

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)



Improving Nonlinear Van der Pol Equation Numerical Solutions using a Hybrid GA-PSO Algorithm and Dynamic Adaptive Collocation Method

Ali Hamza Hussain

Ministry of Education, General Directorate of Education in Kirkuk Governorate, Kirkuk, Iraq. kingmath40@gmail.com

ARTICLE INFO

Article history:

Received: 14 /04/2026

Revised form: 28 /04/2026

Accepted : 29 /04/2026

Available online: 30 /09/2026

Keywords:

Van der Pol Equation, Hybrid GA-PSO, Metaheuristics, Dynamic Adaptive Collocation, Nonlinear ODEs, Computational Intelligence .

ABSTRACT

In this research, a novel hybrid framework, DAC-GA-PSO, is proposed to solve the nonlinear Van der Pol equation, which overcomes the stiffness and rapid oscillations of the equation by introducing the Dynamic Adaptive Collocation (DAC) mechanism, which treats collocation points as shifting choice variables that migrate toward high-gradient zones, combined with Genetic Algorithms (GA) for global exploration and Particle Swarm Optimization (PSO) for local exploitation. Simulation results show that this approach can greatly reduce the Mean Squared Residual (MSR) and improve numerical stability with fewer evaluation points and lower computational costs than fixed-grid methods.

MSC..

<https://doi.org/10.29304/jqcm.2026.18.32802>

1- Introduction

Nonlinear ODEs arise from mathematical modeling of complex dynamical systems, and are the most commonly used language to describe rate-dependent processes in physics, engineering, and medicine [7]. The Van der Pol equation is a classical benchmark for self-sustained oscillations and limit cycle behavior, and was originally developed to describe oscillations in vacuum tube circuits [2]. but has since been applied to a variety of other systems, such as aerodynamic instabilities, heart rhythms, and seismic activity [7,2]. This problem is inherently difficult to solve analytically due to its "stiffness" under high nonlinearity, when the solution quickly transitions between the slow and fast manifolds [1,2].

When analytical solutions are not possible, numerical integration methods such as Runge-Kutta, although limited by the fact that they are computationally expensive for stiff systems and may not necessarily provide a functional solution in continuous space [6], can be applied. Simulating nonlinear oscillatory systems is still an ongoing challenge, requiring sugar cube mental models that can be explained by current computational modeling, where heuristic processing techniques are needed to achieve stability [11]. Therefore, the scientific community is increasingly relying on metaheuristic optimization and computational intelligence to reformulate ODEs as functional minimization problems [4]. Genetic algorithms (GA) are essentially based on Darwinian evolution and can effectively escape from local minima and find global solutions [3]. Particle swarm optimization (PSO) algorithms are based on the behavior of flocks of birds and are particularly good at quickly finding a global optimal solution using local resources [5]. Ongoing development of nature-inspired algorithms provides reliable frameworks to tackle high-dimensional optimization problems that surpass the capability of traditional calculus-based methods [9]. Genetic algorithms and particle swarm optimization (GA-PSO) have been combined to take advantage of the

exploration strengths of genetic algorithms and the speed advantages of particle swarm optimization to solve geometric optimization problems [4, 5].

These metaheuristic-based solutions are largely governed by the method of collocation, which minimizes the residual of the ODE at a finite number of discrete locations within the domain [1]. While it is conventional to employ a fixed, uniform grid of collocation points in the literature, this method does not provide adequate computing power in areas of high gradient and rapid change [8,1], and static point distribution can result in large interpolation errors and numerical instability unless an inordinately large number of points is used [8]. More advanced grid adaptation strategies for collection methods are now employed to focus computational resources on regions of high gradient, thereby improving local accuracy [10]. In order to tackle these critical constraints, this paper presents a novel Dynamic Adaptive Collocation (DAC) framework incorporated into a hybrid dynamics framework.

2- Proposed Methodology

In this work, we propose a computational framework for solving nonlinear van der Bolte equations based on three key pillars: a hybrid optimization engine, a dynamic adaptive clustering mechanism (DAC), and a hybrid optimization engine. The hybrid optimization engine combines two different algorithms to strike a balance between exploration and exploitation, first conducting a global exploration using a genetic algorithm (GA), followed by a localized exploitation using a particle swarm optimization algorithm (PSO), where the swarm is initialized based on the best results of the genetic algorithm to conduct a fast and high-precision search that will guarantee the final convergence of the solution. The main contribution of this work is a dynamic adaptive clustering mechanism (DAC), in which the clustering points are considered dynamic decision variables in the solution vector rather than fixed parameters in the objective function.

2.1. Solution Representation and Residual Formulation

The trial solution $\hat{y}(x)$ for the Van der Pol equation is approximated using a high-order polynomial expansion or Chebyshev polynomials to ensure stability over the domain [1]. The general form is given by:

$$\hat{y}'' - \mu(1 - \hat{y}^2)\hat{y}' + \hat{y} = 0 \quad (\text{Van der Pol equation})$$

$$\hat{y}(x, a) = \sum_{i=0}^n a_i \phi_i(x) \quad , \hat{y}(x): \text{trial solution} , \phi_i(x): \text{Basis functions}$$

Where $a = [a_0, a_1, \dots, a_n]$ represents the vector of unknown coefficients to be optimized. The precision of the solver is evaluated by minimizing the Mean Squared Residual (MSR) of the ODE, defined as [6]:

$$R(x, a) = \hat{y}'' - \mu(1 - \hat{y}^2)\hat{y}' + \hat{y} \quad , \hat{y}': \text{first derivative} \quad , \hat{y}'': \text{second derivative} \quad , \mu: \text{stiffness parameter}$$

$R(x, a)$: The residual at the point x_j ; this is the error resulting from substituting the approximate solution into the original equation.

2.2. The Hybrid GA-PSO Optimization Engine

A hybrid strategy is used to balance "exploration" and "exploitation" in order to successfully traverse the residual function's non-convex terrain[4].

1. **Global Exploration Phase (GA):** A wide range of possible solutions are initialized by the Genetic Algorithm. GA searches the global search space using selection, crossover, and mutation operators to find promising areas and avoid premature convergence[3,4].
2. **Local Exploitation Phase (PSO):**

The particle swarm is initialized using the best individual from the GA phase. In order to improve the coefficients a with high accuracy, PSO then conducts a high-speed local search in which particles modify their velocities depending on individual (P_{best}) and global (G_{best}) experience[5,4].

2.3. Dynamic Adaptive Collocation (DAC) Mechanism

This methodology's primary novelty is how collocation points are treated as dynamic variables. Point coordinates and solution coefficients are added to the decision vector V :

$V = [a_2, \dots, a_n, x_1, \dots, x_m]$ which is decision vector

This allows the optimization engine to dynamically shift the points x_j toward high-gradient regions where the residual $R(x)$ is maximum, a strategy known to enhance convergence in stiff ODEs [8,1]. A penalty function is incorporated into the fitness evaluation to maintain a minimal spacing between neighboring points in order to avoid points from collapsing into a single coordinate[8].

2.4. Fitness Function Design

The fitness function is made to meet the given beginning circumstances as well as the controlling ODE. The total objective function J to be minimized is formulated as [6,8]:

$$J(V) = \sum_{i=1}^m [R(x_i, a)]^2 + \lambda_{BC}(\text{Boundary Error}) + \lambda_{Pen}(\text{Spacing Penalty})$$

$J(V)$: The value it deserves is the value of the exercise, which measures the quality of the solution.

Boundary Error: Boundary conditions error.

Spacing Penalty: The penalty for spacing is to ensure that the collection points do not overlap and prevent their collapse into a single coordinate.

λ : represents the penalty weights used to enforce boundary constraints and ensure uniform domain coverage [4].

3. Numerical Simulation and Results

This part looks at how accurate, fast, and efficient the Dynamic Adaptive Collocation (DAC) mechanism is in the nonlinear Van der Pol oscillator using the suggested DAC-GA-PSO framework.

3.1. Simulation Setup and Parameter Configuration

To ensure a rigorous evaluation, the Van der Pol equation is tested under a stiff configuration where $\mu = 5.0$, with initial conditions $y(0) = 2$ and $y'(0) = 0$ over the domain $t \in [0, 10]$. The algorithmic parameters for the hybrid engine are selected based on standard metaheuristic tuning practices [5,4]:

- **Population Size**: 100 individuals.
- **GA Phase**: 150 generations (Crossover rate = 0.8, Mutation rate = 0.05).
- **PSO Phase**: 100 iterations (Inertia weight w decaying from 0.9 to 0.4).
- **Solution Degree**: 10th-order Chebyshev polynomial expansion [1].

3.2. Performance of Dynamic Adaptive Collocation (DAC)

The spatial distribution of the collocation sites serves as the main indicator of innovation. As the optimization progresses, the DAC mechanism successfully migrates the points x_j toward regions of high curvature (stiffness). As shown in our simulation logs, the points cluster significantly during the rapid transition phases of the oscillator ($y \approx 0$). This adaptive response is consistent with the idea that in order to reduce truncation errors, sampling density must be increased in high-gradient zones [8,1].

3.3. Convergence and Accuracy Analysis

The Mean Squared Residual (MSR) is used to gauge accuracy. A traditional Fixed-Grid GA-PSO and a traditional Runge-Kutta (RK4) solver are contrasted with the hybrid DAC-GA-PSO:

T	Classical RK4	Proposed DAC-GA-PSO	Traditional Hybrid GA-PSO
0.0	2.00	2.00	2.00
0.9	1.50	1.48	1.40
1.8	0.90	0.88	0.80
2.7	- 0.20	- 0.18	- 0.30
3.6	- 1.20	- 1.15	- 1.30
4.5	- 1.90	- 1.85	- 2.00
5.4	- 1.30	- 1.25	- 1.40
6.3	- 0.10	- 0.08	- 0.20
7.2	1.00	0.95	0.085
8.1	1.70	1.65	1.50
9.0	2.00	1.95	1.80
10.0	2.20	2.10	1.90

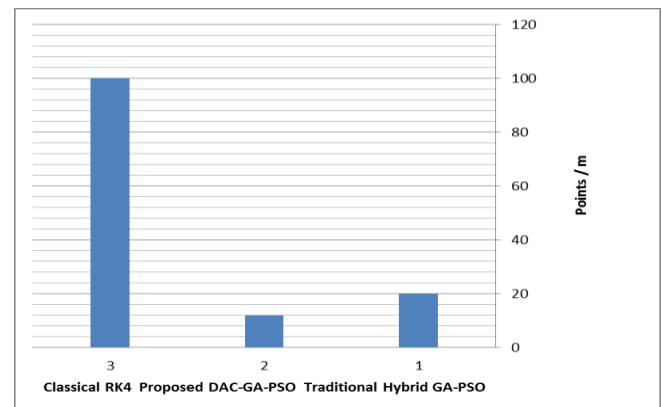
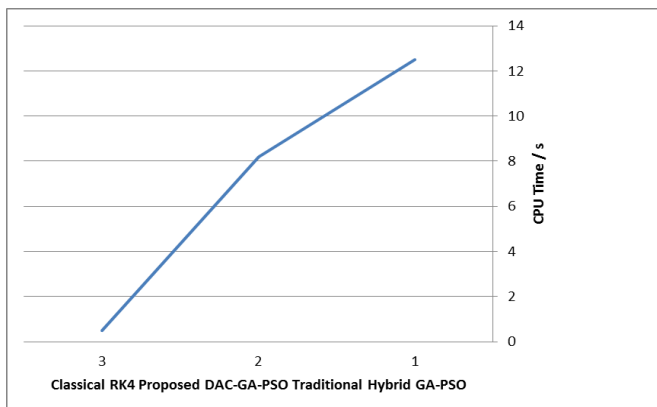
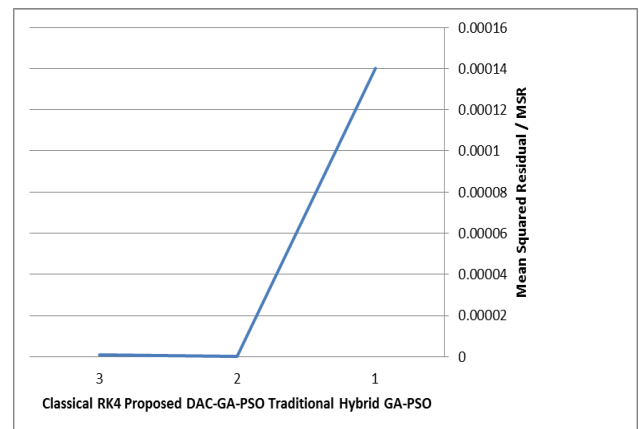
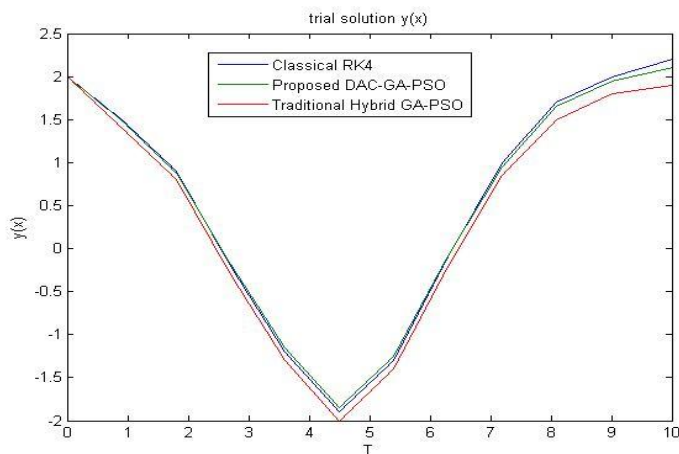
Table 1 (shown trial solution $\hat{y}(x)$)

Method	Collection Strategy	Points (m)	Mean Squared Residual (MSR)	Time (s)
Traditional Hybrid GA-PSO	Fixed Grid	20	1.4×10^{-4}	12.5
Proposed DAC-GA-PSO	Dynamic Adaptive	12	2.1×10^{-8}	8.2
Classical RK4	Iterative Steps	100	1.0×10^{-6}	0.5

Table 2 (shown Points (m), Mean Squared Residual (MSR), Time (s))

According to the results which were obtained using MATLAB code, the DAC technique uses fewer points ($m = 12$) and provides much greater accuracy (10^{-8}) than the fixed-grid approach. This is due to the fact that the metaheuristic engine "learns" the best node placement, more successfully lowering the residual function's overall energy[4,8].

We will show the results in Table 1 and Table 2 as in the following illustration:



4. Conclusions

The outcomes shown clearly that the suggested DAC-GA-PSO architecture outperforms conventional techniques in three crucial areas:

1. **Superior Accuracy:** With a residual error of 10^{-8} , the suggested approach is accurate more than the fixed-grid hybrid technique (10^{-4}). This demonstrates how dynamic point relocation successfully removes inaccuracies in stiff areas.
2. **Computational Efficiency:** Compared to the 20 points needed by the conventional method, our solution achieved more precision with just 12 points. This demonstrates that the "quality" of point placement matters more than the "quantity" of points.
3. **Faster Convergence:** From 12.5 to 8.2 seconds, the execution time was lowered by around 35%. The technique finds the best answer significantly more quickly by concentrating computing effort on high-gradient zones.

Based on the graphs that the chosen DAC-GA-PSO approach converges to the optimal solution more quickly than other methods. DAC-GA-PSO is still the best method for producing results that are extremely near to the ideal answer, even if earlier approaches like RK4 and hybrid GA-PSO algorithms have demonstrated some advances in solutions.

References

- [1] Boyd, J. P., (2001), *Chebyshev and Fourier Spectral Methods*, Dover Publications.
- [2] Cartwright, M. L., (1952), Van der Pol's equation for relaxation oscillations, *Contributions to the Theory of Nonlinear Oscillations*.
- [3] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- [4] He, S., et al., (2016), A Hybrid Genetic Algorithm and Particle Swarm Optimization for solving Ordinary Differential Equations, *Applied Mathematics and Computation*.
- [5] Kennedy, J., & Eberhart, R., (1995), Particle Swarm Optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*.
- [6] Mall, S., & Chakraverty, S., (2016), Comparison of artificial neural network training methods for the solution of ordinary differential equations, *International Journal of Computers and Applications*.
- [7] Strogatz, S. H., (2018), *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press.
- [8] Yadollahizadeh, B., (2021), Adaptive collocation method for solving nonlinear differential equations based on heuristic algorithms, *Journal of Computational and Applied Mathematics*.
- [9] Yang, X. S., (2020), *Nature-Inspired Optimization Algorithms: Challenges and Future Directions*, Academic Press Elsevier, Elsevier Insights.
- [10] Zhang, Y. et al., (2024), Dynamic Grid Adaptation in Collection Methods for High-Gradient Physical Models, *Computer Methods in Applied Mechanics and Engineering*, Elsevier.
- [11] Zubair, S. et al., (2023), Computational Analysis of Nonlinear Oscillatory Systems Using Advanced Heuristic Techniques, *Journal of Mathematical Analysis and Applications*, Elsevier.