
Hybridization of the Rayleigh-Ritz method with the particle swarm algorithm for solving ordinary differential equations

Fatima.A.Y.Majdal^{1*}, Azzam S. Y. Aladool^{2*}

^{1,2*}Department of Mathematics, College of Education for Pure Sciences, University of Mosul, Mosul, Iraq

E -mails :^{1*} fatima.23esp122@student.uomosul.edu.iq

E -mails :^{2*} azzam.aladool@uomosul.edu.iq

Abstract

Ordinary differential equations subject to boundary conditions (BVPs) play a crucial role in many scientific disciplines. Boundary value problems (BVPs) can be solved analytically, however, there are different types of boundary value problems are difficult to solve. Therefore, the numerical and approximation approaches for solving boundary value problems are used. In this paper, an algorithm is proposed for hybridizing the Rayleigh-Ritz method with the particle swarm algorithm to find approximate solutions of boundary value problems. To achieve this, it is vital to minimize the fitness function value. A discrete least squares weighted function is used to calculate the fitness function. Linear and non-linear boundary value problems are solved using the proposed algorithm. A comparison was made between the approximate and exact solutions. This is carried out by presented variety of linear and nonlinear examples. Moreover, the convergence of the proposed algorithm presented. The results of the approximate solution are promising in terms of convergence and accuracy.

Keywords: ordinary differential equations, particle swarm optimization, Rayleigh-Ritz method

1. Introduction.

Ordinary differential equations (ODEs) arise in a wide variety of mathematical fields, as well as in engineering and many other application areas such as Biological applications, statistical physics and molecular dynamics [1].

Although most of differential equations types can be solved analytically, approximation and numerical techniques have gained interest as methods for solving differential equations, such as Runge-Kutta methods, Adams-Moulton method, Euler's method, Adams Bashforth method, Heun's method [2], Shooting technique[3], and Rayleigh-Ritz method(RRM) [4].

Alternatively, the use of intelligence algorithms has recently applied to solve mathematical problems such as Neural Networks (NN), particle swarm optimization (PSO)[5], Genetic Algorithm(GA)[6]. These intelligent

algorithms have been used successfully to solve mathematical problems. This is carried out by hybridizing the particle swarm optimization with Padé Approximant for solving ordinary differential equations[7]. Particle swarm optimization is approximation solutions of nonlinear differential equations with Fourier series[8]. solving the Fredholm integral equation [9], Genetic algorithm was used to solve ordinary differential equations[10].

In general, using intelligent algorithm in association with approximation methods led to more accurate solutions and less computational efforts.

The (RRM) is one of the approximation methods to solve boundary value problems. The Rayleigh-Ritz method was also used to solve the second kind linear Fredholm integral equations[11]. It has also been primarily applied to solve linear differential equations[12,13]. However, there are lacks of attempts to use the Rayleigh-Ritz method for solving nonlinear differential equations. In a recent study, it was found that the applicability of the method to nonlinear problems, nevertheless, the obtained results inconsistent with the exact solution [14] [15].

The concept of the (RRM) is to minimize the functional and then take its derivative with respect to unknown coefficients to create system of algebraic linear equations then determining the unknown coefficients of the approximate solution [16]. In this paper, a hybrid algorithm is proposed based on the particle swarm optimization algorithm and the Rayleigh-Ritz method to approximate solution of linear and nonlinear (BVPs). The proposed algorithm determines the values of the unknown variables resulting from the (RRM). Consequently, this algorithm shortens many algebraic operations and saves time, and computational efforts. In addition, produce an accurate result for solving nonlinear differential equations.

This introduction is followed by preliminary. The methodology of solving linear and nonlinear boundary value problems is illustrated. Then, several examples are shown. Finally, the results are discussed.

2. PRELIMINARY

In this section, an explanation of the (RRM) is presented. There is also a discussion of the fundamentals of particle swarm optimization algorithms (PSO). This is followed by a description of the PSO algorithm.

2.1. Rayleigh-Ritz method (RRM)

The functional form of a given differential equation is essential for the (RRM). Functional is an application of a function (or set of functions) to a value. The concept of the functional is derived from the study of variational principles which is commonly applied to physical problems as well as other

problems. Mathematically, a variational principle is the determination of the extreme value of a typical function's integral,

For more clarification, the Rayleigh-Ritz method can be explained by the following:

second-order (BVP) is considered:

$$y'' + p(x)y + q(x) = 0, \quad y(a) = 0, y(b) = 0 \quad (1)$$

which can be expressed as follows :

$$I(y(x)) = \int_a^b f(x, y, y') dx \quad (2)$$

Where $y(x)$ satisfies the boundary conditions $y(a) = y(b) = 0$, the necessary condition for $I(y)$ to have a maximum is that $f(x, y, y')$ must satisfy the Euler- Lagrange differential equation.

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left(\frac{\partial f}{\partial y'} \right) = 0 \quad (3)$$

To find the functional associated with any equation follow these steps:[17]

- Multiply the equation by $v(x)$ from the left and integrate according to the given x values (on the problem domain).
- Use integration by parts for the term containing the second-order derivative.
- The boundary integrals should be expressed on the boundary conditions.
- Convert the equation to a homogeneous one and divide the nonlinear terms within the integral by the exponent of each one of them.

So that the functional of Eq.(1) is written as follows:

$$I(V(x)) = \int_a^b \left[\left(\frac{dv}{dx} \right)^2 - pv^2 - 2qv \right] dx = 0 \quad (4)$$

Let $\varphi_i(x)$ for $1 \leq i \leq n$ be n independent weight functions satisfying the corresponding boundary conditions of the given differential equation and let consider:

$$\varphi_i(x) = (x - a)^i * (b - x) \quad (5)$$

Define

$$V(x) = \sum_{i=1}^n \alpha_i \varphi_i(x) \quad (6)$$

Substituting for $v(x)$ in Eq.(4) we get:

$$I(V(x)) = \int_a^b \left[\left(\frac{d}{dx} \sum_{i=1}^n \alpha_i \varphi_i(x) \right)^2 - p \left(\sum_{i=1}^n \alpha_i \varphi_i(x) \right)^2 - 2q \sum_{i=1}^n \alpha_i \varphi_i(x) \right] dx \quad (7)$$

After completing the integration process, the result of the integration is derived with respect to unknown coefficients and equated to zero

$$\frac{\partial I(V(x))}{\partial \alpha_1} = \frac{\partial I(V(x))}{\partial \alpha_2} = \dots = \frac{\partial I(V(x))}{\partial \alpha_n} = 0 \quad (8)$$

Where a set of algebraic equations is obtained Which are then solved algebraically to obtain the values of the unknown coefficients α_i .

Where $v(x)$ is the approximate solution after substituting the values of the coefficients α_i [18].

2.2. Particle swarm optimization

Kennedy and Eberhart introduced in 1995 a new algorithm inspired by the collective behavior of animals such as birds, fish, and ants, called the particle swarm algorithm (PSO)[5]. Each member of the algorithm is called a particle, and each particle has a (position) and (velocity) that can be continuously updated based on time. The position and velocity of each particle can be represented by a mathematical model[19][20]:

$$\delta_i(t+1) = \omega \times \delta_i(t) + c_1 \times rand_1 \times (\sigma_i(t) - \gamma_i(t)) + c_2 \times rand_2 \times (\lambda(t) - \gamma_i(t)) \quad (9)$$

$$\gamma(t+1) = \gamma_i(t) + \delta_i(t+1) \quad (10)$$

where δ_i represents the velocity of the i^{th} particle, and γ_i represents its position, ω is a damping factor, c_1 and c_2 are two acceleration coefficients it is important to note that They are positive constants, $rand_1$, $rand_2$ are two random factors in the interval [0,1], number of population in swarm ($nPop$), $\sigma_i(t)$ is the best position for i^{th} particle at time t and $\lambda(t)$ is the global best position among all particles, $t = 1, 2, 3, \dots, Maxit$, represents the number of iterations, and ($Maxit$) is the max iteration, In order to decrease the

probability that a particle leaves search space, the velocity vectors are clamped within a range of $[varmax, varmin]$.

A particle swarm optimization (PSO) algorithm can search in N dimensions depending on the number of variables $nVar$, the values of c_1 and c_2 and ω are determined as follows:[7]

$$\chi = \left\{ \begin{array}{ll} \frac{2k}{|2 - \vartheta - \sqrt{\vartheta^2 - 4\vartheta}|} & \text{if } \vartheta > 4 \\ k & \text{if } \vartheta \leq 4 \end{array} \right\} \quad (11)$$

$$\vartheta = \vartheta_1 + \vartheta_2 . c_1 = \vartheta_1 \times \chi . c_2 = \vartheta_2 \times \chi . \omega = \chi \quad (12)$$

$k \in [0,1]$, and the values ϑ_1, ϑ_2 are chosen randomly to achieve $\vartheta > 4$. Using the following parameter ε , the non-converging behavior of PSO can be avoided by continuously damping the velocity in each iteration.

$$(t + 1) = \varepsilon \times \omega(t) \quad (13)$$

We can obtain the velocity space by following these steps:

$$\begin{aligned} \text{Max Velocity} &= \alpha \times (\text{VarMax} - \text{VarMin}), \\ \text{Min Velocity} &= -\text{Max Velocity} \end{aligned} \quad (14)$$

and α is a positive parameter.

2.3. The PSO Algorithm

PSO consists the following steps:[21][7]

- i) Parameters and inputs are initialized, $\vartheta_1, \vartheta_2, k, \varepsilon, \alpha, VarMax, VarMin, nVar, Maxit$ and $(nPop)$
- ii) Let the initial velocity be equal to zero.
- iii) Initializing an arbitrary particles' position in the search space.
- iv) The fitness function of the swarm particles is evaluated
- v) Find the global optimal position (gbest) of the particle swarm, and the particles' optimal position (pbest $_i$), where $i = 1, 2, \dots, N$.
- vi) The position and velocity of particles are updated according to Eq. (6) and Eq. (7).
- vii) Go to Step (iv) if the number of iterations is less than $Maxit$.

3. METHODOLOGY

An approach for finding the approximate solution ($V_{appr}(x)$) of linear and nonlinear (ODEs) with boundary conditions is presented. In this work, the Rayleigh-Ritz method is hybridized, by first finding the functional of the

given (BVP). In addition, the fitness function and the weighted discrete least squares function are introduced. Finally, an algorithm for finding the $V_{appr}(x)$ of the given (BVP) is presented.

3.1 The proposed method

The Rayleigh-Ritz method is hybrid with particle swarm optimization to solve (ODEs) with boundary conditions [18].

Suppose that the approximate solution to the ODEs can be written as follows:

$$y(x) \cong V_{appr}(x) = \sum_{i=1}^n \alpha_i \varphi_i(x) \quad (15)$$

Where $x \in I = [x_0, x_n]$, and α_i are real coefficients, $\varphi_i(x)$ be n independent weight functions satisfying the corresponding boundary conditions, $V_{appr}(x), y(x)$ are the approximate solution and the exact solution respectively.

As the central idea of Rayleigh – Ritz is to determine coefficients α_i , the integral is found and then the equation obtained from the integral is derived with respect to α_i and set equal to zero where $i = nVar$

$$E_r(x) = \left| \frac{\partial (I(V(X)))}{\partial \alpha_i} \right| \quad (16)$$

An optimal solution to the ODEs can be obtained when $E_r(x)$ approach to zero. It is necessary to use a quantitative criterion that determines the accuracy of the approximate solution to reduce $E_r(x)$. This can be minimized using a discrete least squares weighted function[7].

3.2. The Discrete Least Squares Weighted Function (DLSWF)

The discrete least squares weighted function is computed as follows:[7][9]

Taking the interval I and dividing it into N points $\{x_0 = a, x_1, x_2, x_3, \dots, x_n = b\}$, where $x_k = x_0 + hk, \forall k = 0, 1, 2, 3, \dots, n$ and $h > 0$, and

$$DLSWF = \sqrt{\frac{\sum_{k=1}^N (Er(x_k))^2}{N}} \quad (17)$$

3.3. RRM -PSO Algorithm

Here is an outline of the proposed algorithm for solving ODEs:

Step (1): Create an array of coefficients functional $nVar$ as

follows: $[a_1, a_2, \dots, a_n]$

Step (2): Convert the ODE into the implicit form:

$$E_r(x) = \left| \frac{\partial (I(V(X)))}{\partial \alpha_i} \right|$$

Step (3): The fitness function is determined as follows:

$$FITFUN = DLSWF = \sqrt{\frac{\sum_{k=1}^N (Er(x_k))^2}{N}} \quad (18)$$

Step (4): To determine $V(x)$ coefficients, PSO parameters are initialized.

Step (5): A PSO algorithm applied to minimize fitness function

Step (6): Do step (5) until $FITFUN < TOL$ or the maximum number of iterations is reached.

By calculating the Mean Absolute Error (MAE) using the approximate solution $V_{app}(x)$ and exact solution y , the algorithm is evaluated:

$$MAE = \frac{\sum_{k=1}^N |y(x_k) - V_{app}(x_k)|}{N} \quad (19)$$

4. NUMERICAL RESULTS:

RRM -PSO Algorithm is utilized in this section to approximate the solution of linear and nonlinear (BVPs). Additionally, the algorithm's convergence is shown

4.1. Numerical Examples

In this paper, linear and non- linear ODEs are included in **TABLE1:** [18,22,23]

TABLE 1: Shows different examples of ODEs, with their exact solutions.

Examples	ODEs	boundary condition	Exact Solutions
LODE1	$y'' = 2(x - 1)$	$y(1) = 0, y(2) = 0$	$y(x) = \frac{(x-1)^3}{3} - \frac{x-1}{3}$
LODE2	$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x)$	$y(0) = 0, y(1) = 0$	$y(x) = \sin(\pi x)$
LODE3	$y'' + \frac{\pi^2}{4} y = \frac{\pi^2}{16} \cos\left(\frac{\pi}{4} x\right)$	$y(0) = 0, y(1) = 0$	$y(x) = -\frac{1}{3} \cos\left(\frac{\pi}{4} x\right) - \frac{\sqrt{2}}{6} \sin\left(\frac{\pi}{2} x\right) + \frac{1}{3} \cos\left(\frac{\pi}{4} x\right)$
NLODE4	$y(x) + y^2 + y = -2 \cos(x) + \sin(x) * (1 - x)^2$	$y(0) = 0, y(1) = 0$	$y(x) = -\frac{x^2}{2} + \frac{1}{2} x$

$$\text{NLODE5} \quad y'' + y^2 = x^2(1-x)^2 - 2 \quad y(0) = 0, y(1) = 0$$

$$y(x) = x(1-x)$$

$$\text{NLODE6} \quad y'' = -(y)^2 + \sin^2(\pi x) - \pi^2 \sin(\pi x) \quad y(0) = 0, y(1) = 0$$

$$y(x) = \sin(\pi x)$$

TABLE 2 shows examples of parameter values. After running the algorithm ten times, its reliability is verified. To implement the algorithm, Matlab R2020a software is used, There is a type of computer Lenovo laptop, Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 2.49 GHz , RAM 4.00 GB , Windows 10 Pro and system type 64-bit.

TABLE2: The following parameters are used in all examples:

Parameter	$\vartheta 1$	$\vartheta 2$	k	ε	α	h	TOL	$Maxit$	$npop$
Value	2.05	2.05	1	0.8	0.4	0.01	1e-10	200	50

As a result of the PSO algorithm, the coefficients for approximate solutions of ODE for all examples are listed in TABLE 3. For example, one of the approximate solutions of can be found for LODE1 ($V_{appr}(x)$) is shown, where $nVar=5$, and $x \in [0,1]$ as follows:

$$y(x) \cong V_{appr}(x)$$

$$= -0.348437(x-1) * (2-x) \quad (20)$$

$$- 0.260558(x-1)^2 * (2-x) \quad)$$

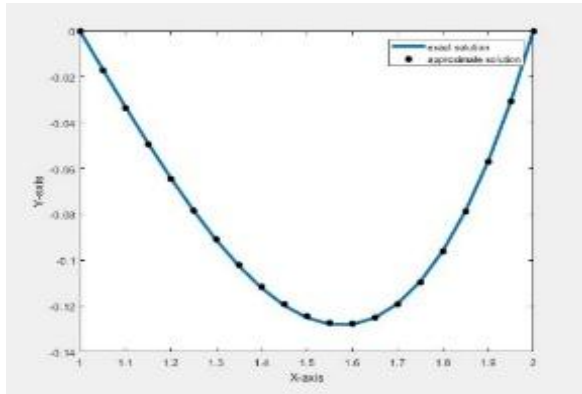
$$- 0.024038(x-1)^3 * (2-x)$$

$$- 0.184347(x-1)^4 * (2-x) + 0.1595837(x-1)^5 * (2-x)$$

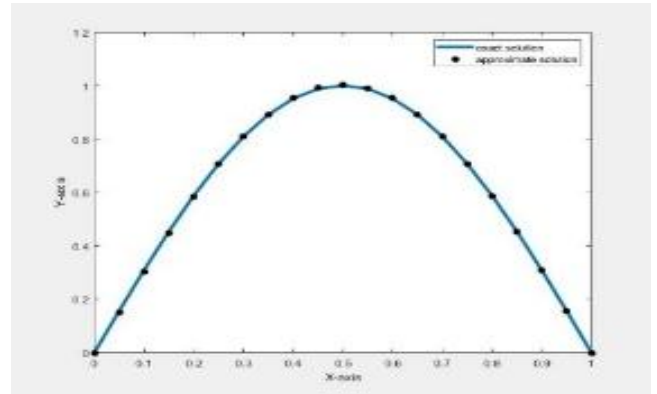
TABLE 3: Displays variables based on their values using by FIE - PSO algorithm.

The number of variables, $nVar=5$, $[Var \max, Var \min]=[6,-6]$.

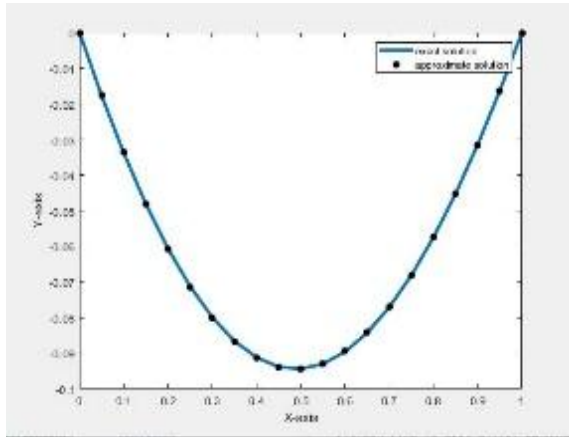
<i>coeff.</i>	Examples					
	LODE1	LODE2	LODE3	NLODE4	NLODE5	NLODE6
a_1	-0.348437	3.003620	-0.418504	1	1	3.147213
a_2	-0.260558	3.9999999	0.154931	-0.03802	-0.156086	2.663599
a_3	-0.024038	-3.695963	-0.246357	0.064266	0.394280	-0.48843
a_4	-0.184347	-0.919661	0.385042	-0.41505	0.044966	-3.19456
a_5	0.1595837	0.746937	-0.239774	0.237617	-0.343871	0.891574



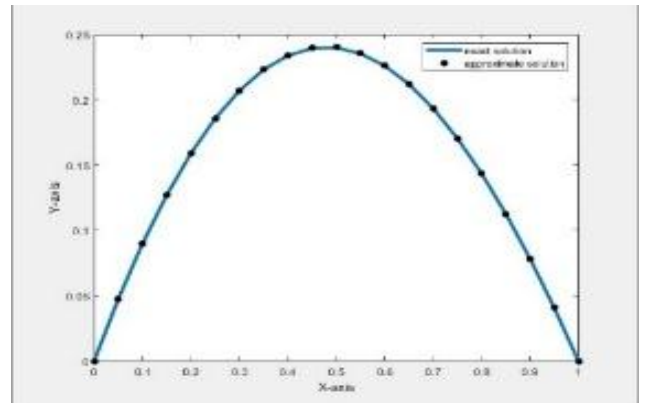
LODE1



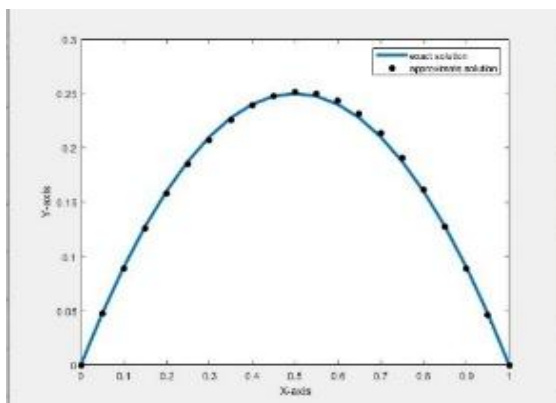
LODE2



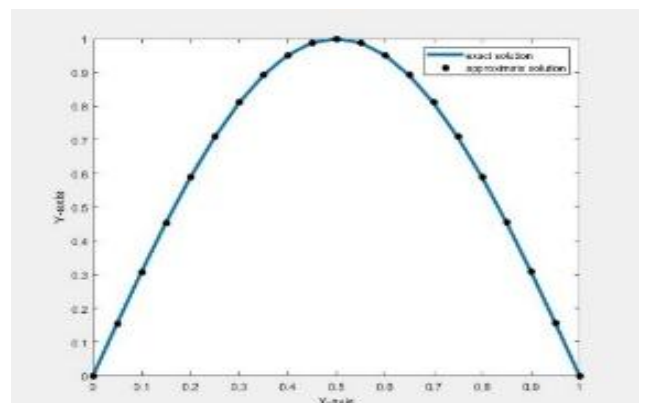
NLODE3



NLODE4



NLODE5



NLODE6

Figure 1: Exact and approximate solutions are compared

The exact solutions of several examples are shown in Figure 1. Within the defined domain, the approximate solutions in Figure 1 were consistent with the exact solution. Hence, the RRM-PSO algorithm can solve linear and nonlinear ODEs.

Though the algorithm may reach the exact solution, the approximation often leaves an error rate. The acceptable error is determined by the value of TOL. The Mean Absolute Error (*MAE*) is obtained to give the error between the exact solution and the approximate solution shown in TABLE 4.

TABLE 4: Indicates the Mean Absolute Error between the approximate and exact solutions

Examples	<i>MAE</i>
ODE1	3.2627167897355e-04
ODE2	0.002001728345009
ODE3	0.001565151398762
ODE4	1.506774892229766e-04
ODE5	0.001808212701365
ODE6	0.001912864083887

As a result of this method, convergence occurs rapidly. FIGURE 2 shows the convergence of the RRM-PSO algorithm over 200 iterations for all examples, acceptable solutions are obtained with less than 60 iterations and are stable to 200 iterations (see FIGURE 2). Display the convergence of the algorithm.

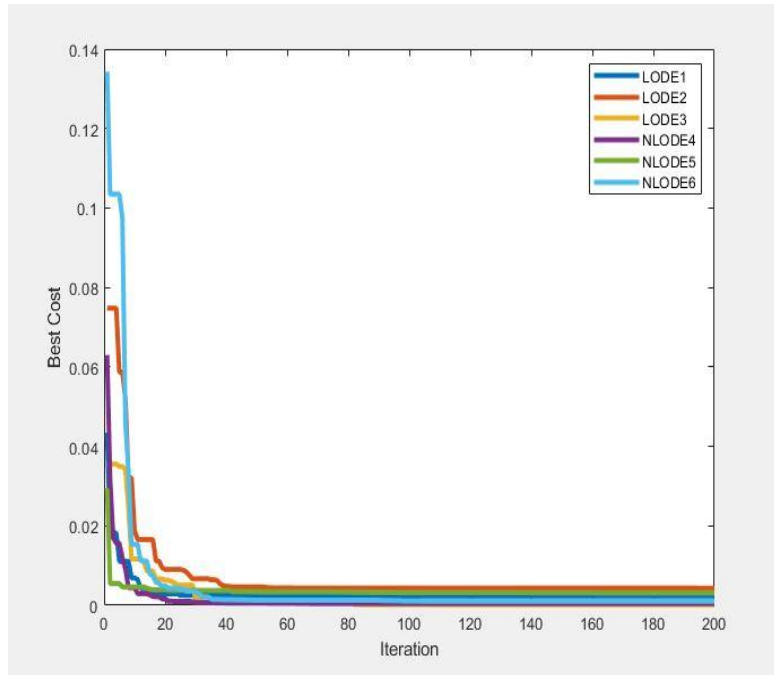


Figure 2: Display the convergence of the algorithm

5. Conclusion

The particle swarm algorithm is practical for hybridization with Rayleigh-Ritz method to solve boundary value problems (BVPs). The functional was formulated and then the integral result was derived with respect to α_i . The proposed RRM was used to find the values of α_i which in turn represent an approximate solution of linear and nonlinear BVPs. Based on the results, the algorithm successfully solves both linear and nonlinear BVPs. The advantages of using this approach are the elimination of algebraic operations, convergence, and accuracy.

6. Acknowledgements

Gratitude is expressed by the authors to both the technical and academic staff of the University of Mosul as the College of Education for pure Science. for their support to facilitate this work.

7. References

- [1] A. Grigorian, "Ordinary Differential Equation," no. July, 2009.
- [2] K. E. Atkinson, W. Han, and D. Stewart, *Numerical Solution of Ordinary Differential Equations*. Wiley, 2009. doi: 10.1002/9781118164495.
- [3] A. W. Manyonge, R. Opiyo, D. Kweyu, and J. S. Maremwa, "Numerical solution of non-linear boundary value problems of ordinary differential equations using the shooting technique," *J. Innov. Technol. Educ.*, vol. 4, no. 1, pp. 29–36, 2017, doi: 10.12988/jite.2017.61250.
- [4] T. Błaszczuk, "Application of the Rayleigh-Ritz method for solving fractional oscillator equation," *Pr. Nauk. Inst. Mat. i Inform. Politech. Częstochowskiej*, vol. 8, no. 2, pp. 29–36, 2009.
- [5] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Comput.*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.
- [6] F. J. da Cunha Pires Soeiro, L. B. Neto, and A. J. da Silva Neto, "Genetic algorithms," *Comput. Intell. Appl. to Inverse Probl. Radiat. Transf.*, pp. 43–50, 2023, doi: 10.1007/978-3-031-43544-7_6.
- [7] A. T. Abed and A. S. Y. Aladool, "Applying particle swarm optimization based on Padé approximant to solve ordinary differential equation," *Numer. Algebr. Control Optim.*, vol. 12, no. 2, p. 321, 2022, doi: 10.3934/naco.2021008.
- [8] M. Babaei, "A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization," *Appl. Soft Comput.*, vol. 13, no. 7, pp. 3354–3365, Jul. 2013, doi: 10.1016/j.asoc.2013.02.005.

- [9] I. sheekhoo and A. Aladool, "Solving Fredholm integral equation based on Padé approximants using Particle swarm optimization," *J. Educ. Sci.*, vol. 32, no. 1, pp. 81–90, Mar. 2023, doi: 10.33899/edusj.2023.137531.1309.
- [10] D. A. Diver, "Applications of genetic algorithms to the solution of ordinary differential equations," *J. Phys. A. Math. Gen.*, vol. 26, no. 14, pp. 3503–3513, Jul. 1993, doi: 10.1088/0305-4470/26/14/017.
- [11] Ruşen Kaya & Hasan Taşeli, "A Rayleigh–Ritz Method for Numerical Solutions of Linear Fredholm Integral Equations of the Second Kind," 2022. [Online]. Available: <https://link.springer.com/article/10.1007/s10910-022-01344-9>
- [12] A. K. Jimoh, "On Rayleigh-Ritz and Collocation Methods for Solving Second Order Boundary Value Problems of Ordinary Differential Equations," *Earthline J. Math. Sci.*, vol. 14, no. 2, pp. 205–220, Dec. 2023, doi: 10.34198/ejms.14224.205220.
- [13] S. Bao, J. Cao, and S. Wang, "Vibration analysis of nanorods by the Rayleigh-Ritz method and truncated Fourier series," *Results Phys.*, vol. 12, no. October 2018, pp. 327–334, Mar. 2019, doi: 10.1016/j.rinp.2018.11.085.
- [14] J. Wang and R. Wu, "The Extended Galerkin Method for Approximate Solutions of Nonlinear Vibration Equations," *Appl. Sci.*, vol. 12, no. 6, p. 2979, Mar. 2022, doi: 10.3390/app12062979.
- [15] H. Jing, X. Gong, J. Wang, R. Wu, and B. Huang, "An Analysis of Nonlinear Beam Vibrations with the Extended Rayleigh-Ritz Method," *J. Appl. Comput. Mech.*, vol. 8, no. 4, pp. 1299–1306, 2022, doi: 10.22055/JACM.2022.39580.3434.
- [16] S. S. SASTRY, *Introductory Methods of Numerical Analysis*, vol. 4, no. 1. 2012, pp. 1–23.
- [17] A. A. Salih, "Finite Element Method," pp. 188–195, 2012, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Finite_element_method&oldid=891797554
- [18] J. C. P. MILLER, "Numerical Analysis," *Nature*, vol. 177, no. 4517, pp. 953–954, May 1956, doi: 10.1038/177953b0.
- [19] A. Sahu, S. K. Panigrahi, and S. Pattnaik, "Fast Convergence Particle Swarm Optimization for Functions Optimization," *Procedia Technol.*, vol. 4, pp. 319–324, 2012, doi: 10.1016/j.protcy.2012.05.048.
- [20] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering," *J. Zhejiang Univ. A*, vol. 10, no. 4, pp. 512–519, Apr. 2009, doi:



وقائع المؤتمر العلمي لكلية التربية الأساسية في مجال العلوم المصرفية

وتحت شعار

(العلوم المصرفية والتطبيقية بوابة لخدمة المجتمع)

يومي الاربعاء و الخميس 28-29/5/2025

10.1631/jzus.A0820196.

[21] D. Yong, W. Chuansheng, and G. Haimin, "Particle Swarm Optimization Algorithm with Adaptive Chaos Perturbation," *Cybern. Inf. Technol.*, vol. 15, no. 6, pp. 70–80, Dec. 2015, doi: 10.1515/cait-2015-0068.

[22] P. S. Phang, Z. Abdul Majid, F. Ismail, K. I. Othman, and M. Suleiman, "New Algorithm of Two-Point Block Method for Solving Boundary Value Problem with Dirichlet and Neumann Boundary Conditions," *Math. Probl. Eng.*, vol. 2013, no. 3, pp. 1–10, 2013, doi: 10.1155/2013/917589.

[23] J. D. F. Richard L. Burden, "Numerical Analysis, Seventh Edition," 2000.