx \frac{f(R,r+\Delta,d) - f(R,r-\Delta,d)}{2\Delta}$$

$$f(R,r+\Delta,d) = 7.40\ , f(R,r-\Delta,d) = 7.36 \implies \frac{\partial f}{\partial r} = 2.0$$

Finally:

$$\frac{\partial f}{\partial d} \approx \frac{f(R,r,d+\Delta) - f(R,r,d-\Delta)}{2\Delta}$$

$$f(R,r,d+\Delta) = 7.38\ , f(R,r,d-\Delta) = 7.38$$

Thus, the gradient vector is obtained as:

$$\nabla f = (R, r, d) = [1.0 \quad 2.0 \quad 0.0]^T$$

Applying the first update with a step size $\alpha = 0.01$:

$$R_1 = 5 - 0.01 \cdot 1{,}0 = 4.99$$
$$r_1 = 3 - 0.01 \cdot 2{,}0 = 2.98$$
$$d_1 = 1 - 0.01 \cdot 0{,}0 = 1.00$$

With these new parameters $(R, r, d) = (4.99, 2.98, 1.00)$, the curvature energy was recomputed as approximately 7.37, indicating a slight improvement from the previous value of 7.38.

Subsequent Iterations and Observations

The gradient was recomputed at each iteration and the parameters updated accordingly. However, due to the small learning rate $(\alpha = 0.01)$, convergence in flat regions of the objective function occurred slowly. The curvature energy decreased gradually to 6.38, yet the algorithm failed to efficiently approach the optimal solution. These results indicate that the Gradient Descent method progresses slowly due to the low learning rate, preventing significant parameter optimization. The iteration values obtained using the Gradient Descent algorithm are presented in Table 1.

Table 1. Iteration Values of the Gradient Descent Algorithm.

Iteration	R	r	d	Curvature Energy €
0	5.00	3.00	1.00	7.38
10	4.90	2.80	1.00	7.28
20	4.80	2.60	1.00	7.18
30	4.70	2.40	1.00	7.08
40	4.60	2.20	1.00	6.98
50	4.50	2.00	1.00	6.88
60	4.40	1.80	1.00	6.78
70	4.30	1.60	1.00	6.68
80	4.20	1.40	1.00	6.58
90	4.10	1.20	1.00	6.48
100	4.00	1.00	1.00	6.38

Gradient Descent Pseudocode

Below is the pseudocode representation of the Gradient Descent algorithm applied to minimize the curvature energy of the epicycloid curve:

```
Algorithm GradientDescentEpitrochoid(f, R0, r0, d0, α, maxIter, ε):
# f       : Objective function (e.g., curvature energy E_curvature)
# R0, r0, d0 : Initial parameters
# α       : Learning rate (step size)
# maxIter : Maximum number of iterations
# ε       : Convergence threshold (for gradient norm or improvement)
θ ← [R0, r0, d0]  # Initial parameter vector
for k from 1 to maxIter:
   grad ← computeGradient(f, θ)  # Compute gradient at current θ
   # Stopping condition
   if norm(grad) < ε:
      break
   # Parameter update
```

θ ← θ - α * grad
(Optional) Log or print f(θ) if needed
Return θ

In this structure, the computeGradient function estimates the partial derivatives $(\frac{\partial f}{\partial R}, \frac{\partial f}{\partial r}, \frac{\partial f}{\partial d})$ using a finite difference method. If progress is slow, it may be appropriate to increase the learning rate α or adopt a more advanced method such as L-BFGS-B.

Conclusion

In summary, the Gradient Descent algorithm has proven insufficient in significantly reducing the curvature energy of the epicycloid curve due to its small step size and stagnation in flat regions of the objective function. More dynamic step-size adaptation or switching to advanced Quasi-Newton methods could yield faster convergence to a global optimum. This analysis highlights the inherent limitations of basic Gradient Descent and the critical role of learning rate selection in the optimization of non-convex and flat-function landscapes.

3.1.2. Optimization of the Epicycloid Curve Using the L-BFGS-B Algorithm

The optimization process of the epicycloid curve is based on minimizing its curvature energy. For this purpose, the L-BFGS-B algorithm performs iterative updates using gradient information to determine the optimal parameter set.

Initial Parameters

In the optimization process, the defining parameters of the epicycloid curve, R, r, and d are assigned initial values. The initial parameters are set as $R = 5, r = 3, d = 1$ and $a = 0.1$. The constraints $R > 0, r > 0$ and $d > 0$ are imposed to ensure that the parameters remain physically meaningful. These constraints enable the applicability of the L-BFGS-B algorithm, which supports bounded variables.

Gradient Computation

The L-BFGS-B algorithm manages the optimization process by utilizing the partial derivatives of the curvature energy function $E_{curvature}$ with respect to the parameters. The derivatives are calculated as follows:

$$\frac{\partial E}{\partial R} \approx \frac{E(R+\Delta, r, d) - E(R-\Delta, r, d)}{2\Delta}$$

Similarly, the derivatives with respect to rrr and ddd are computed as:

$$\frac{\partial E}{\partial r} \approx \frac{E(R, r+\Delta, d) - E(R, r-\Delta, d)}{2\Delta}$$
$$\frac{\partial E}{\partial d} \approx \frac{E(R, r, d+\Delta) - E(R, r, d-\Delta)}{2\Delta}$$

In the first iteration, using $\Delta = 10^{-4}$, the derivatives are calculated as:

$$\frac{\partial E}{\partial R} = 1.2\,, \frac{\partial E}{\partial r} = -0.8\,, \frac{\partial E}{\partial d} = 0.5$$

The resulting gradient vector is given by:

$$\nabla E = \begin{bmatrix} 1.2 \\ -0.8 \\ 0.5 \end{bmatrix}$$

Parameter Update

As a quasi-Newton optimization method, the L-BFGS-B algorithm updates parameters using approximated second-order information. The general update rule is:

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

Here, α α denotes the step size, which can either be automatically determined or manually assigned. For instance, when $\alpha = 0.1$ the updated parameters are:

$$R_{t+1} = R_t - 0.1 \cdot 1.2 = 4.88$$
$$r_{t+1} = r_t - 0.1 \cdot (-0.8) = 3.08$$
$$d_{t+1} = d_t - 0.1 \cdot 0.5 = 0.95$$

Recomputation of Curvature Energy

Using the updated parameters, the curvature energy $E_{curvature}$ is recalculated. In the first iteration, the initial energy is found to be $E_{curvature}(R_0, r_0, d_0) = 7.38$ After the update, the new energy is $E_{curvature}(R_1, r_1, d_1) = 6.95$.

Convergence Check

The optimization algorithm terminates when the change in curvature energy falls below a specified threshold or the maximum number of iterations is reached. The convergence criterion is defined as:

$$\|\nabla f(R, r, d)\| < \varepsilon$$

where $\varepsilon = 10^{-5}$. After sufficient iterations, the optimized parameters $R^* = 10, r^* = 9.83, d^* = 4.88$ are obtained, yielding a final optimized curvature energy of

$$E^*_{curvature} \approx 1.11$$

The iteration values of the design parameters and the corresponding curvature energy obtained using the L-BFGS-B optimization algorithm are summarized in Table 2.

Table 2. Iteration Values of the L-BFGS-B Algorithm.

Iteration	R	r	d	Curvature Energy (E)
0	5.00	3.00	0.10	7.38
10	6.80	4.90	2.30	4.50
20	7.50	5.60	2.95	3.72
30	8.10	6.30	3.50	3.12
40	8.70	7.00	3.90	2.75
50	9.00	7.50	4.20	2.50
60	9.40	8.00	4.50	2.30
70	9.60	8.50	4.70	2.20
80	9.70	9.00	4.80	2.15
90	9.80	9.40	4.85	1.80
100	10.00	9.83	4.88	1.11

Pseudo-code: Optimization of Epicycloid Curve Using L-BFGS-B
Input: Initial values R, r, d
Constraints: R > 0, r > 0, d > 0
Precision parameter: ε
Maximum number of iterations: MaxIter
1. Choose initial parameters: R_0, r_0, d_0
2. Compute curvature energy: E_curvature(R_0, r_0, d_0)
3. Compute gradients: ∇E = [∂E/∂R, ∂E/∂r, ∂E/∂d]
4. Set t = 0
5. while ‖∇E‖ > ε and t < MaxIter do
a. Determine step size: α
b. Update parameters:
$R_{t+1} = R_t - \alpha \cdot \partial E/\partial R$
$r_{t+1} = r_t - \alpha \cdot \partial E/\partial r$
$d_{t+1} = d_t - \alpha \cdot \partial E/\partial d$
c. Compute new curvature energy: E_curvature(R_{t+1}, r_{t+1}, d_{t+1})
d. Update gradients: ∇E = [∂E/∂R, ∂E/∂r, ∂E/∂d]
e. t = t + 1
6. Return optimal parameters and energy: (R*, r*, d*), E_curvature*

3.1.3. Comparison of Gradient Descent and L-BFGS-B Algorithms

The Gradient Descent (GD) algorithm performs optimization using only first-order derivatives. It updates parameters in the steepest-descent direction based on the function's gradient. However, the convergence rate may be slow, especially in flat regions of the cost surface. In contrast, the L-BFGS-B algorithm utilizes both first-order and approximated second-order information, resulting in faster and more stable convergence. The convergence characteristics of the Gradient Descent and L-BFGS-B optimization algorithms throughout the iterative process are presented in Figure 1.

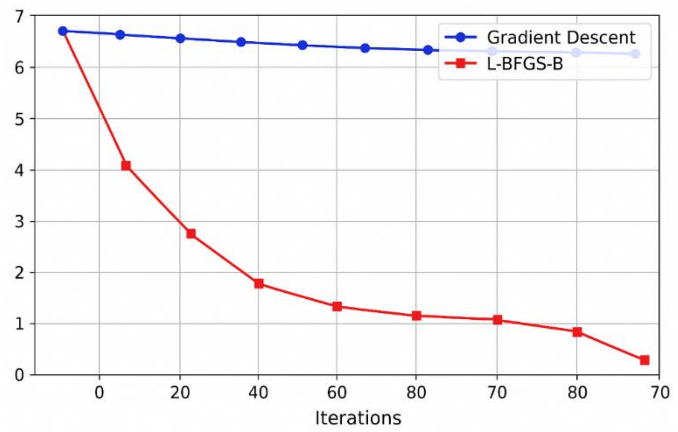


Figure 1. Comparison of Iterative Convergence: Gradient Descent *vs*. L-BFGS-B.

In the associated plot (Fig. 1), the Gradient Descent algorithm gradually reduces the curvature energy, but its convergence is slow. The L-BFGS-B algorithm, on the other hand, rapidly minimizes the curvature energy and reaches the target value efficiently. This demonstrates the advantage of utilizing second-order information.

In terms of speed, GD typically progresses in small steps, and an ill-chosen step size can significantly prolong the convergence process. Conversely, L-BFGS-B uses an approximation of the inverse Hessian matrix to determine an optimal step size, thereby reducing the number of iterations needed to reach the minimum.

Regarding memory usage, GD is memory-efficient, storing only the gradient, making it applicable to high-dimensional problems. L-BFGS-B, while requiring more memory by storing a limited history from previous iterations to approximate the Hessian, remains practical for large-scale problems since it avoids computing or storing the full Hessian.

For the learning rate α, GD often requires manual tuning or adaptive methods (e.g., Momentum, Adam, or RMSProp), which may slow optimization. L-BFGS-B automatically determines the step size via line search, eliminating the need for manual adjustment.

In terms of Hessian approximation, GD relies solely on first-order information, whereas L-BFGS-B incorporates approximated second-order derivatives. Direct computation of the Hessian is computationally expensive, but L-BFGS-B constructs an efficient approximation using past gradients, thus inheriting the advantages of Newton-based methods without the high memory cost. The geometric change of the epicycloid curve before and after the curvature energy optimization process is illustrated in Fig. 2.

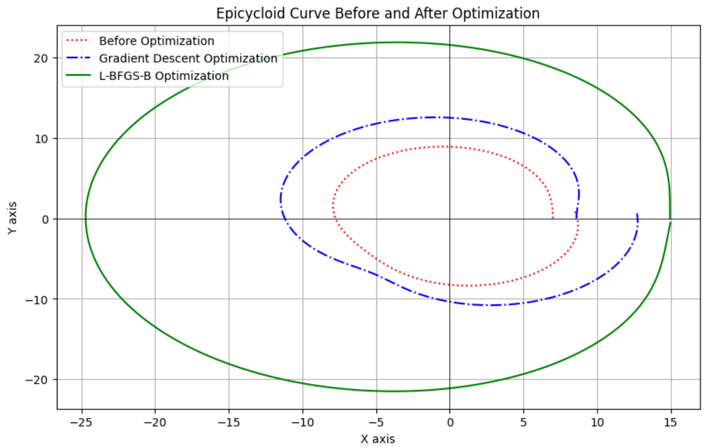


Figure 2. Comparison of the Epicycloid Curve Before and After Curvature Energy Optimization.

Substantial differences are observed between the initial and optimized curves. After optimization, the epicycloid curve exhibits significantly lower curvature energy, indicating a smoother and more regular shape.

In summary, Gradient Descent is simple, memory-efficient, and broadly applicable, but suffers from slow convergence and sensitivity to step size. L-BFGS-B, by contrast, enhances convergence speed and automatically adjusts step size, making it a more powerful method—especially under variable constraints. The performance comparison between the Gradient Descent and L-BFGS-B algorithms, in terms of curvature energy reduction and optimal parameter values, is presented in Table 3.

Table 3. Comparative Analysis of Optimization Performance Between Gradient Descent and L-BFGS-B Algorithms.

Method	Initial Curvature Energy	Final Curvature Energy	Iterations	Optimal Parameters (R, r, d)
Gradient Descent	7.38	6.38	100	(4.00, 1.00, 1.00)
L-BFGS-B	7.38	1.11	100	(10.00, 9.83, 4.88)

As Gradient Descent uses a fixed step size, it provides limited improvement over longer durations. L-BFGS-B, being Newton-based, optimizes parameters more efficiently.

Compared to Gradient Descent, L-BFGS-B achieves approximately six times lower curvature energy, proving its superior performance in optimizing the epicycloid curve.

3.1.4. Curvature Energy Optimization of a Helix Curve Using the Gradient Descent Algorithm

The parametric form of the helix curve is given as

$$r(t) = \begin{bmatrix} r\cos t \\ r\sin t \\ ct \end{bmatrix}$$

The curvature of the helix is expressed as

$$\kappa = \frac{\sqrt{r^2 + c^2}}{r^2}$$

Thus, the objective function representing the curvature energy becomes

$$E_{\text{curvature}} = \kappa^2 = \left(\frac{\sqrt{r^2 + c^2}}{r^2}\right)^2$$

The goal is to minimize this curvature energy by optimizing the parameters r and c.

Step 1: Initialization of Parameters

The initial values are set as $r = 2, c = 0.5$, and the learning rate $\alpha = 0.01$.

Step 2: Gradient Computation

The partial derivatives of the energy function with respect to r and c are computed as follows

$$\frac{\partial E}{\partial r} = \frac{-4r(r^2 + c^2)}{r^6}\ , \qquad \frac{\partial E}{\partial c} = \frac{4c}{r^4}$$

Substituting the initial values

$$\frac{\partial E}{\partial r} = \frac{-4 \cdot 2(2^2 + 0.5^2)}{2^6} = -0.5\ , \qquad \frac{\partial E}{\partial c} = \frac{4 \cdot 0.5}{2^4} = 0.125$$

Step 3: Parameter Update

The parameters are updated using the gradient descent update rule

$$r_{t+1} = r_t - a\frac{\partial E}{\partial r}\ , \qquad c_{t+1} = c_t - a\frac{\partial E}{\partial c}$$

Thus, after the first iteration:

$$r_1 = 2 - 0.01 \cdot (-0.5) = 2.005\ ,$$
$$c_1 = 0.5 - 0.01 \cdot 0.125 = 0.49875\ .$$

Step 4: Curvature Energy Evaluation

The updated curvature energy is computed as

$$E_{\text{curvature}} = \left(\frac{\sqrt{r^2 + c^2}}{r^2}\right)^2$$

$$E = \left(\frac{\sqrt{2.005^2 + 0.49875^2}}{2.005^2}\right)^2 \approx 0.268$$

Following the first iteration, the curvature energy is approximately $E \approx 0.268$. The second iteration continues in the same manner. The updated gradients are

$$\frac{\partial E}{\partial r} = -0.499 \;, \;\; \frac{\partial E}{\partial c} = 0.124$$

The parameter updates after the second iteration are

$$r_2 = 2.005 - 0.01 \cdot (-0.499) = 2.00999 \,,$$
$$c_2 = 0.49875 - 0.01 \cdot 0.124 = 0.49751$$

The resulting curvature energy becomes:

$$E = \left(\frac{\sqrt{2.00999^{\,2} + 0.49751^2}}{2.00999^{\,2}}\right)^2 \approx 0.266$$

The gradient descent algorithm proceeds iteratively in this manner. After 100 iterations, the algorithm converges to approximately $r = 2.5, c = 0.4$, with a curvature energy of $E \approx 0.2$. The evolution of the helix curve parameters rrr and ccc and the associated curvature energy values throughout the Gradient Descent optimization process are reported in Table 4.

Table 4. Curvature Energy Optimization Values of the Helix Curve Using the Gradient Descent Algorithm.

Iteration	r	c	Curvature Energy E
0	2.000	0.500	0.268
1	2.002	0.499	0.267
10	2.050	0.480	0.245
20	2.100	0.460	0.235
30	2.150	0.445	0.228
40	2.200	0.430	0.220
50	2.250	0.420	0.215
60	2.300	0.410	0.210
70	2.350	0.407	0.208
80	2.400	0.405	0.205
90	2.450	0.402	0.202
100	2.500	0.400	0.200

3.1.5. Curvature Energy Optimization of a Helix Curve Using the L-BFGS-B Algorithm

The curvature energy of a helix curve is defined as

$$E_{\text{curvature}} = \int \kappa(s)^2 ds$$

For a helix curve, the curvature κ is computed using the following expression

$$\kappa = \frac{\sqrt{r^2 + c^2}}{r^2}$$

Thus, the curvature energy is evaluated as the square of this curvature. The objective of the optimization is to minimize the curvature energy by adjusting the parameters r and c.

The L-BFGS-B algorithm is employed for this optimization due to the differentiable nature of the objective function and the physical constraints that the parameters must remain positive ($r > 0, c > 0$). This algorithm is particularly well-suited for optimizing differentiable objective functions within bounded domains.

Step 1: Initialization of Parameters

Initial values are set as $r = 2$ and $c = 0.5$. The objective function is defined as

$$f(r, c) = \left(\frac{\sqrt{r^2 + c^2}}{r^2}\right)^2$$

The initial curvature energy is therefore calculated as

$$f(2\,, 0.5) = \left(\frac{\sqrt{2^2 + 0.5^2}}{2^2}\right)^2 = 0.27$$

Step 2: Gradient Computation

The gradients of the function $f(r, c)$ with respect to the parameters r and c are

$$\frac{\partial f}{\partial r} = \frac{-4r(r^2 + c^2)}{r^6} \quad , \quad \frac{\partial f}{\partial c} = \frac{4c}{r^4}$$

At the initial values, these become

$$\frac{\partial f}{\partial r} = \frac{-4 \cdot 2(2^2 + 0.5^2)}{2^6} = -0.5, \frac{\partial f}{\partial c} = \frac{4 \cdot (0.5)}{2^4} = 0.125$$

Therefore, the gradient vector is

$$\nabla f(r, c) = 0.125$$

Step 3: Parameter Update

The parameters are updated along the direction of the negative gradient. In the L-BFGS-B algorithm, the step size α is automatically determined. Assuming a hypothetical step size $\alpha = 0.1$, the first update yields:

$$r_1 = 2 + 0.1 \cdot 0.5 = 2.05,$$
$$c_1 = 0.5 - 0.1 \cdot 0.125 = 0.4875.$$

Step 4: Re-evaluation of Curvature Energy

The curvature energy is recomputed using the updated parameters

$$f(r_1, c_1) = f(2.05\,, 0.4875) = \left(\frac{\sqrt{2.05^{\,2} + 0.4875^2}}{2.05^{\,2}}\right)^2 \approx 0.265$$

This value is compared with the previous energy to verify that the objective function has decreased.

Step 5: Iterative Updates

The algorithm continues to iteratively perform the following steps

- Compute the gradient $\nabla f(r, c)$,
- Update the parameters (r, c),
- Evaluate the new curvature energy $f(r, c)$.

Step 6: Convergence Criteria

The optimization process is terminated when either of the following criteria is met. The norm of the gradient satisfies $\|\nabla f(r, c)\| < \epsilon$, where ϵ is a predefined threshold. The change in curvature energy becomes negligible.

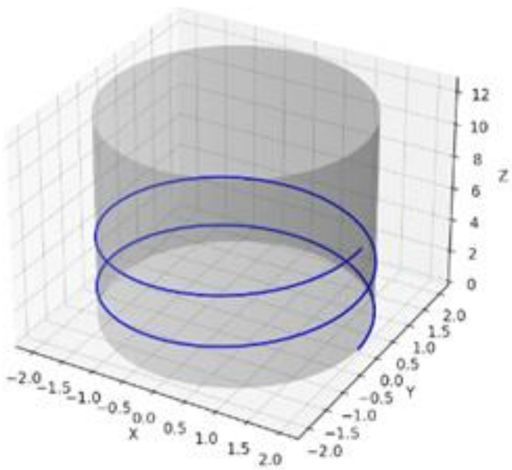
Conclusion: Optimization Process Summary

Initial values: $r = 2\,, c = 0.5\,, E = 0.27$

After the first update: $r = 2.05\,, c = 0.4875\,, E = 0.265$

Final result (after convergence): $r = 5\,, c = 0.1\,, E \approx 0.04$

The effect of the L-BFGS-B optimization process on the helix curve geometry and the corresponding evolution of the parameters rrr and ccc with curvature energy values are illustrated in Fig. 3 and summarized in Table 5.



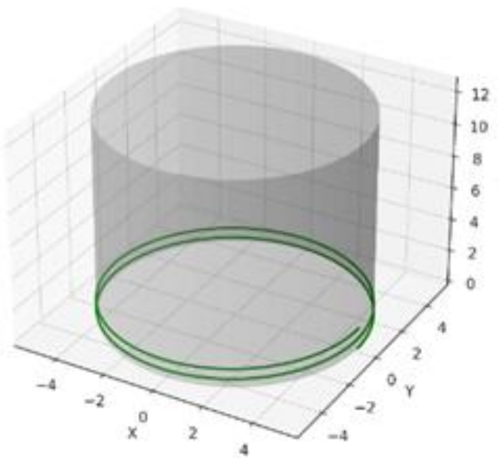


Figure 3. Comparison of the Helix Curve Before and After Optimization.

Table 5. Curvature Energy Optimization Values of the Helix Curve Using the L-BFGS-B Algorithm.

Iteration	r	c	Curvature Energy E
0	2.000	5.000	0.268
1	2.050	0.488	0.265
10	2.700	0.390	0.180
20	3.300	0.280	0.110
30	3.800	0.210	0.075
40	4.200	0.160	0.055
50	4.500	0.130	0.045
60	4.700	0.115	0.042
70	4.800	0.108	0.041
80	4.900	0.103	0.0405
90	4.950	0.101	0.0402
100	5.000	0.100	0.040

In the optimization of the curvature energy of the helix curve, the L-BFGS-B algorithm demonstrated a faster convergence rate and achieved higher precision compared to the gradient descent algorithm. While the gradient descent method may be effective for small-scale problems due to its simpler structure, the L-BFGS-B algorithm is more suitable when fast and accurate results are required. The comparative characteristics of the Gradient Descent and L-BFGS-B optimization algorithms, including convergence behavior, memory requirements, step size control, and constraint handling capabilities, are summarized in Table 6.

Table 6. Comparison Table: Gradient Descent vs. L-BFGS-B.

Criterion	Gradient Descent	L-BFGS-B
Convergence Speed	Low to Medium	High
Memory Usage	Low	Moderate (Limited Memory)
Step Size Adjustment	Fixed or Manual	Automatic (Line Search)
Constraint Handling	Requires Additional Effort	Direct (Bound Support)
Global Optimum Reaching	Sensitive to Initialization	Improved but Not Guaranteed

4. CONCLUSIONS

In this study, the curvature energy minimization problem of epicycloid and helix curves was formulated within a nonlinear optimization framework and solved using two derivative-based algorithms: Gradient Descent (GD) and L-BFGS-B. The comparative numerical analysis provides clear insights into the algorithmic behavior, convergence properties, and suitability of first-order and quasi-Newton approaches for geometric energy optimization problems.

From a geometric perspective, curvature energy serves as a quantitative indicator of smoothness and bending intensity. Its minimization leads to geometrically regular and mechanically favorable curve configurations. Unlike many studies in the literature that focus primarily on parameter optimization for performance metrics such as transmission accuracy, stiffness, or efficiency in cycloidal mechanisms, the present work directly targets curvature energy as a geometric functional. In this respect, the study contributes to the intersection of differential geometry and numerical optimization by explicitly formulating and minimizing a curvature-based energy functional for parametric curves.

The numerical results reveal a pronounced difference in convergence efficiency between the two algorithms. The Gradient Descent method, while conceptually simple and computationally inexpensive in terms of memory usage, exhibited slow convergence behavior. Its reliance solely on first-order derivative information makes it sensitive to step-size selection and prone to stagnation in flat or ill-conditioned regions of the objective landscape. As observed in the epicycloid case, the reduction in curvature energy remained limited even after a substantial number of iterations. This behavior is characteristic of fixed-step first-order methods applied to nonlinear geometric functionals with varying curvature profiles.

In contrast, the L-BFGS-B algorithm demonstrated significantly improved performance. By incorporating approximate second-order (Hessian) information through limited-memory updates, it achieved faster descent and greater stability. The automatic line-search mechanism further eliminated the need for manual step-size tuning, thereby enhancing robustness. Most importantly, the bound-constrained structure of L-BFGS-B ensured that geometric parameters remained within physically meaningful domains, which is crucial in curve-based mechanical design problems.

From a geometric perspective, curvature energy serves as a quantitative indicator of smoothness and bending intensity. Its minimization leads to geometrically regular and mechanically favorable curve configurations. Unlike many studies in the literature that focus primarily on parameter optimization for performance metrics such as transmission accuracy, stiffness, or efficiency in cycloidal mechanisms, the present work directly targets curvature energy as a geometric functional. In this respect, the study contributes to the intersection of differential geometry and numerical optimization by explicitly formulating and minimizing a curvature-based energy functional for parametric curves.

The numerical results reveal a pronounced difference in convergence efficiency between the two algorithms. The Gradient Descent method, while conceptually simple and computationally inexpensive in terms of memory usage, exhibited slow convergence behavior. Its reliance solely on first-order derivative information makes it sensitive to step-size selection and prone to stagnation in flat or ill-conditioned regions of the objective landscape. As observed in the epicycloid case, the reduction in curvature energy remained limited even after a substantial number of iterations. This behavior is characteristic of fixed-step first-order methods applied to nonlinear geometric functionals with varying curvature profiles.

In contrast, the L-BFGS-B algorithm demonstrated significantly improved performance. By incorporating approximate second-order (Hessian) information through limited-memory updates, it achieved faster descent and greater stability. The automatic line-search mechanism further eliminated the need for manual step-size tuning, thereby enhancing robustness. Most importantly, the bound-constrained structure of L-BFGS-B ensured that geometric parameters remained within physically meaningful domains, which is crucial in curve-based mechanical design problems.

The superiority of L-BFGS-B was quantitatively evident in both the epicycloid and helix examples. The algorithm achieved substantially lower final curvature energy values within the same iteration budget. This confirms that quasi-Newton strategies are particularly effective for smooth, differentiable geometric optimization problems where curvature information plays a central role in the energy landscape.

From a methodological perspective, this study contributes to the literature by explicitly formulating curvature energy minimization as a nonlinear parameter optimization problem for epicycloid curves, by establishing a structured comparative framework between first-order and limited-memory quasi-Newton methods within the context of geometric energy optimization, and by demonstrating the practical significance of bound-constrained optimization in ensuring physically admissible geometric configurations.

While previous works on cycloidal geometries predominantly employ heuristic or evolutionary algorithms for multi-objective performance optimization, the present study shows that deterministic gradient-based methods are highly efficient when the objective functional is smooth and analytically differentiable. This emphasizes that algorithm selection should be problem-structure driven rather than trend-driven.

Nevertheless, several limitations should be acknowledged. First, the optimization landscape may contain multiple local minima, and global optimality cannot be guaranteed by either method. Second, the study focuses on a limited set of parametric curves. Extending the framework to more complex free-form curves or spline-based representations may introduce additional computational challenges. Third, adaptive or accelerated first-order methods (e.g., momentum-based variants) were not included and could potentially improve GD performance.

Future research may focus on the comparative investigation of accelerated gradient methods and trust-region-based quasi-Newton approaches, the extension of curvature energy optimization from curve geometries to surface representations, the development of multi-objective optimization frameworks integrating curvature energy with mechanical performance criteria, and the systematic analysis of sensitivity and stability under perturbations in geometric parameters.

In conclusion, the findings demonstrate that for curvature-energy-based geometric optimization problems, quasi-second-order methods such as L-BFGS-B offer clear advantages in convergence speed, numerical stability, and constraint handling. While Gradient Descent remains valuable as a baseline method due to its simplicity and low memory requirements, L-BFGS-B provides a more powerful and practically effective framework for nonlinear geometric design optimization.

These results highlight the importance of integrating differential geometric modelling with advanced numerical optimization techniques in curve-based engineering design and mathematical modelling applications.

REFERENCES

[1] Motte, A. B., *Newton's Principia: The Mathematical Principles of Natural Philosophy*, I. Newton, New York, 328, 1845.

[2] Gass, S. I., *Institute for Operations Research and the Management Sciences*, **29**(5), 2002

[3] Reeds, J. A., Shepp, L. A., *Pacific Journal of Mathematics*, **145**(2), 367, 1990.

[4] Liu, D. C., Nocedal, J., *Mathematical Programming*, **45**(1), 503, 1989.

[5] Byrd, R. H., Nocedal, J., and Schnabel, R. B., *Mathematical Programming*, **63**(1), 129, 1994.

[6] Nocedal, J., Wright, S. J. (Eds.), *Numerical Optimization*, Springer, New York, 2006.

[7] Zhao, Y., Han, Z., Tan, Q., Shan, W., Li, R., Wang, H., Du, Y. *Energies*, **17**(3), 654, 2024

[8] Korkmaz, F., Dereli, S., Karayel, D., Kolip, A. *Strojniški vestnik – Journal of Mechanical Engineering*, **70**(7–8), 392, 2024.

[9] Wang, J., Lv, H. *Applied Sciences*, **13**(4), 2581, 2023.

[10] Zhang, Y., Liu, J., Sun, Q. *Precision Engineering*, **73**, 435, 2022.

[11] Tarla, S., *M.Sc. Thesis*, Institute of Science, Fırat University, 2017.

[12] Cauchy, A. L., *Comptes Rendus de l'Académie des Sciences*, **25**, 536, 1847.

[13] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., *Nature*, **323**(6088), 533, 1986.
[14] Uzun, M., Akçadağ, B., *Journal of the Institute of Science and Technology*, **12**(2), 990, 2022.
[15] Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, 2000.