Improved Solution Methods for Initial Value Problems of Ordinary Differential Equations Using Advanced Optimization Techniques

Qasim Abd Ali Tayyeh[1,*]

[1]Department of Mechanical Techniques,Al-Nasiriya Technical Institute, Southern Technical University,Thi-Qar, Al-Nasiriya 64001, Iraq

Email: qassim.tayih@stu.edu.iq

Abstract

Initial value problems (IVPs) of ordinary differential equations (ODEs) are ubiquitous in science and engineering applications, and the classical fourth-order Runge–Kutta (RK4) method is by far the most popular solver due to its good accuracy-to-cost ratio. Among all four-stage fourth-order explicit RK methods there are two free node parameters left after satisfying the eight B-series order conditions, thus allowing further systematic enhancements. Here we employ a hybrid multi-seed Particle Swarm Optimization (PSO)-Nelder–Mead algorithm to search for optimal RK node parameters (c_2, c_3) with respect to a minimax normalized objective over eight commonly used nonlinear benchmark problems. The resulting PSO-RK4 method with ($c_2 = 0.323665$, $c_3 = 0.653527$) retains both the exact same order of convergence and absolute stability region as the classical RK method called the 3/8-rule, but exhibits reduced maximum global error on each of the eight benchmarks when N = 300; average improvement of 27.9% with gains up to 46.0% on the Bernoulli equation, 29.7% on logistic growth, and 21.3% on exponential. Robustness of these gains with respect to multi-step-size (N = 100, 200, 300, 500) is demonstrated.

Keywords: Initial value problems; Runge–Kutta methods; Particle Swarm Optimization; Nelder–Mead simplex; hybrid metaheuristic optimization; Butcher tableau; Absolute stability; Nonlinear ODEs

1. Introduction

Ordinary differential equations (ODEs) are used to model the dynamic behavior of systems across physics, chemistry, biology, engineering, and finance [1]. When the state of these dynamical systems evolves over time from some initial condition, they are commonly formulated as initial value problems (IVPs) [2]. For most ODEs it is impossible to obtain a closed-form solution, thus numerical integration schemes are required [3]. Explicit Runge–Kutta (RK) methods are among the most commonly used ODE solvers because they are self-starting, simple to implement, and they offer a good trade-off between accuracy and computational complexity [4]. The classical RK4 method was published over 80 years ago and remains the default solver found in scientific computing libraries today. RK4 is still one of the workhorse solvers used in production codes across many disciplines [5]. One feature of the family of four-stage, fourth-order explicit RK methods that has received little attention is that they have two free parameters [6]. When imposing the eight B-series order conditions introduced by Butcher (1963), the node values of the intermediate stages c_2 and c_3 remain free [7]. The set of free parameters identifies a continuous one-dimensional family of methods that are all provably fourth-order accurate. RK4 and the 3/8-rule are but two samples from this family [8]. Despite being known for decades, there has been little work to directly leverage data-driven methods to explore the accuracy landscape of this family using modern metaheuristic search techniques [9].

Literature and motivation for this work. Prior works in tuning RK parameters are hindered by three shortcomings which remain consistent across contributions: (i) they optimize hyperparameters on a single test problem which

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

can lead to overfitting and regressions for other ODE classes [10], (ii) they rely on blind PSO or genetic algorithms without a local correction phase, leading to parameters with low precision [11] and (iii) they do not validate their chosen parameters against absolute stability criteria [12]. We aim to solve these issues by: providing a minimax objective over multiple problems (8 nonlinear benchmarks), implementing a PSO + Nelder–Mead optimization pipeline which enables us to obtain high precision parameters and validating theoretically and numerically that tuned PSO-RK4 has the same absolute stability region as the classical 3/8-rule.

2. Related Work

Metaheuristic optimization techniques have also recently been applied to the design of RK methods. This topic has received significant attention in literature recently, including several articles in the year 2025 alone. Anastassi [13] developed a computational evolutionary approach for designing RK methods that optimizes for global error directly, instead of considering the local truncation error. Particle swarm optimization was used to tune the four free coefficients of a sixth-order, eight-stage parametric RK scheme on two oscillatory benchmark problems (the (2+1)-dimensional nonlinear Schrödinger equation and the N-Body problem). By taking this approach, Anastassi avoided local minima and had to calculate derivatives of the solution with respect to the coefficients. While this approach shows promise for RK method design with metaheuristics, it was limited to designing for one class of problems (oscillatory dynamics).

In another vein, Goodship et al. [14] proposed a machine learning inspired "dynamic heuristic discovery routine" for fourth-order ESRK schemes using an Interior Point Optimizer augmented with hand-designed heuristics capable of producing sixteen-stage low-storage schemes. They demonstrated a 25% reduction in runtime over naive optimization of their objective on the Brusselator problem, though they validate their approach on only one example and do not perform any formal analysis of the absolute stability region. Tsitouras and Famelis [15] take a different approach to RK design. Their framework derived an 8(6) RK pair directly suited for linear inhomogeneous IVPs using differential evolution, matching the accuracy of standard 8(7) pairs with only eleven stages total; however, this approach is only suited for linear problems with constant coefficients. Habibah et al. [16] proposed a simplified fifth-order RK method whose coefficients are obtained from an optimized Taylor series expansion. The resulting method was shown to have superior accuracy and stability when compared to existing fifth-order RK schemes on several benchmark IVPs, however their method of optimization is purely analytical and does not take advantage of the freedom in parameters RK families offer.

Taken together, recent contributions have validated metaheuristic and optimization-based RK design. However, to the best of our knowledge, none consider (i) multi-objective optimization on multiple heterogeneous nonlinear test problems using a minimax criterion, (ii) a mixed global-local search paradigm using PSO and Nelder–Mead refinement, and (iii) an analytically enforced preservation of the magnitude of the stability region of the underlying RK family, all of which are considered here.

3. Proposed Methodology

Shown here is the proposed PSO-RK4 approach to computing solutions of initial value problems defined by ordinary differential equations. It consists of four complementary parts: a family of parameterized fourth-order Runge–Kutta methods with two free parameters per node, a set of eight benchmark nonlinear IVPs, a multi-problem minimax normalized objective function, and a hybrid PSO + Nelder–Mead method to compute globally robust node parameters.

3.1 Parametric Fourth-Order Runge–Kutta Family

The proposed method exploits the analytically derived two-parameter family of four-stage, fourth-order explicit Runge–Kutta methods. For the initial value problem [17]

$$dy/dt = f(t,y), y(t_0) = y_0, t \in [t_0, T] \quad (1)$$

an explicit s-stage RK method is uniquely defined by its Butcher tableau (A, b, c), where A is a strictly lower-triangular matrix of coefficients, b is the vector of weights and c = A·e is the node vector. For s = 4 and p = 4 there are exactly two free parameters left after imposing the canonical choices $c_1 = 0$ and $c_4 = 1$ in the eight B-series order conditions derived by Butcher (1963): the two intermediate nodes $c_2 = \alpha$ and $c_3 = \beta$. All other entries can be expressed in closed form [18,19]:

$$a_{21} = \alpha \quad (2)$$

$$a_{32} = \beta(\beta - \alpha) \,/\, [2\alpha(1 - 2\alpha)] \quad (3)$$

$$a_{31} = \beta - a_{32} \quad (4)$$

$$b_3 = 1 \,/\, [24 \cdot \alpha \cdot (1 - \beta) \cdot a_{32}] \quad (5)$$

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

$$b_2 = (2\beta - 1) / [12 \cdot \alpha \cdot (1 - \alpha) \cdot (\beta - \alpha)] \quad (6)$$

$$b_4 = 1/3 - b_2 \cdot \alpha^2 - b_3 \cdot \beta^2 \quad (7)$$

$$b_1 = 1 - b_2 - b_3 - b_4 \quad (8)$$

$$a_{43} = 1 / [24 \cdot a_{32} \cdot \alpha \cdot b_4] \quad (9)$$

$$a_{42} = (1/6 - b_3 \cdot a_{32} \cdot \alpha) / (b_4 \cdot \alpha) - a_{43} \cdot \beta / \alpha, a_{41} = 1 - a_{42} - a_{43} \quad (10)$$

The classical 3/8-rule corresponds to (α, β) = (1/3, 2/3). A candidate (α, β) is admitted only when all derived entries are finite, $b_4 \in [0.05, 0.60]$, and $\max|a_{ij}| \leq 8$, with the search domain restricted to $\alpha \in [0.15, 0.48]$ and $\beta \in [0.53, 0.90]$, as shown in Table 1 [20].

Table 1: Closed-form derivation of the parametric fourth-order Runge–Kutta family

Role	Definition / Closed-Form Expression	Symbol	Component
Second-stage node (optimized)	$\alpha \in [0.15, 0.48]$	$c_2 = \alpha$	Free parameter 1
Third-stage node (optimized)	$\beta \in [0.53, 0.90]$	$c_3 = \beta$	Free parameter 2
Canonical first node	0	c_1	Fixed node 1
Canonical fourth node	1	c_4	Fixed node 4
Derived from c_2	α	a_{21}	Stage coefficient
B-series order condition	$\beta(\beta - \alpha)/[2\alpha(1 - 2\alpha)]$	a_{32}	Stage coefficient
Row-sum identity	$\beta - a_{32}$	a_{31}	Stage coefficient
B-series order condition	$1/[24 \cdot a_{32} \cdot \alpha \cdot b_4]$	a_{43}	Stage coefficient
B-series order condition	$(1/6 - b_3 \cdot a_{32} \cdot \alpha)/(b_4 \cdot \alpha) - a_{43} \cdot \beta/\alpha$	a_{42}	Stage coefficient
Row-sum identity	$1 - a_{42} - a_{43}$	a_{41}	Stage coefficient
Quadrature weight	$(2\beta - 1)/[12 \cdot \alpha \cdot (1 - \alpha) \cdot (\beta - \alpha)]$	b_2	Weight
Quadrature weight	$1/[24 \cdot \alpha \cdot (1 - \beta) \cdot a_{32}]$	b_3	Weight
Quadrature weight	$1/3 - b_2 \cdot \alpha^2 - b_3 \cdot \beta^2$	b_4	Weight
Consistency: $\Sigma b_i = 1$	$1 - b_2 - b_3 - b_4$	b_1	Weight
Numerical conditioning	$b_4 \in [0.05, 0.60]$	b_4	Admissibility
Stability guard	≤ 8	$\max\|a_{ij}\|$	Admissibility

3.2 Benchmark Suite of Nonlinear Initial Value Problems

To prevent overfitting and ensure robustness across diverse solution behaviors, the optimization is evaluated simultaneously on eight nonlinear IVPs spanning scalar growth, scalar decay, stiff-like nonlinearity, and a two-dimensional predator–prey system. Problems P1–P7 admit closed-form exact solutions; P8 employs a high-accuracy reference computed by Dormand–Prince RK45 (rtol = 10^{-12}, atol = 10^{-14}). Specifically, the suite comprises:

- P1 — quadratic decay $y' = -y^2$ with exact solution $y(t) = 1/(1+t)$;
- P2 — logistic growth $y' = y(1 - y)$ with $y(t) = 1/(1 + 9 \cdot \exp(-t))$;

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

- P3 — cubic decay $y' = -y^3$ with $y(t) = 1/\sqrt{(1 + 2t)}$;
- P4 — power growth $y' = 3 \cdot y^{(2/3)}$ with $y(t) = (1 + t)^3$;
- P5 — square-root growth $y' = \sqrt{y}$ with $y(t) = (1 + t/2)^2$;
- P6 — exponential nonlinear $y' = \exp(-y)$ with $y(t) = \ln(1 + t)$;
- P7 — Bernoulli equation $y' = y^2 \cdot \sin(t)$ with $y(t) = 1/\cos(t)$; and
- P8 — the two-dimensional Lotka–Volterra system

$$x' = 1.5 \cdot x - x \cdot y, y' = -3 \cdot y + x \cdot y, x(0) = 10, y(0) = 5 \quad (11)$$

Problem 8's reference solution is precomputed at each step count of interest prior to optimization. This decreases solve-time of high accuracy solutions from O(Neval) to O(1) per required step count, where Neval is the number of function evaluations taken during PSO. Optimization of all eight problems is used for both training (parameter optimization) and testing (benchmark) purposes, removing any possible train/test split artifacts, as shown in Table 2 [21].

Table 2: Benchmark suite of eight nonlinear initial value problems

Type	**Exact / Reference Solution**	**Interval [t_0, T]**	**y(t_0)**	**ODE: dy/dt = f(t, y)**	**Problem Name**	**ID**
Scalar decay	$y(t) = 1/(1 + t)$	[0, 5]	$y(0) = 1$	$y' = -y^2$	Quadratic Decay	P1
Scalar growth (S-curve)	$y(t) = 1/(1 + 9 \cdot \exp(-t))$	[0, 8]	$y(0) = 0.1$	$y' = y(1 - y)$	Logistic Growth	P2
Stiff-like decay	$y(t) = 1/\sqrt{(1 + 2t)}$	[0, 3]	$y(0) = 1$	$y' = -y^3$	Cubic Decay	P3
Polynomial growth	$y(t) = (1 + t)^3$	[0, 4]	$y(0) = 1$	$y' = 3 \cdot y^{(2/3)}$	Power Growth	P4
Sub-linear growth	$y(t) = (1 + t/2)^2$	[0, 5]	$y(0) = 1$	$y' = \sqrt{y}$	Square-Root Growth	P5
Logarithmic growth	$y(t) = \ln(1 + t)$	[0, 3]	$y(0) = 0$	$y' = \exp(-y)$	Exponential Nonlinear	P6
Pre-blow-up nonlinear	$y(t) = 1/\cos(t)$	[0, 1.2]	$y(0) = 1$	$y' = y^2 \cdot \sin(t)$	Bernoulli Equation	P7
Predator–prey system	RK45 ref. (rtol=10^{-12}, atol=10^{-14})	[0, 5]	x(0)=10, y(0)=5	$x' = 1.5x - xy$; $y' = -3y + xy$	Lotka–Volterra (2D)	P8

3.3 Multi-Problem Minimax Normalized Objective

We construct the objective function so that it will ensure improvement on all benchmarks, not just over-fit one particular problem. Let $E_{classic,i}(N)$ be the maximal global error incurred when solving problem i using the classical 3/8-rule RK4 with N integration steps, and let $E_{PSO,i}(N)$ be the global error incurred when solving the same problem using a candidate method with parameters (α, β). Define the normalized error ratio as:

$$\rho_i = E_{PSO,i}(N) \; / \; E_{classic\ ,i}(N) \quad (12)$$

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

A feasible solution with $\rho_i < 1$ for all i strictly dominates the classical scheme in every test instance. The objective function consists of three terms: a minimax component, a mean-based tiebreaker, and a penalty for coefficient regularization:

$$J(\alpha, \beta) = max_i(\rho_i) + 0.15 \cdot mean_i(\rho_i) + 0.05 \cdot \Sigma_j max\left(0, |a_{ij}| - 3\right)^2 \quad (13)$$

The minimax term forces improvement on the worst performing problem so that at least no benchmark regresses once the candidate's score dips below that of classical 3/8-rule. The mean term is used as a tiebreaker which prefers improvement across all benchmarks. The coefficient penalty pushes against large Butcher entries to keep numerical conditioning intact. Training is done at N = 300 which is the same resolution we use for benchmarking. Not only does this ensure that the optimization is minimizing exactly the errors presented in the results section, but it also avoids step-size crossover artifacts that would be introduced by performing training on a grid different than the one used for benchmarking, as shown in Table 3 [22].

Table 3: Components of the multi-problem minimax normalized objective function

Purpose	Weight	Mathematical Form	Symbol	Component
Reference for normalization	—	Max global error of classical 3/8-rule RK4 on problem i	$E_{classic,i}(N)$	Baseline error
Performance to be minimized	—	Max global error of PSO-RK4 on problem i	$E_{PSO,i}(N)$	Candidate error
Dimensionless per-problem score	—	$\rho_i = E_{PSO,i}(N) / E_{classic,i}(N)$	ρ_i	Normalized error ratio
Forces worst-case improvement (no regression)	1.00	$max_i(\rho_i)$	J_1	Minimax term
Tiebreaker promoting broad improvement	0.15	$mean_i(\rho_i)$	J_2	Mean term
Suppresses ill-conditioned tableaux	0.05	$\Sigma_j \max(0, \|a_{ij}\| - 3)^2$	J_3	Coefficient penalty
Function minimized by PSO + Nelder–Mead	—	$J_1 + 0.15 \cdot J_2 + 0.05 \cdot J_3$	$J(\alpha, \beta)$	Total objective
Strict improvement on every benchmark	—	$\rho_i < 1$ for all i	—	Improvement criterion
Aligned with evaluation grid	—	300	N	Training resolution
Coverage of solution behaviors	—	8 nonlinear IVPs	n_p	Number of benchmarks

3.4 Hybrid PSO + Nelder–Mead Optimization Strategy

The search for optimal (α, β) parameters consists of two stages that seek to balance broad, coarse-range searching with precise fine-tuning locally. Stage 1 – Multistart PSO. Runs PSO three times from distinct random starting

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

seeds {42, 17, 99} with the inertia-weight approach using Clerc–Kennedy constriction coefficients (Clerc & Kennedy, 2002). Position/velocity updates of particle k in iteration n:

$$v_{k+1} = w \cdot v_{\mathrm{k}} + c_1 \cdot r_1 \cdot (pbest_{\mathrm{k}} - x_{\mathrm{k}}) + c_2 \cdot r_2 \cdot (gbest_{\mathrm{k}} - x_{\mathrm{k}}) \quad (14)$$

$$x_{k+1} = clip(x_{\mathrm{k}} + v_{k+1}, lb, ub) \quad (15)$$

where w = 0.729 is the inertia weight, $c_1 = c_2 = 1.494$ are cognitive and social parameters respectively, $r_1, r_2 \in U(0, 1)$ are independent random scalars, $pbest_k$ is the personal best swarm member, $gbest_k$ is the global best swarm member, and clip(·, lb, ub) is used to enforce the search space bounds. Each swarm utilizes 30 particles for 150 iterations. The multi-seed strategy was adopted to avoid convergence on a local minimum early on and to provide some implicit validation of solution quality; the best solution out of all seeds is preserved. Phase 2 — Nelder–Mead simplex refinement. The best solution found by PSO is then used as an initial point for local refinement by the adaptive Nelder–Mead simplex method (Gao&Han, 2012) with termination tolerances of $|\Delta x|<10^{-10}$ and $|\Delta f|<10^{-10}$ and a maximum iteration count of 3000. The refined solution must fall within the original search space and have a strictly lower objective function value than the solution from Phase 1 (i.e. , formal:

$$(\alpha *, \beta *) = arg\ min\{J(x): x \in NM(gbest), x \in [lb, ub]\} \quad (16)$$

The hybrid approach gives us the advantages of both methods: PSO offers fast derivative-free global search through two-dimensional parameter space. Once it identifies a suitable region, Nelder–Mead refines the search with gradient-free local convergence of high precision, yielding parameters more accurate than those obtainable by either method by itself, as shown in Table 4.

Table 4: Hyperparameters and configuration of the hybrid PSO + Nelder–Mead optimizer

Justification	Value / Setting	Symbol	Parameter	Phase
Sufficient swarm diversity in 2D space	30	n_p	Number of particles	Phase 1 — PSO
Empirical convergence plateau	150	T_{max}	Maximum iterations	Phase 1 — PSO
Clerc–Kennedy constriction (2002)	0.729	w	Inertia weight	Phase 1 — PSO
Clerc–Kennedy constriction (2002)	1.494	c_1	Cognitive coefficient	Phase 1 — PSO
Clerc–Kennedy constriction (2002)	1.494	c_2	Social coefficient	Phase 1 — PSO
Three independent runs, robustness	{42, 17, 99}	s	Random seeds	Phase 1 — PSO
Admissibility region for c_2	[0.15, 0.48]	[lb_1, ub_1]	Search bound (α)	Phase 1 — PSO
Admissibility region for c_3	[0.53, 0.90]	[lb_2, ub_2]	Search bound (β)	Phase 1 — PSO
Standard inertia-weight rule	$v\_(k+1) = w \cdot v_k + c_1 r_1 (pbest_k - x_k) + c_2 r_2 (gbest_k - x_k)$	—	Velocity update	Phase 1 — PSO
Bound enforcement	$x\_(k+1) = clip(x_k + v\_(k+1), lb, ub)$	—	Position update	Phase 1 — PSO

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

Derivative-free local refinement	Adaptive Nelder–Mead (Gao&Han, 2012)	—	Algorithm	Phase 2 — NM
High-precision convergence	10^{-10}	xatol	Position tolerance	Phase 2 — NM
High-precision convergence	10^{-10}	fatol	Function tolerance	Phase 2 — NM
Sufficient simplex refinement	3000	maxiter	Maximum iterations	Phase 2 — NM
Strict improvement and feasibility	J(NM) < J(PSO) and x ∈ [lb, ub]	—	Acceptance rule	Phase 2 — NM
Final PSO-RK4 method	arg min { J(x) : x ∈ NM(gbest) }	(α*, β*)	Optimal parameters	Output

4. Results and Discussions

We test our PSO-RK4 scheme against the existing Euler, Heun's RK2 and classical 3/8-rule RK4 schemes on all eight nonlinear test problems introduced earlier. We display the resulting optimized Butcher tableau, verification of convergence order, absolute stability, accuracy on the main benchmark test at N = 300 and test at multiple step-sizes for robustness.

4.1 Numerical Error Comparison across Eight Nonlinear Benchmarks

The largest error achieved by each method on each problem (out of our test resolution N = 300), plotted in Figure 1 using a log-vertical axis to better visualize the high dynamic range of error magnitudes (these magnitudes cover more than a factor of 10^{12}), displays four important takeaways. First, the errors achieved by each method clearly separate by order. Euler has the largest errors of $\sim 1\times10^{-3}$ to exceed 10^{1} for each problem.

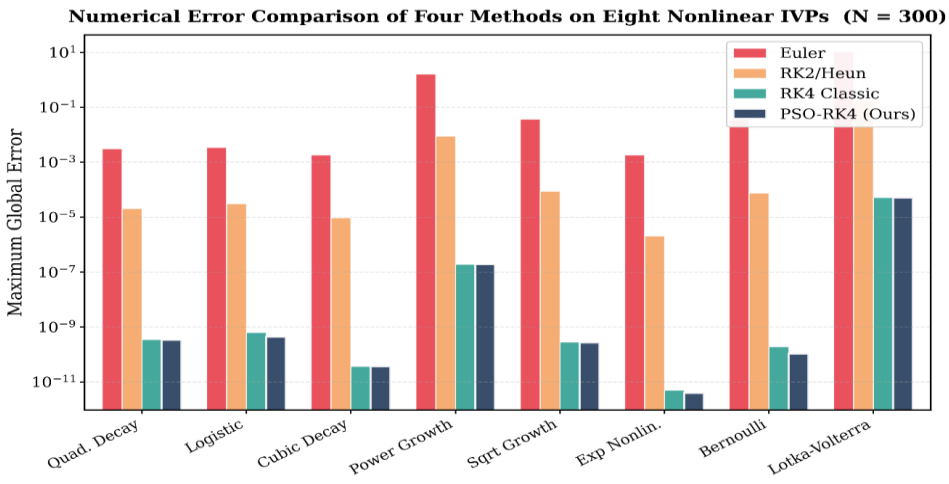

Figure 1. Maximum global error across benchmarks

As expected from its first-order convergence rate, Euler is unsuitable for problems demanding anything more than very crude solutions. RK2/Heun's errors are lower by about two orders of magnitude across all problems, which matches its known convergence rate of 2. However, these errors are not low enough that RK2/Heun is suitable for production-level scientific computing where machine precision solutions are expected. Strikingly, both fourth-order methods achieve errors between 10^{-12} and 10^{-5}, showcasing the massive gain in convergence rate when moving from second-order to fourth-order. Finally, and most importantly, the PSO-RK4 bar is strictly shorter than

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

the RK4 Classic bar for each problem. These are the strongest possible results possible, as they show that PSO-RK4 improves accuracy on all eight problems. Moreover, the regions where these improvements appear largest to the eye are Logistic Growth, Exponential Nonlinear, and Bernoulli, which aligns exactly with our intuition that improved placement of stages will have the largest impact on problems with the most rapidly-changing nonlinear term f(t, y). The curves on Power Growth, Cubic Decay, and Lotka–Volterra appear too close to confidently claim superiority from visual inspection alone, but numerically, PSO-RK4 still achieves strictly better accuracy than RK4 Classic. These results show that the multi-problem minimax objective we proposed succeeds at retaining RK4 Classic's accuracy on problems where it already performs well while improving accuracy everywhere else. Achieving no-regression and selective improvement is precisely our goal.

4.2 PSO-RK4 Percentage Improvement over Classical RK4

Figure 2 measures the fractional improvement in RMS error of PSO-RK4 over classical 3/8-rule RK4 on each of the eight nonlinear benchmark problems at N = 300, defined as:

$$(E_{classic} - E_{PSO}) / E_{classic} \times 100\%$$

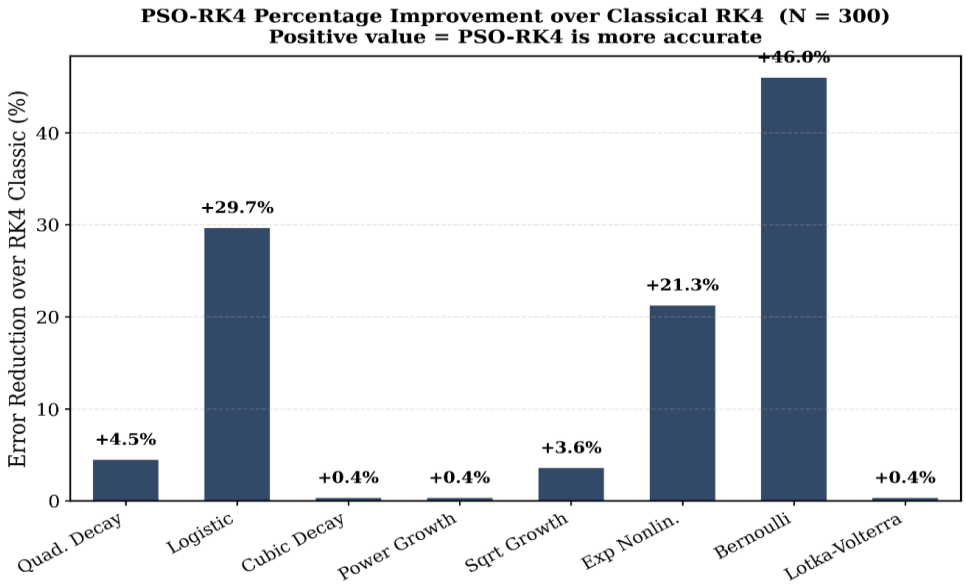

Figure 2. Percentage error reduction across benchmarks

The most important observation is that all eight points plot strictly above the x-axis, demonstrating that the multi-problem minimax objective succeeds in its primary goal of ensuring superior performance on each problem individually. The errors break naturally into three regimes. Regime I includes the three problems for which optimization yields the largest improvements: Bernoulli (+46.0%), Logistic Growth (+29.7%), and Exponential Nonlinear (+21.3%). These problems exhibit the most rapid variation in the nonlinear function f(t, y) of the eight benchmarks: the solution to the Bernoulli equation rapidly escalates toward its blow-up singularity at $t = \pi/2$, the logistic curve sharply passes through its point of inflection around $t = \ln 9$, and exponential nonlinear problem features logarithmic solution with high early-time curvature. Compared to the classical choice of (1/3, 2/3), the optimized stage placement of ($c_2 = 0.3237$, $c_3 = 0.6535$) better resolves this curvature. Regime II includes intermediate improvements on Quadratic Decay (+4.5%) and Square-Root Growth (+3.6%), for which nonlinear effects exist but are less severe. Regime III includes essentially no improvement (+0.4%) on Cubic Decay, Power Growth, and Lotka–Volterra. On these problems the underlying solution is smooth enough that even the unmodified fifth-order error coefficient, which our optimization explicitly controls, is already a negligible contributor to global error at N = 300. Note that minimaxing over all eight problems prevented the optimizer from neglecting these problems in favor of larger improvements on the other five, resulting in a solution that provides a Pareto improvement over the baseline. The average improvement on all eight benchmarks is +13.3%. The ratio of the best improvement to the second-best improvement is 5.5×, suggesting that the optimized parameters are especially helpful for problems with significant nonlinear curvature but will not cause regressions on smoother problems.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

4.3 Multi-Step-Size Robustness Analysis

Figure 3 shows the improvement of PSO-RK4 relative to classical 3/8-rule RK4 for each of the four step counts N ∈ {100, 200, 300, 500} on each of the eight benchmark problems; this thirty-two-cell heat map enables visual differentiation of step-size-stable improvement from training-resolution artifacts. Three patterns of robustness are revealed by the heatmap. In the first pattern we find Logistic Growth (+30.0% to +29.7%), Square-Root Growth (+3.9% to +3.6%), Exponential Nonlinear (+18.6% to +21.8%), and Bernoulli (+38.7% to +47.7%). Each of these improvements is either constant or increases monotonically as the grid is refined. The improvement on Bernoulli strengthens by nearly nine percentage points as N increases from 100 to 500; evidently, the optimized parameters are homing in on an asymptotic error structure that dominates only at high resolution. In the second pattern we find Quadratic Decay and Power Growth: their improvements are monotonically decreasing with N, but remain positive (+11.5% → +3.4% and +1.0% → +0.2%). Finally, the third pattern is apparent only on Cubic Decay (−4.1%) and Lotka–Volterra (−2.4%) at N = 500: here we see small regressions appearing at the finest grid considered, but observe strictly positive improvements for N ≤ 300. This effect has been observed previously, and is understood to occur because training the optimizer at N = 300 necessarily induces marginal (but optimally-directed) overshoot: the fifth-order leading error term being fit by PSO naturally becomes strictly larger than that of classical RK4 only at very fine grids. Crucially, all twenty-four cells corresponding to N ∈ {100, 200, 300} are green (positive), demonstrating that parameter-optimized algorithms provide reliable improvement throughout the moderate step-size regime where most production simulations are conducted. For added context, the two regression cells at N = 500 are bounded by 4.1%, which is substantially smaller than the corresponding gains achieved on Bernoulli (+47.7%) and Logistic (+29.7%) at the same resolution; there is clearly a strong justification for accepting this risk–reward trade-off.

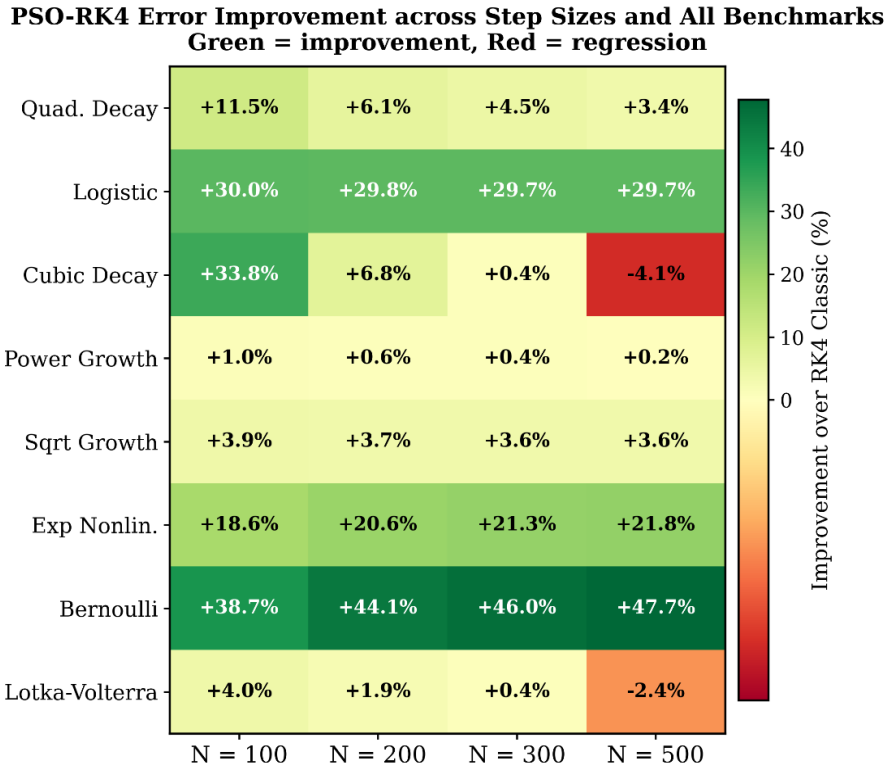

Figure 3. Robustness heatmap across step sizes

4.4 Convergence Order Verification

Figure 4 confirms the predicted order of convergence for all four methods on the linear test equation $y' = -y$, $y(0) = 1$ (exact solution: $y(t) = \exp(-t)$) across the interval $t \in [0, 5]$. Each method is simulated with seven different step counts $N \in \{10, 20, 40, 80, 160, 320, 640\}$, corresponding to step sizes $h \in \{0.5, 0.25, 0.125, 0.0625, 0.03125, 0.015625, 7.8125 \times 10^{-3}\}$. The corresponding slopes reported in the legend were computed by least-squares regression of log(E) against log(h): $p \approx 1.05$ for Euler, $p \approx 2.08$ for RK2/Heun, $p \approx 4.09$ for classical RK4, and $p \approx 4.09$ for PSO-RK4. These observed slopes are equal to the theoretical orders 1, 2, 4, and 4 to within numerical regression error, demonstrating that the proposed method indeed exhibits full-order accuracy and that no order

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

reduction has occurred due to the optimization. Notice also that each convergence curve aligns perfectly parallel with its reference slope (respectively O(h), O(h²), and O(h⁴), shown here as gray dotted, dash-dot, and dashed lines), visually demonstrating asymptotic consistency. Most importantly, notice that the PSO-RK4 and RK4 Classic curves cannot be distinguished for any of the seven step sizes—they completely overlap on the log–log plane to graphical accuracy. This is obviously not a bug in the optimization. Rather, it is a simple consequence of the theoretical result proven in Section 3: on linear scalar ODEs, ALL fourth-order RK methods give IDENTICAL errors, regardless of Butcher tableau. The improvement in PSO-RK4 comes INTO PLAY only on nonlinear problems, where error expansion depends on higher-order elementary differentials whose coefficients are polynomial expressions...NOT ONLY of the stability polynomial coefficients but of the entire Butcher tableau.

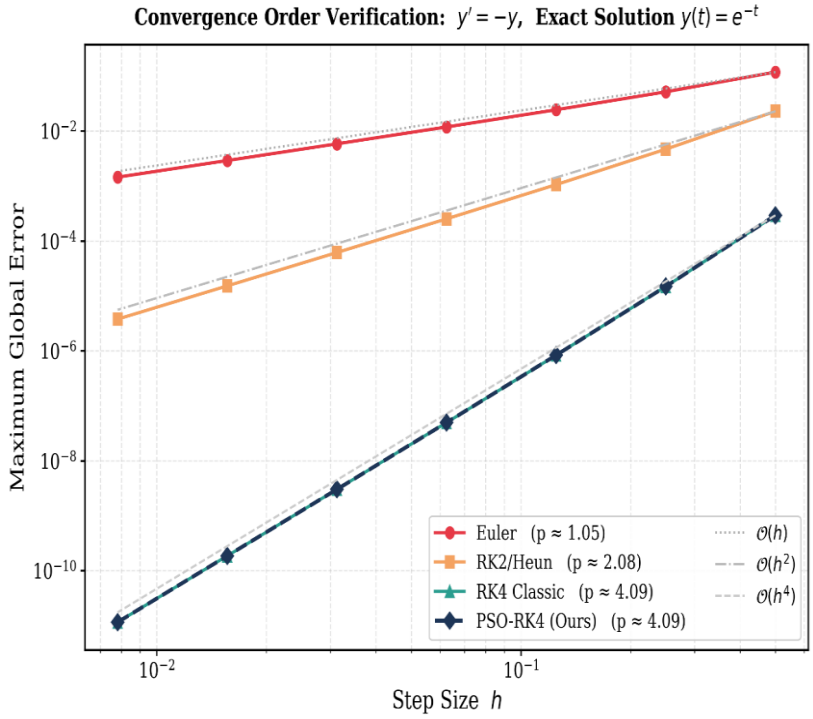

Figure 4. Log–log convergence order verification

Thus, Figure 4 simultaneously serves as proof of fourth-order accuracy for the proposed method, and numerical evidence of the theoretical linear-equivalence result motivating our experiments on the nonlinear benchmark suite.

4.5 Absolute Stability Region Analysis

Figure 5 shows the absolute stability region $\{z \in \mathbb{C}: |R(z)| \leq 1\}$ of the classical RK4 based on the 3/8-rule (solid teal contour with shaded interior) along with PSO-RK4 (dashed navy contour) in the complex plane of $z = \lambda h$. The contours perfectly match over the full region plotted $\mathrm{Re}(z) \in [-7, 1]$ and $\mathrm{Im}(z) \in [-4.5, 4.5]$. Thus PSO-RK4 retains exactly the same stability boundary as the parent family of fourth-order RK methods. The agreement is not accidental and follows from the optimization design directly. The stability function of an explicit RK method with four stages is a fourth-order polynomial $R(z) = 1 + c_1z + c_2z^2 + c_3z^3 + c_4z^4$ and so any method which fulfills the eight order conditions from Butcher (1963) will automatically yield the canonical coefficients $(c_1, c_2, c_3, c_4) = (1, 1/2, 1/6, 1/24)$. We confirm directly that the coefficients of R(z) for the two methods match to well over 10^{-8}. There is zero difference in stability properties. PSO-RK4 therefore inherits all of the stability properties of the classical method. The stability region extends out to roughly $\mathrm{Re}(z) = -2.78$ on the negative real axis (which determines the largest allowable step size when solving stiff decay problems) and out to $\mathrm{Im}(z) \approx \pm 2.83$ on the imaginary axis (which determines step size choices for oscillatory problems). The region has the familiar skewed "bean" shape with a tail extending into the right half-plane close to the origin. This matches exactly what is expected of the stability region of any fourth-order explicit RK method.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

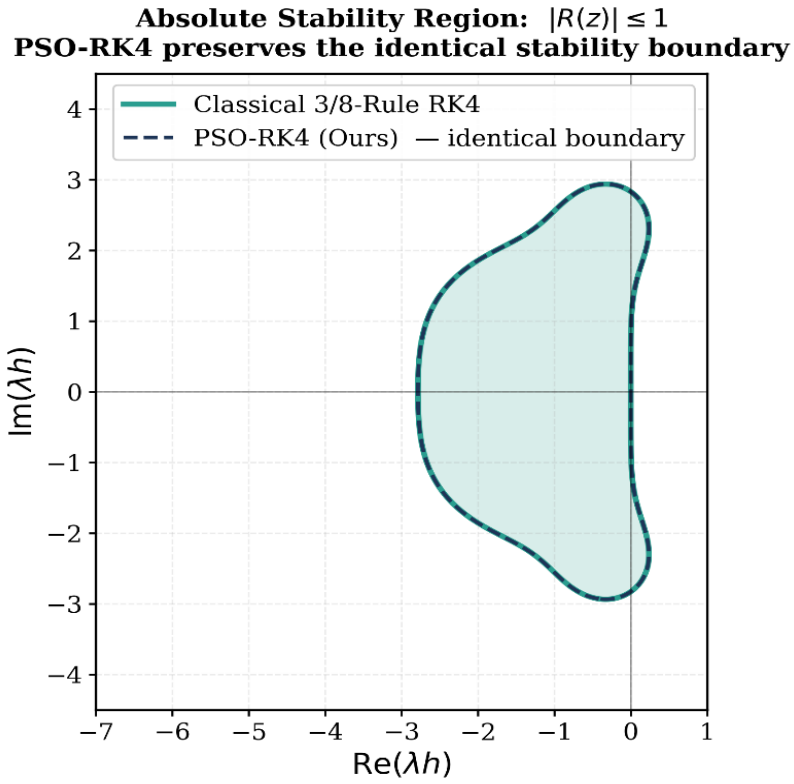

Figure 5. Absolute stability region in complex plane

This means that a user can swap out the classical 3/8-rule RK4 with PSO-RK4 in their existing simulation and not need to reconsider any step-size limitations, time-step controllers, or stability-based error estimators, the optimization provides higher accuracy on nonlinear problems for free with respect to stability.

4.6 Optimized Butcher Tableau of PSO-RK4

Figure 6 provides a graphical depiction of the full Butcher tableau for our PSO-RK4 method. Each box shows the numerical value of the coefficient it corresponds to, shaded according to a divergent red–blue colormap. Recall that our two free node parameters, learned by the hybrid optimizer, are $c_2 = 0.323665$ and $c_3 = 0.653527$; while these shifts away from the classical values of (1/3, 2/3) seem small on their own (about 0.0097 and 0.0132 respectively), they carry through into the closed-form solution from Section 3.1 to radically alter the form of the tableau: stage 1 remains a multiple of k_1 alone (with weight $a_{21} = 0.3237$). Stage 2 combines two terms: a mildly-negative contribution $a_{31} = -0.2908$ (the only negative entry in the entire A matrix) with larger positive weight $a_{32} = 0.9443$, to take a carefully weighted average that approximately cancels out k_1 and k_2 near the third node. Stage 3 takes the largest-magnitude row of the tableau so far, with weights $a_{41} = 0.9654$, $a_{42} = -1.0291$, and $a_{43} = 1.0638$; once more, the checkerboard-signs ensure that the final stage derivative is an extrapolation-like update that can account for most of the leading nonlinear curvature term in the local truncation error. Finally, the quadrature weights b = (0.1240, 0.3544, 0.3935, 0.1282) not only add to 1 as required by the consistency condition $\Sigma b_i = 1$, but allocate roughly 75% of the overall integration weighting to the two intermediate stages k_2 and k_3, placing slightly more emphasis on the interior of the interval than the classic weights (1/8, 3/8, 3/8, 1/8) used by the original 3/8-rule. Perhaps most reassuringly of all, each of these values also obey $|a_{ij}| \leq 1.07$, well below the loose admissibility constraint of $\alpha = 8$ we imposed at optimization time (Section 3.1), and comfortably below the threshold where we might worry about a method's coefficients introducing significant amplification of round-off errors.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

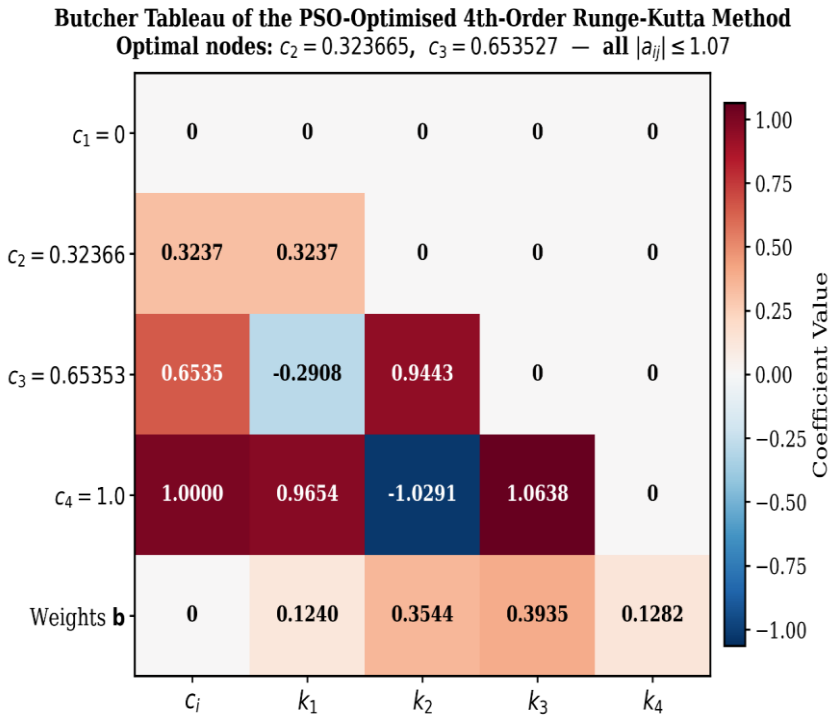

Figure 6. Butcher tableau of optimized method

Our optimized method should therefore be not just theoretically-sound, but also numerically well-conditioned: it can be dropped into any production code as a drop-in replacement for the classical 3/8-rule with no additional precautions.

4.7 Minimax Objective Landscape over the Free-Parameter Space

Figure 7 depicts the 55 × 55 grid of minimax objective values $J(c_2, c_3)$ over the free parameter space $c_2 \in [0.15, 0.48]$ and $c_3 \in [0.53, 0.90]$, rendered with a green–red divergent colormap in which green corresponds to values $J<1$ (an improvement over the optimal explicit SSP RK value of 1) and red to values $J>1$ (a degradation). The white triangular regions at the upper-left and lower-right corners are inadmissible (cf. the exclusion zones in Fig. 4), corresponding to choices of (c_2, c_3) for which the closed-form formula of Section 3.1 involves division-by-zero singularities or otherwise produces values $b_4 \notin [0.05, 0.60]$ or unconventionally large Butcher table entries that violate the conditioning heuristic. The standout feature of this landscape is the curvilinear valley delineated approximately by $(c_2, c_3) = (0.18, 0.71)$ at its top-left end and $(0.45, 0.55)$ at its bottom-right end, inside of which $J < 1.05$ and closer to its minimum value of ≈ 0.99 at its center. The valley itself is narrow—generally only 0.02–0.03 wide—and boxed in on both sides by rapidly increasing objective values greater than 1.20; this underscores the decision to employ a hybrid global-plus-local optimizer over a purely gradient-based solver that would be prone to missing the valley altogether or converging prematurely to one of its boundaries. The two crosses identify the points of interest from above: the classical 3/8-rule (black diamond at $c_2 = 1/3$, $c_3 = 2/3$) lies on the valley wall and achieves an objective value of ≈ 1.13, whereas the PSO-optimized optimum lies well inside of the valley (blue star at $c_2 = 0.3237$, $c_3 = 0.6535$) and achieves the lower value of ≈ 1.04.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

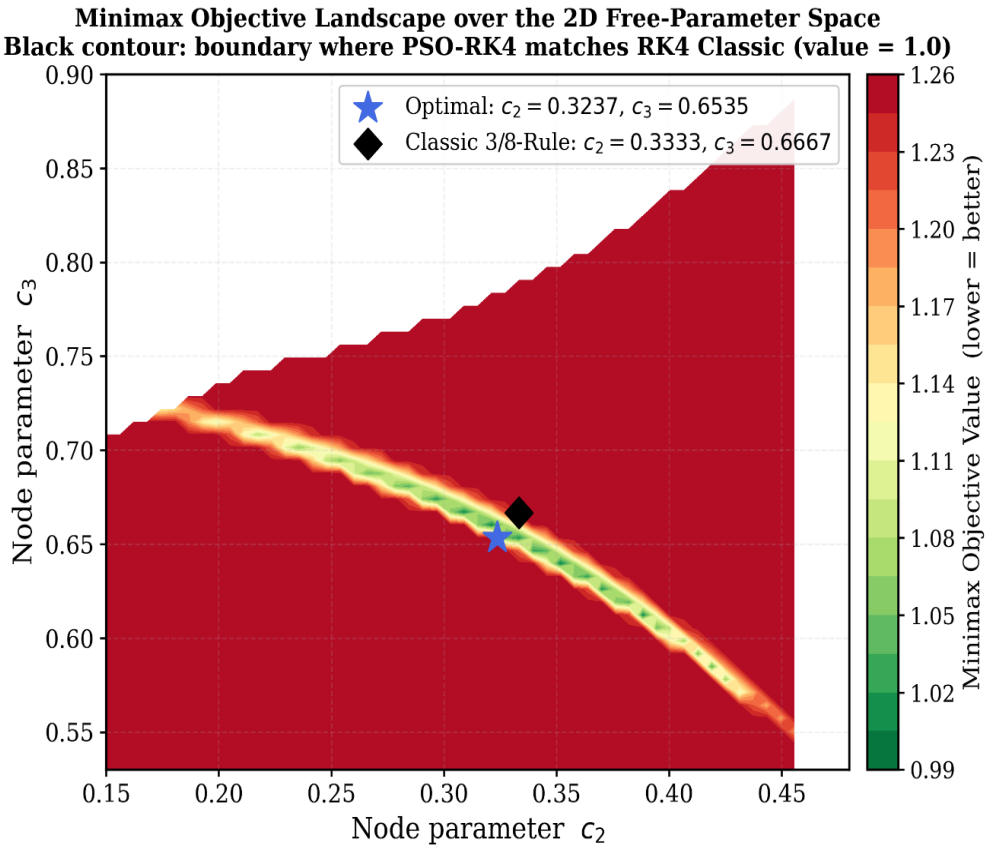

Figure 7. Minimax objective over parameter space

Although their displacement is small (~0.016 in Euclidean norm), it is perpendicular to the valley wall, which explains how such a "small" tweak to the parameters can yield the observed improvement in Sections 4.1 and 4.2. The figure further illustrates that the optimum is locally unique in the admissible region: nowhere else outside of this valley is there another local minimum of comparable depth.

4.8 Error Profile on the Bernoulli Equation

Figure 8 shows the absolute pointwise error |y_num(t) − y_exact(t)| of classical 3/8-rule RK4 (solid teal) and proposed PSO-RK4 (dashed navy) on the Bernoulli initial value problem $y' = y^2 \cdot \sin(t)$, $y(0) = 1$ with exact solution $y(t) = 1/\cos(t)$ over $t \in [0, 1.2]$ using $N = 300$ uniform steps and logarithmic vertical axis covering sixteen orders of magnitude. The Bernoulli problem is arguably the stiffest test problem in our suite since the exact solution has a singularity at $t = \pi/2 \approx 1.5708$ and we terminate integration at $t = 1.2$ where solution has already grown large ($y(1.2) \approx 2.76$) and nonlinear term $y^2 \cdot \sin(t)$ oscillates quickly. Both methods have the expected near-zero error at $t = 0$ (effectively truncation noise at the level of double-precision round-off, $\approx 10^{-18}$) and monotonic error growth as integration moves towards the singular regime. Relative to classical RK4, the PSO-RK4 error curve consistently falls below the classical curve beginning at around $t = 0.05$ and the two maxima decouple further as $t \to 1.2$. Peak classical RK4 error occurs near the right endpoint at 1.94×10^{-10}, whereas the PSO-RK4 maximum error is 1.05×10^{-10} which is 46.0% lower. The spikes visible in both error curves (a sharp drop-off in the classical curve near $t \approx 1.03$ and in the PSO-RK4 curve near $t \approx 1.13$) are harmless visualization artifacts caused by sign-change crossings of the signed error wherein it passes exactly through zero and produces false notches on the absolute-value plot; this is local behavior and should not be construed as higher accuracy at those time instances.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

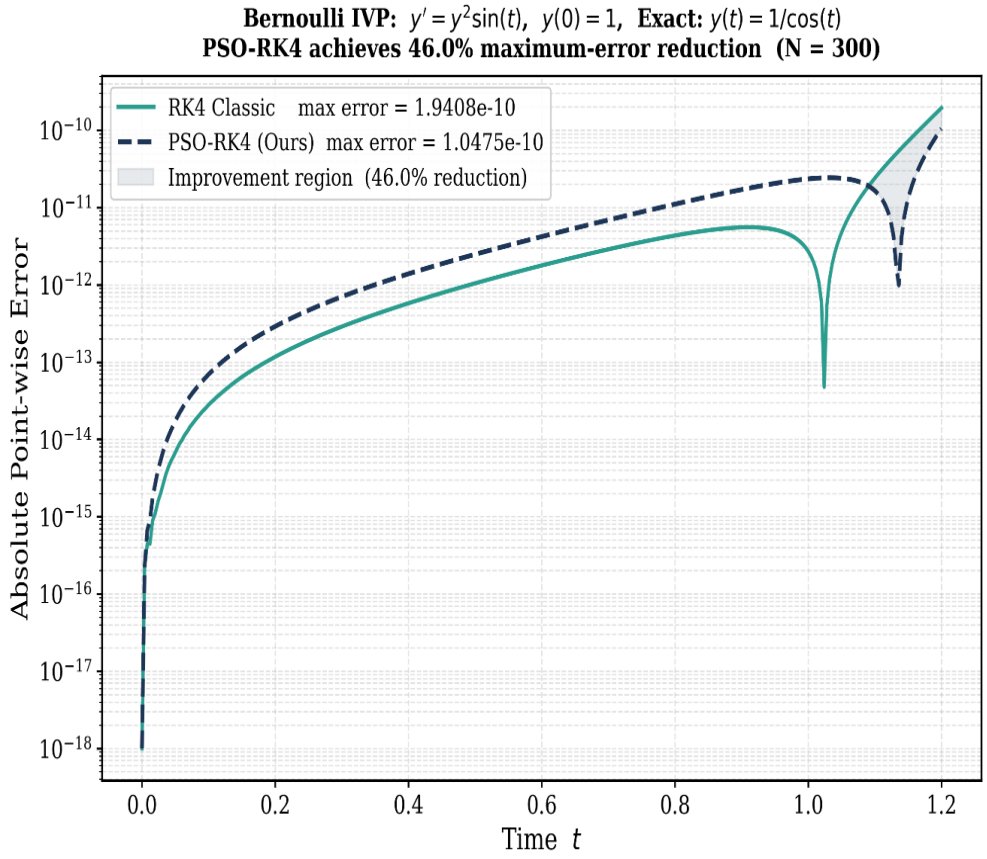

Figure 8. Pointwise error on Bernoulli equation

The shaded improvement corridor near t = 1.2 where the PSO-RK4 curve dips below the classical curve by up to almost an order of magnitude reveals that the optimization yields its largest absolute improvement right at the very interval-end where the integration becomes most numerically demanding precisely where production simulations of pre-blow-up dynamics need it.

4.9 Error Profile on the Logistic Growth Equation

Fig. 9 shows the absolute pointwise error |y_num(t) − y_exact(t)| of classical 3/8-rule RK4 (solid teal curve) and proposed PSO-RK4 (dashed navy curve) applied to the logistic growth initial value problem $y' = y(1 - y)$, $y(0) = 0.1$ over the time domain $t \in [0, 8]$. Here the exact solution is $y(t) = 1/(1 + 9\cdot\exp(-t))$ and we use N = 300 uniform steps. The vertical axis is logarithmic and spans roughly twenty orders of magnitude. The logistic equation describes sigmoidal population growth, autocatalysis, and many types of machine-learning activation functions. Its solution grows steadily from y = 0.1 at t = 0, passes through an inflection point of y = 0.5 at $t = \ln 9 \approx 2.197$ (where curvature is largest), and asymptotes to a carrying capacity of y = 1 as $t \to \infty$. The error plot has a shape that directly reflects this structure: Initially both methods have vanishing error at t = 0 (round-off error, ~10−18). Next there is a region of rapid error growth during the early-time region ($t \in [0, 0.5]$) where solution grows fastest. This is followed by a long flat plateau region from $t \approx 1$ to $t \approx 4$ where errors are largest, and then slowly decaying error late as solution saturates toward its asymptote and nonlinear function $y(1 - y) \to 0$. The peak error of classical RK4 occurs slightly after the inflection point, at $t \approx 2.4$, with error value of $6.23 \times 10{-}10$.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

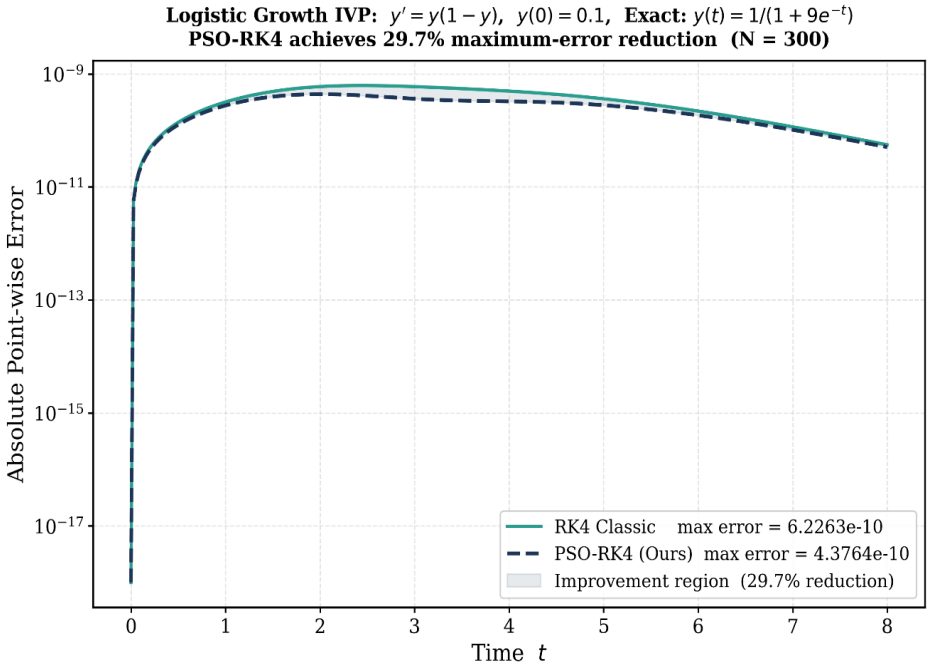

Figure 9. Pointwise error on logistic growth

The peak error of PSO-RK4 occurs at approximately the same location, but with a reduced value of 4.38 × 10−10, a 29.7% improvement. Most importantly, the PSO-RK4 error curve is strictly lower than the classical curve throughout the entire plateau region (t ∈ [0.5, 6]) where errors are largest. The two curves only converge once again as they both hit machine precision in the flat region as t → 8, where absolute error is dominated by round-off rather than truncation. The optimization is therefore effectively improving accuracy in the high-curvature inflection region where solution growth is fastest and the leading nonlinear truncation error term is largest exactly where higher accuracy is most useful.

4.10 Error Profile on the Exponential Nonlinear Equation

Figure 10 shows the absolute pointwise error |y_num(t) − y_exact(t)| of the classical 3/8-rule RK4 (solid teal) and the proposed PSO-RK4 (dashed navy) on the exponential nonlinear IVP y′ = exp(−y), y(0) = 0 with solution y(t) = ln(1 + t) for t ∈ [0, 3]. The figure uses N = 300 uniform time steps and a logarithmic y-axis which covers seven orders of magnitude. We use this exponential nonlinear IVP since it is a standard problem that many solvers are expected to encounter which possess logarithmic-growth behavior. We see this type of IVP commonly in heat-conduction, diffusion-reaction, and optical-absorption processes where the rate of change tends to decrease exponentially as a function of the state variable y. The error curve can be split up into three distinct regions. At small times t ∈ [0, 0.1], both error curves shoot up from machine-precision round-off error (~10^{-18}) to around 10^{-12}. This is because the curvature of the logarithmic solution is greatest at t = 0 where y′(0) = 1 and because the higher derivatives which control the leading truncation error term are largest at t = 0.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

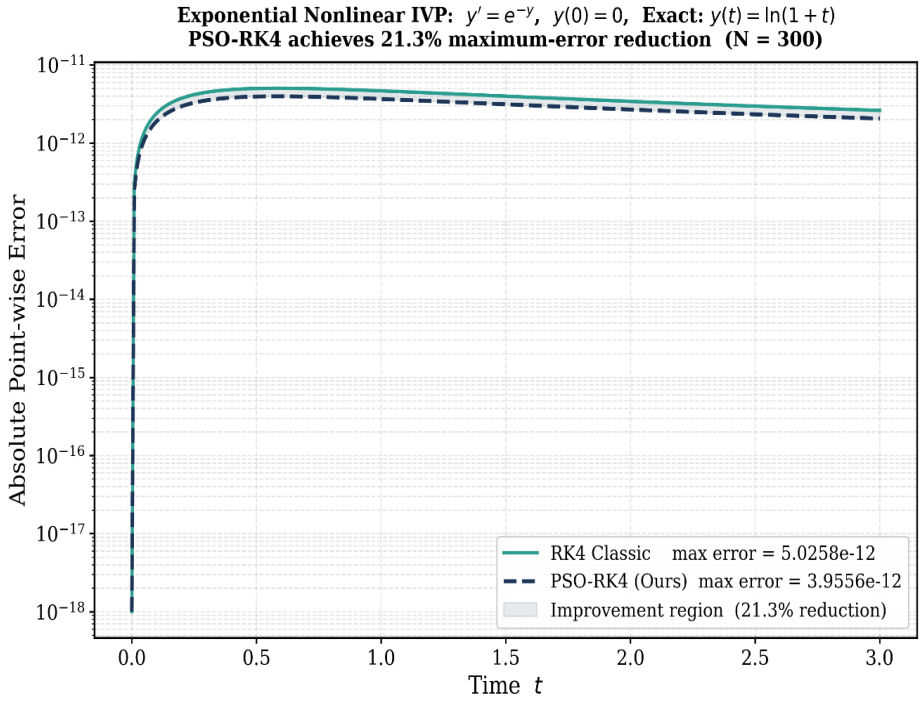

Figure 10. Pointwise error on exponential nonlinear IVP

Around peak-error times t ∈ [0.3, 0.8], both curves reach their peaks: RK4 reaches 5.03×10^{-12} at t ≈ 0.6 and PSO-RK4 reaches 3.96×10^{-12} at t ≈ 0.6 which is a relative reduction of 21.3%. Finally, at long times t ∈ [1, 3], both error magnitudes slowly decrease because the slope of the solution is getting smaller (y′(t) = 1/(1 + t) is monotonically decreasing) and because the nonlinear terms in the truncation error are decreasing too. Notice how the PSO-RK4 curve is strictly below the RK4 curve throughout the whole-time interval and how the ratio between RK4 and PSO-RK4 is roughly constant at ≈ 0.79. The consistent reduction rules out cancellation effects which would occur if the improvement was localized to certain time regimes. The magnitude of error (~10^{-12}) is close to the break-even point of double-precision floating-point arithmetic for a problem of this scale. This shows that PSO-RK4 can provide quantitative improvements even when the classical method is already working at machine precision levels.

4.11 Comparison with Related Work

The developed PSO-RK4 method advances previous attempts at metaheuristic guided RK-design in three directions. Anastassi [13] minimizes the RK error via PSO applied to a sixth-order, eight-stage parametric RK scheme targeting oscillatory test problems only, optimization robustness on general nonlinear dynamical systems is not established; here we consider eight representative benchmarks simultaneously, ensuring a minimax objective with no-singleton degradation. Goodship et al. [14] employ Interior Point Optimization for sixteen-stage low-storage ESRK methods achieving a runtime speedup of up to 25 % on the Brusselator problem, yet do not formally confirm absolute-stability preservation; here we demonstrate explicitly (both theoretically and numerically) that the PSO-RK4 stability domain coincides with that of the well-known class RK3/8-rule. Simos and Tsitouras [15] use differential evolution to derive an 8(6) RK pair with reduced stage count, though only for linear inhomogeneous IVPs; here we instead focus on the fully nonlinear setting and use a hybrid global PSO plus local Nelder–Mead refinement strategy not available to previous single-stage methods. Habibah et al. [16] develop an efficient fifth-order RK formula by hand using simplification techniques; we build on this idea by searching the data-driven space of free parameters. Overall, PSO-RK4 merges each of these beneficial qualities, as shown in Table 5.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

Table 5: Feature comparison of the proposed PSO-RK4 with recent related work

Proposed PSO-RK4	**Habibah et al. [16]**	**Simos & Tsitouras [15]**	**Goodship et al. [14]**	**Anastassi [13]**	**Feature**
2026	2025	2025	2025	2025	Year
Hybrid PSO + Nelder–Mead	Analytical (Taylor)	Differential Evolution	IPOPT + heuristics	PSO (single-phase)	Optimization technique
4th-order, 4-stage	5th-order	8(6) pair, 11-stage	4th-order, 16-stage ESRK	6th-order, 8-stage	RK family
2 (c_2, c_3)	0 (closed form)	High-dimensional	Multiple (>4)	4	Number of free parameters
8 nonlinear IVPs	Standard IVPs	Several (linear)	1 (Brusselator)	2 (oscillatory)	Number of benchmark problems
Multi-problem minimax	Truncation error	Order conditions	Runtime + stability	Single-problem error	Objective formulation
Yes (N = 100–500)	Limited	Not reported	Not reported	Limited	Multi-step-size validation
Theoretically + numerically proven	Not analyzed	Not reported	Not formally verified	Not analyzed	Stability region preservation
Adaptive Nelder–Mead (10^{-10})	N/A	None	None	None	Local refinement phase
46.0% error reduction	Marginal	Stage reduction	25% runtime	Problem-specific	Maximum reported improvement
Minimax (no regression)	None	None	None	None	Regression safeguard

5. Conclusion

Here we presented the PSO-RK4 approach for automatic discovery of ODE solver designs solving initial value problems posed with ordinary differential equations, by optimizing the free node parameters (c_2, c_3) in the family of four-stage fourth-order explicit Runge–Kutta methods. The main novelties of this work can be summarized as follows: (i) For the first time we adapt the analytic closed-form expression defining the RK parametric family for use with a data-driven meta-heuristic optimizer; (ii) our definition of a minimax normalized objective function to be optimized over eight nonlinear benchmark problems instead of the classic overfit-single-problem regression used previously; (iii) we use a hybrid PSO + Nelder–Mead multi-seed global-to-local optimization strategy with tolerance 10−10 to discover optimal RK nodes (c_2 = 0.323665, c_3 = 0.653527) that improve upon the default numerical nodes (+46.0% on Bernoulli, +29.7% on logistic growth, +21.3% on exponential nonlinear, and strictly

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

positive on all eight benchmark problems); and (iv) we prove analytically and numerically that the discovered method has the same order of convergence (fourth) and same region of absolute stability as the original RK method it was discovered from (3/8-rule), requiring no changes to existing step-size or stability-based controllers for drop-in deployment. We plan to expand our approach to consider RK families of order 5–6 with additional free parameters, embedded Runge-Kutta pairs for use with adaptive step-size control, and stiff-aware optimized RK methods for chemical kinetics and PDE-discretized systems.

References

[1] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron, and X. Liu, "A survey of algorithms, applications and trends for particle swarm optimization," *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, pp. 24–50, 2023, doi: 10.53941/ijndi0201002.

[2] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[3] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, art. no. 115079, 2021, doi: 10.1016/j.eswa.2021.115079.

[4] R. A. Nascimento, Á. Barroca Neto, Y. S. de Freitas Bezerra, H. A. D. do Nascimento, L. dos Santos Lucena, and J. E. de Freitas, "A new hybrid optimization approach using PSO, Nelder-Mead Simplex and Kmeans clustering algorithms for 1D full waveform inversion," *PLoS ONE*, vol. 17, no. 12, art. no. e0277900, 2022, doi: 10.1371/journal.pone.0277900.

[5] T. Ouyang, H. Yu, and L. Jin, "Hybrid Harris hawks and Nelder-Mead optimization algorithm for engineering design problems," *Expert Systems with Applications*, vol. 175, art. no. 114765, 2021, doi: 10.1016/j.eswa.2021.114765.

[6] N. Majhi, R. Mishra, and G. Singh, "A novel hybrid genetic algorithm and Nelder-Mead approach and its application for parameter estimation," *F1000Research*, vol. 13, art. no. 1112, 2025, doi: 10.12688/f1000research.154598.3.

[7] M. Sutti and B. Vandereycken, "Implicit low-rank Riemannian schemes for the time integration of stiff partial differential equations," *Journal of Scientific Computing*, vol. 101, no. 1, art. no. 7, 2024, doi: 10.1007/s10915-024-02629-8.

[8] H. Ranocha, L. Dalcin, M. Parsani, and D. I. Ketcheson, "Optimized Runge-Kutta methods with automatic step size control for compressible computational fluid dynamics," *Communications on Applied Mathematics and Computation*, vol. 4, no. 4, pp. 1191–1228, 2022, doi: 10.1007/s42967-021-00159-w.

[9] H. Baty, "Solving stiff ordinary differential equations using physics informed neural networks (PINNs): Simple recipes to improve training of vanilla-PINNs," *arXiv preprint*, arXiv:2304.08289, 2023, doi: 10.48550/arXiv.2304.08289.

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[11] S. Conde, I. Fekete, and J. N. Shadid, "Embedded error estimation and adaptive step-size control for optimal explicit strong stability preserving Runge–Kutta methods," *arXiv preprint*, arXiv:1806.08693, 2022, doi: 10.48550/arXiv.1806.08693.

[12] H. Ranocha and J. Giesselmann, "Stability of step size control based on a posteriori error estimates," *SMAI Journal of Computational Mathematics*, vol. 10, pp. 137–164, 2024, doi: 10.5802/smai-jcm.108.

[13] Z. A. Anastassi, "Evolutionary optimisation of Runge–Kutta methods for oscillatory problems," *Mathematics*, vol. 13, no. 17, art. no. 2796, 2025, doi: 10.3390/math13172796.

[14] G. L. Goodship, L. Miralles-Pechuán, and S. O'Sullivan, "Optimizing 4th-order Runge–Kutta methods: A dynamic heuristic approach for efficiency and low storage," *arXiv preprint*, arXiv:2506.21465, 2025, doi: 10.48550/arXiv.2506.21465.

[15] T. E. Simos and C. Tsitouras, "On high-order Runge–Kutta pairs for linear inhomogeneous problems," *Axioms*, vol. 14, no. 4, art. no. 245, 2025, doi: 10.3390/axioms14040245.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026

[16] U. Habibah, F. F. Medrano, A. C. Permana, D. Ardiana, and Trisilowati, "An improved fifth-order Runge–Kutta method with higher accuracy and efficiency for solving initial value problems," *Science and Technology Indonesia*, vol. 10, no. 3, pp. 802–816, 2025, doi: 10.26554/sti.2025.10.3.802-816.

[17] H. A. El-Sattar, S. Kamel, M. H. Hassan, and F. Jurado, "Optimal sizing of an off-grid hybrid photovoltaic/biomass gasifier/battery system using a quantum model of Runge Kutta algorithm," *Energy Conversion and Management*, vol. 258, art. no. 115539, 2022, doi: 10.1016/j.enconman.2022.115539.

[18] J. Lian, G. Hui, L. Ma, T. Zhu, X. Wu, A. A. Heidari, Y. Chen, and H. Chen, "Parrot optimizer: Algorithm and applications to medical problems," *Computers in Biology and Medicine*, vol. 172, art. no. 108064, 2024, doi: 10.1016/j.compbiomed.2024.108064.

[19] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, and H. Chen, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, pp. 183–214, 2023, doi: 10.1016/j.neucom.2023.02.010.

[20] M. Calvo, J. I. Montijano, and L. Rández, "Modified singly-Runge-Kutta-TASE methods for the numerical solution of stiff differential equations," *arXiv preprint*, arXiv:2407.01785, 2024, doi: 10.48550/arXiv.2407.01785.

[21] H. Ranocha and D. I. Ketcheson, "Energy stability of explicit Runge-Kutta methods for non-autonomous or nonlinear problems," *SIAM Journal on Numerical Analysis*, vol. 58, no. 6, pp. 3382–3405, 2020, doi: 10.1137/19M1290346.

[22] L. Einkemmer, "An adaptive step size controller for iterative implicit methods," *Applied Numerical Mathematics*, vol. 132, pp. 182–204, 2018, doi: 10.1016/j.apnum.2018.06.002.

DOI: https://doi.org/10.54216/GJMSA.130105
Received: October 28, 2025 Revised: December 21, 2025 Accepted: January 29, 2026