Journal of Education and Science (ISSN 1812-125X)

Using Invasive Weed Optimization Algorithm with a Padé Approximation to Solve a System of Volterra Integral Equations

M. J. H. Almawlaa [(1)], A. S. Y. Aladool [(2)]

[(1,2)] Department of Mathematics, College of Education for Pure Sciences, University of Mosul, Mosul, Iraq

Article information

Article history:
Received: June 04, 2024
Revised: August 09, 2024
Accepted: August 11, 2024
Available online: December 01, 2024

Keywords:
Padé Approximation.
System of Volterra Integral Equations.
Invasive Weed Optimization Algorithm.

Correspondence:
Mohammed J.H. Almawlaa
mohammed.22esp34@student.uomosul.edu.iq

Abstract

Engineering and physics applications are described mathematically as a system of Volterra integral equations (SVIEs). Although many numerical techniques have been considered to solve SVIEs, developing more stable, and efficient algorithms is still challenging. In this work, an algorithm for solving linear and nonlinear systems of Volterra integral equations of the second kind is proposed. The Invasive Weed Optimization (IWO) algorithm is combined with Padé approximant expansion. Since the solution is represented as functions of different forms, Padé approximation which is fractional expansion is used to obtain results of high accuracy. SVIEs are transformed into an unconstrained optimization problem. Thereafter, the discrete least squares weighted function was estimated to minimize the value of the fitness function. This algorithm is applied to solve a variety of linear and nonlinear examples and compare their solutions to the exact solutions. The performance of the algorithm provides accurate results in terms of convergence and stability.

DOI: 10.33899/edusj.2024.150513.1465,

1. Introduction:

Integral Equations is one of the interesting topics in Mathematics, which is one of the most important mathematical tools in both pure and applied mathematics courses. The integral equation comes naturally in solving the initial and boundary value problems that can be protracted into finding some phenomenon of physical significance, [1]. These integral equations are common in history-dependent cases, including chemical reactions, damped vibrations, fluid motion, population systems, seismology, viscoelasticity, heat transmission, and diffusion, [2]. The Volterra System of Integral Equations (SVIEs) is a set of equations in which the required function appears within the integral. Many applications of these equations lead researchers to encounter different ways to solve them, [3].

Various analytical techniques have been designed to find a solution to a system of Volterra integral equations, including the Adomian decomposition method, [4], the direct calculation method[5], and the Laplace transform method, [6]. Moreover, numerical methods can be used to solve a "system of Volterra integral equations" when it is difficult to obtain their analytical solution. Some of these numerical methods are the rationalized Haar functions, [7], Implicit Trapezoidal, [8], Monte-Carlo Method, [9], and Spline Functions, [10].

Researchers then used optimization algorithms as a new method to find approximate solutions to this type of equation. The most vital algorithms are those based on artificial intelligence, [11]. Artificial intelligence algorithms are algorithms inspired by nature and the behavior of living organisms. For example, particle swarm optimizations (PSO), [12], Genetic Algorithms (GA), [13], and Invasive Weed Optimization (IWO), [14]. One of the researchers, Daniel, combined the genetic algorithm with Taylor expansion to solve differential equations, [15]. The particle swarm optimizations algorithm (PSO) has been used with a Padé approximation to solve an ordinary differential equation, [16], The use of the genetic algorithm with

Padé approximation and the least squares weight function in solving a system of a differential equation, [17] and Solving Fredholm Integral Equation Based On Padé Approximants Using Particle Swarm Optimization,[18].

Using the Padé approximation with optimization algorithms to solve differential equations leads to more accurate results than Fourier and Taylor expansions. This is because the Padé approximation is a fractional expansion that provides highly accurate results using a small number of variables. In addition, the search field can be defined in the interval [-1,1] in the Padé expansion,[16]. The IWO algorithm is a population-based algorithm that simulates the process of weed colonization and competition to find the best solution to a given problem, [19]. An Invasive Weed Optimization Algorithm with Padé approximation is proposed to solve the System of Volterra Integral Equations (SVIEs). In this algorithm, The Discrete Least Squares Weighted Function(DLSWF) is used to reduce the fitness function value in each iteration, and so the process is repeated until the iterations end or the algorithm stops according to the stopping criterion. In the end, The exact solution is compared to the approximate solution to determine the accuracy of the results. Additionally, the errors that arise from the approximate solution are shown, and its stability.

This study consists of five sections. The first section is the introduction, where an introduction to integral equations is presented, with some analytical and numerical methods for solving them. The second section presents some basic concepts of the research. The third section explains the methodology used to solve the system of Volterra integral equations. The fourth section includes the numerical results obtained from applying the proposed algorithm to a variety of linear and nonlinear examples. The fifth section includes the most important conclusions reached in this study.

1. Preliminary

This section presented the second-order Systems Volterra integral equations. In addition, the foundations of the Invasive Weed Optimization Algorithm are explained.

1.1. System Volterra Integral Equations of the Second Kind (SVIEs)

The system of Volterra integral equations of the second kind can be defined as follows:

$$\eta_1(ţ) = ƒ_1(ţ) + \int_0^ţ (ķ_1(ţ,ş)\eta_1(ş) + ķ`_1(ţ,ş)\eta_2(ş) + \cdots)dş$$

$$\eta_2(ţ) = ƒ_2(ţ) + \int_0^ţ (ķ_2(ţ,ş)\eta_1(ş) + ķ`_2(ţ,ş)\eta_2(ş) + \cdots)dş$$

$$\vdots$$

$$\eta_i(ţ) = ƒ_i(ţ) + \int_0^ţ (ķ_i(ţ,ş)\eta_j(ş) + ķ`_i(ţ,ş)\eta_{j+1}(ş) + \cdots)dş \quad ,i,j = 1,\dots,n \qquad (1)$$

The unknown functions, $\eta_1(ţ), \eta_2(ţ), \dots, \eta_i(ţ)$ will be calculated inside and outside the integral sign. A kernels $ķ_i(ţ,š)$ and $ķ`_i(ţ,š)$, and the function $ƒ_i(ţ)$ are real-valued functions, [20].

2.2. Invasive Weed Optimization Algorithm (IWO)

The Invasive Weed Optimization Algorithm is based on the growth of weeds within the area. Initially, weeds are distributed randomly throughout the area. As weeds grow, they produce seeds in proportion to the fitness function value. The lowest and highest number of seeds are fixed arbitrarily, with the least fit weed producing the fewest seeds and the most fit weed producing the highest number of seeds. The number of seeds produced by each weed varies directly with the fitness function value. The seeds produce copies of the parent weed and then the weeds randomly change their locations with a normal distribution with zero mean but with a variable variance parameter that decreases as the number of iterations increases. The total number is kept constant after reaching the maximum plant colony size by removing the weakest weeds, [21].

2.2.1. Basic Concepts of Invasive Weed Algorithm

Mehrabian and Lucas (2006) introduced the Invasive Weed Optimization Algorithm, a simple numerical optimization approach based on colonized weeds. This approach is basic but successful in convergence. Optimize weed colony solutions by considering basic factors like seeding, growth, and competition. To run the IWO algorithm, we follow the steps below.

1. **Initialization:** The number of initial weeds W =(w_1, w_2,..., w_m), representing the empirical solution of the optimization problem, and is distributed randomly over the search space of dimension d, and the fitness function value is calculated for each weed.
2. **Reproduction:** Each seed grows into a blooming plant, which produces seeds according to its suitability. The amount of grass grains decreases linearly from S_{max} to S_{min}, as shown below.

$$seed_i = S_{min} + floor\left((S_{max} - S_{min}) \times \frac{f_i - f_{min}}{f_{max} - f_{min}}\right) \quad (2)$$

Where f_i is the fitness of the ith weed, $floor$ Shows that the seeds have been rounded to the closest whole number, S_{min}, and S_{max} denoted the maximum and minimum number of seeds, and f_{min} is the min values of the fitness function's min, f_{max} is the max value of fitness functions. Equation (2) shows the correlation between function value and number of weed seeds. Increasing the fitness function value causes the quantity of seeds to fluctuate between the maximum and minimum values.

3. **Spatial distribution:** The generated seeds were dispersed randomly in a d-dimensional search area using a natural distribution of random integers utilizing mean zero and coefficients of variance that decreased with the number of repetitions. The seeds produced are formed far away but close to the mother plant, which leads to the healthiest plants being grouped, and unsuitable plants being eliminated on time. The random deviation (σ) of the random function is generated by minimizing it from a predetermined initial value ($\sigma_{initial}$) to the final value (σ_{final}) in unit iterations, computed using Eq. at every time step.

$$\sigma_t = \frac{(T-t)^n}{T^n}\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \quad (3)$$

where n is the nonlinear modulation index, which is typically set to 2, and T is the most iterations that can be made, where the standard deviation at the current time step is denoted by σ_t.
To calculate the location of the new seeds, we apply the subsequent formula:

$$x_{son} = x_{parent} + randn(0,1) * \sigma_t \quad (4)$$

Where x_{parent} symbolizes the parents' site, x_{son} denotes the location of the sons, and $randn$ generates random numbers using the normal distribution (0,1).

4. **Competitive exclusion**: Due to rapid reproduction, after several iterations the number of plants produced in the colony reaches a maximum (P_{max}). In this step, a competitive mechanism is activated to eliminate unwanted plants with poor fitness function value and allow plants with better fitness to reproduce more seeds. This process continues until the maximum number of iterations is reached or other stopping criteria are met, and the plant with the best fitness function value is chosen as the optimal solution.

Implementing the IWO algorithm involves the following procedures, taking into account the aforementioned important phases.

1. Randomly initialize the weeds in the search space.
2. Calculate the fitness of each individual in the population.
3. Each member of the population creates seeds, as some members of the population with better fitness create more seeds (i.e., reproduction).
4. The seeds generated in the search area with random numbers are distributed normally with a mean equal to zero and with different variances (i.e. spatial dispersion).
5. We use competitive exclusion when the quantity of weeds approaches its limit.
6. Check stopping criteria. [22][23][24][25]

3. Methodology

A method for determining the approximate solution of SVIEs is offered. The work starts by defining the expansion's function. Additionally, SVIEs are transformed into unrestricted optimization issues. Furthermore, the discrete least squares weighted function and the fitness function are presented. Finally, the approximation algorithm for the SVIEs solution is introduced.

3.1. The Padé Expansion Approximation

Padé approximants are frequently used in the solution of mathematical problems. The approximate solution of the SVIEs can be represented using the expansion of Padé, [26].

$$\eta(t) \cong \eta_{appr}(t) = \frac{\Psi(t)}{\Phi(t)} = \frac{\sum_{m=0}^{N} \alpha_m t^m}{\sum_{m=0}^{M} \beta_m t^m} \qquad (5)$$

In this case, $t \in I = [t_0, t_n]$; $N + M = nVar$; and α_m, β_m are real coefficients that are part of the search space $[Var\ min, Var\ max]$. $\eta_{appr}(t)$ is the approximate solution, $\eta(t)$ is the exact solution, $\Phi(t) \neq 0\ \forall\ t \in I$.

3.2. Convert SVIEs into an Unconstrained Optimization Problem

Suppose that $\eta_1(t), \eta_2(t), \ldots, \eta_i(t)$ are approximate solutions of (SVIEs) and substitute into Eq. (6) into as an unconstrained optimization problem:

$$F(t) = \left|\eta_1(t) - f_1(t) - \int_0^t (k_1(t,s)\eta_1(s) + k`_1(t,s)\eta_2(s) + \cdots)ds\right| + \left|\eta_2(t) - f_2(t) - \int_0^t (k_2(t,s)\eta_1(s) + k`_2(t,s)\eta_2(s) + \cdots)ds\right| + \cdots \qquad (6)$$

F(t) is called the error function, and the optimal solution to the system of Volterra equations can be obtained when $F(t)$approaches zero. To minimize the value of $F(t)$, a quantitative criterion is used, which is called the efficiency function, represented here by the discrete least squares weight function, [18].

3.2.1. The Discrete Least Squares Weighted Function (DLSWF)

The DLSWF is obtained using the approach below: Divide the interval into N points: $t_0 = a, t_1, \ldots, t_n = b$, with $t_k = t_0 + h_k, \forall k = 1, \ldots n,$ and $h > 0, and$, [27][28]

$$DLSWF = \sqrt{\frac{\sum_{k=1}^{N} (F(t_k)^2}{N}} \qquad (7)$$

3.3. SVIEs-IWO Algorithm

This section presents the steps within the algorithm for solving SVIEs suggested, as described below:

1. Use the Padé expansion approximation to represent the approximate solution of the systems of Volterra integral equations as in Equation (5).
2. Convert the SVIEs to the implicit form using Eq. (6).
3. The FF is obtained as shown :

$$FF = DLSWF \qquad (8)$$

4. Set up the IWO parameters to calculate Padé expansion coefficients.
5. Use the IWO algorithm to minimize the fitness function.
6. Repeat step 5 until $FF < TOL$ or the maximum number of repetitions is achieved.

It is the procedure that may be evaluated by computing the Mean Absolute Error (MAE) for both the approximate solution $\eta_{appr}(t)$ and the precise solution $\eta(t)$.

$$MAE = \frac{\sum_{k=1}^{N} |\eta(t_k) - \eta_{appr}(t_k)|}{N} \qquad (9)$$

4. Numerical Results

In this part, we provide an approximate solution to SVIEs utilizing IWO and the padé approximant. Moreover, the convergence and stability of the algorithm are demonstrated.

4.1. Numerical Examples

Table 1. The table displays different examples of the Volterra integral equations system both linear and nonlinear, along with their precise solutions. The linear and nonlinear Volterra integral equations are respectively referred to as LSVIEs and NLSVIEs, [6][10][29][30].

Examples	SVIEs	Exact Solutions
LSVIEs1	$\eta_1(t) = t^2 - \frac{1}{12}t^5 + \int_0^t ((t-s)^2\eta_1(s) + (t-s)\eta_2(s))ds$ $\eta_2(t) = t^3 - \frac{1}{30}t^6 + \int_0^t ((t-s)^3\eta_1(s) + (t-s)^2\eta_2(s))ds$	$\eta_1(t) = t^2$ $\eta_2(t) = t^3$
LSVIEs2	$\eta_1(t) = \cos t + 2\sin t - 1 + \int_0^t (\cos(t-s)\,\eta_1(s) + \sin(t-s)\,\eta_2(s))ds$ $\eta_2(t) = t\cos t + \sin t - 1 + \int_0^t (\cos(t-s)\,\eta_2(s) + \sin(t-s)\,\eta_1(s))ds$	$\eta_1(t) = t + \sin t$ $\eta_2(t) = t - \cos t$
LSVIEs3	$\eta_1(t) = t\cos t - t\sin t - \sin t + 1 + \int_0^t (t\,\eta_1(s) + (s-1)\eta_2(s))ds$ $\eta_2(t) = t^2\cos t - t\sin t - t^2\sin t + t + \int_0^t (s\,\eta_2(s) + ts\,\eta_3(s))ds$ $\eta_3(t) = -5\cos t - 5t\sin t + t + 6 + \int_0^t (5s\,\eta_1(s) + (s-t)\eta_3(s))ds$	$\eta_1(t) = \cos t$ $\eta_2(t) = \sin t$ $\eta_3(t) = \sin t + \cos t$
NLSVIEs4	$\eta_1(t) = t^2 - \frac{1}{2} - \frac{2}{3}t^4 + \frac{\cos t^2}{2} + \int_0^t (s\sin(\eta_1(s)) + 2ts\,\eta_2(s))ds$ $\eta_2(t) = t - t\sin t - \frac{t^2\sin t^2}{2} + \int_0^t (t^2 s\cos(\eta_1(s)) + t\ \cos(\eta_2(s)))\,ds$	$\eta_1(t) = t^2$ $\eta_2(t) = t$
NLSVIEs5	$\eta_1(t) = 1 + t^2 - t^3 - \frac{1}{3}t^7 + \int_0^t (ts\,(\eta_1(s))^2 + ts\,(\eta_2(s))^2)ds$ $\eta_2(t) = 1 - t^2 - \frac{1}{3}t^4 + \int_0^t ((t-s)(\eta_1(s))^2 - (t-s)\,(\eta_2(s))^2)ds$	$\eta_1(t) = 1 + t^2$ $\eta_1(t) = 1 - t^2$
NLSVIEs6	$\eta_1(t) = \tan^{-1}t + \int_0^t (\frac{((\eta_2(s))^2 + (\eta_3(s))^2)}{1 + (\eta_1(s))^2})ds$ $\eta_2(t) = t\cos t - \ln(1+t^2) + \sqrt{2}\int_0^t \left(\frac{\sqrt{(\eta_1(s))^2 + (\eta_2(s))^2 + (\eta_3(s))^2}}{1 + (\eta_2(s))^2 + (\eta_3(s))^2}\right)ds$ $\eta_3(t) = 1 + \cos t(\frac{t^2}{3} - 1) + \frac{1}{3}\int_0^t (\eta_2(s) + \eta_1(s)\,\eta_3(s))ds$	$\eta_1(t) = t$ $\eta_2(t) = t\cos t$ $\eta_3(t) = t\sin t$

Table 2. The parameter values used in all examples

Parameter	S_{min}	S_{max}	σ_{final}	TOL	$Maxit$	$npop$
Value	0	5	0.0001	1e-10	200	100

Table (2) shows the parameters and inputs for all of the examples. The algorithm is tested 16 times to ensure its reliability. The Matlab R2023a software package is used to implement the technique. The device is an HP laptop with an Intel(R) Core (TM) i7-6820HQ CPU @ 2.70GHz, 16.00GB of RAM, and the Windows 10 Pro 64-bit operating system.

4.2. SVIEs Solution

Tables 3 and 4 show the coefficients of approximate SVIEs solutions for all examples produced using the IWO algorithm.

Table 3. The values of variables of SVIEs of two unknown functions obtained by the SVIEs-IWO algorithm Where $(nVar = 20)$, $[VarMin, VarMax] = [-1,1]$

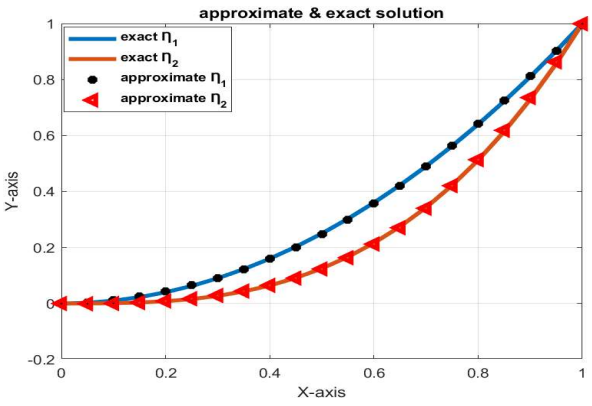
SVIEs	coff	m = 0	$m = 1$	m = 2	m = 3	$m = 4$
LSVIEs1	α_{m1}	0.001944	-0.051314	-0.83238	-0.46608	-0.124237
	β_{m1}	-0.93610	-0.73244	0.49346	-0.08061	-0.225139
	α_{m2}	-0.000067	-0.03489	0.32039	0.335059	0.770305
	β_{m2}	0.978330	0.87917	-0.61302	-0.55448	0.701726
LSVIEs2	α_{m1}	0.002338	0.887847	0.86779	0.79758	0.37921
	β_{m1}	0.46458	0.358819	0.579419	0.037832	0.155505
	α_{m2}	-0.971106	0.09886	0.779934	0.701432	0.642769
	β_{m2}	0.971081	0.97026	0.00575	0.89648	-0.06604
NLSVIEs4	α_{m1}	0.002184	-0.02291	0.97369	0.72919	0.92228
	β_{m1}	0.964129	0.96412	0.43591	0.912358	-0.457434
	α_{m2}	-0.00179	0.928919	0.415341	0.92059	-0.190629
	β_{m2}	0.823746	0.968289	0.108159	0.024336	0.156110
NLSVIEs5	α_{m1}	0.994389	0.871444	0.825308	0.47022	0.701765
	β_{m1}	0.999732	0.717618	0.452125	-0.94596	0.714807
	α_{m2}	-0.99942	-0.94545	0.63607	0.530088	0.778195
	β_{m2}	-0.99941	-0.91701	-0.73761	0.678148	-0.60698

Table 4. The values of variables of SVIEs of tree unknown functions obtained by SVIEs- IWO algorithm Where $(nVar = 24)$, $[VarMin, VarMax] = [-1,1]$

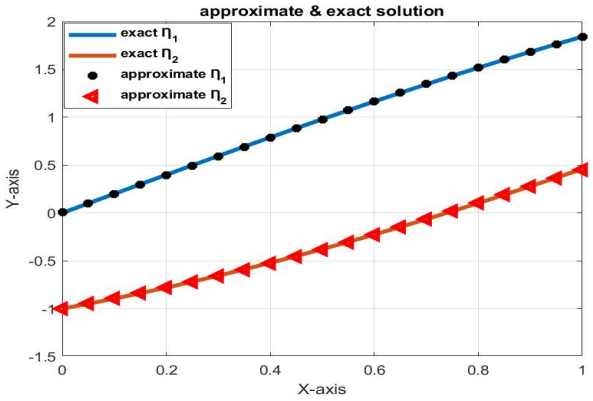
SVIEs	coff	m = 0	$m = 1$	m = 2	m = 3
LSVIEs3	α_{m1}	-1	-0.98202	-0.09897	0.898738
	β_{m1}	-1	-0.95823	-0.747478	0.522604
	α_{m2}	-0.01459	-0.769123	-0.824254	0.339015
	β_{m2}	-0.999526	-0.064728	-0.692811	0.247356
	α_{m3}	0.856706	0.4760759	0.655320	0.289834
	β_{m3}	0.8526845	-0.253517	0.7734329	0.2588448
NLSVIEs6	α_{m1}	-0.000037	-0.896159	0.3822662	-0.99110
	β_{m1}	-0.85366	0.1293778	-0.565614	-0.216348
	α_{m2}	-0.005513	-0.805460	-0.938811	0.991414
	β_{m2}	-0.920776	-0.40863	-0.373068	0.296172
	α_{m3}	-0.004765	0.007122	-0.576201	-0.984296
	β_{m3}	-0.88265	-0.173732	-0.487331	-0.324409

Table 5. It displays the mean absolute error (MAE) between the exact solution and the approximate solution and the time taken to reach the optimal solution for systems of linear and nonlinear Volterra integral equations of the second kind resulting from the proposed algorithm.

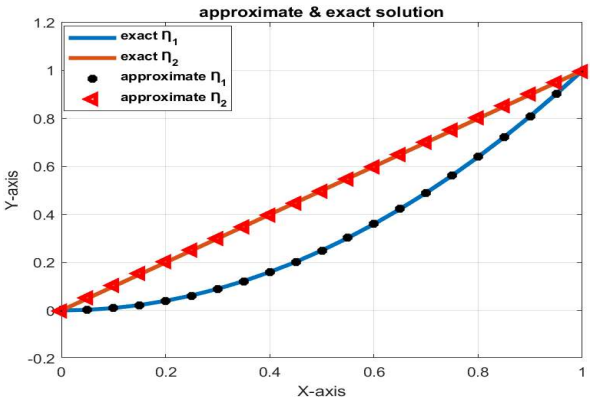
SVIEs	MAE		Time(s)
LSVIEs1	MAE1	0.001928	169.7472
	MAE2	0.0021	
LSVIEs2	MAE1	0.00177	169.8697
	MAE2	0.00286	
LSVIEs3	MAE1	0.00217	197.3125
	MAE2	0.00330	
	MAE3	0.00290	
NLSVIEs4	MAE1	0.00083	167.0785
	MAE2	0.00225	
NLSVIEs5	MAE1	0.003611	189.1598
	MAE2	0.001263	
NLSVIEs6	MAE1	0.00136	267.0285
	MAE2	0.0010	
	MAE3	0.00442	



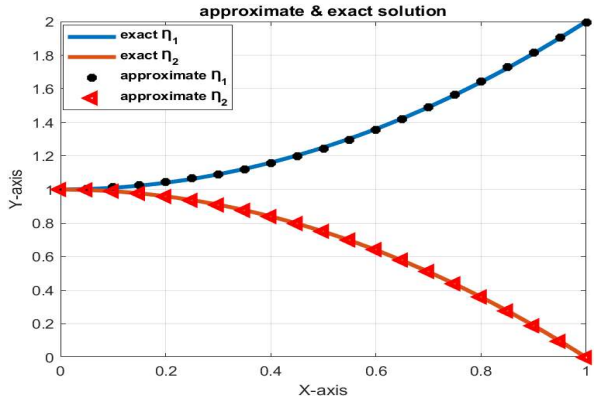
(a) LSVIEs1



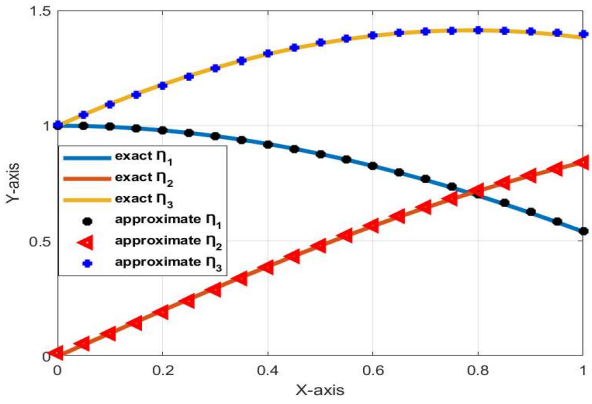
(b) LSVIEs2



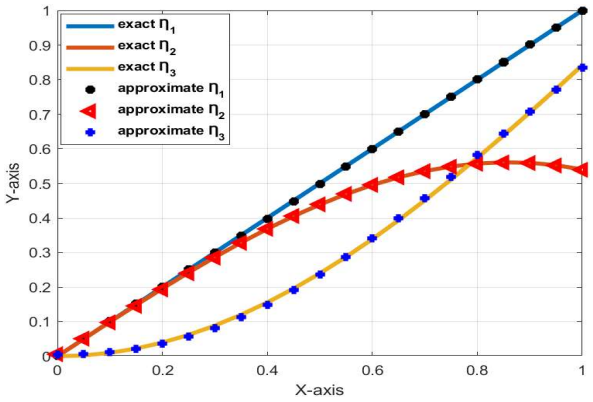
(c)NLSVIEs4



(d)NLSVIEs5



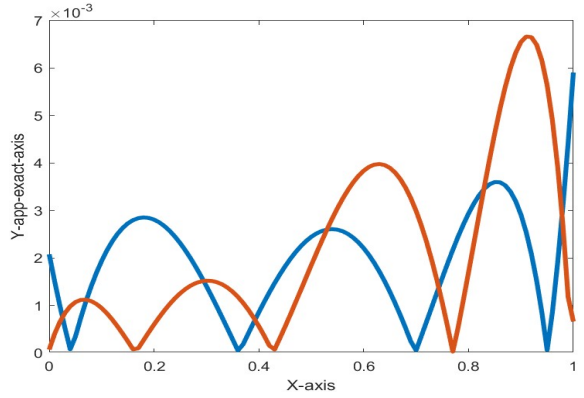
(e)LSVIEs3



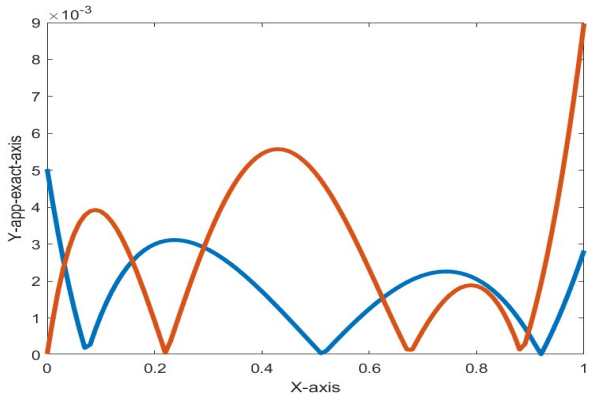
(f)NLSVIEs6

Figure 1. A comparison of exact and approximate solutions.

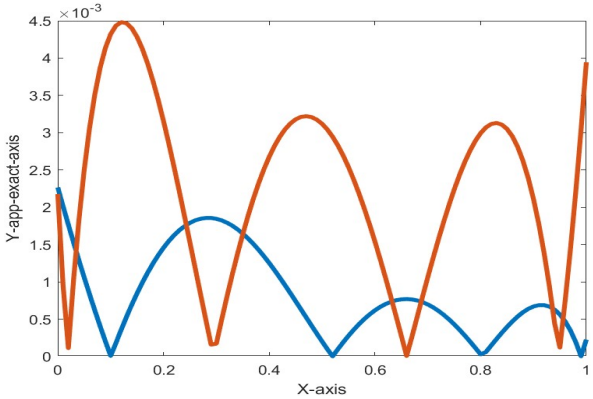
By looking at the diagram above, we can conclude that the algorithm SVIEs- IWO is capable of finding approximate solutions that closely resemble the exact solution.



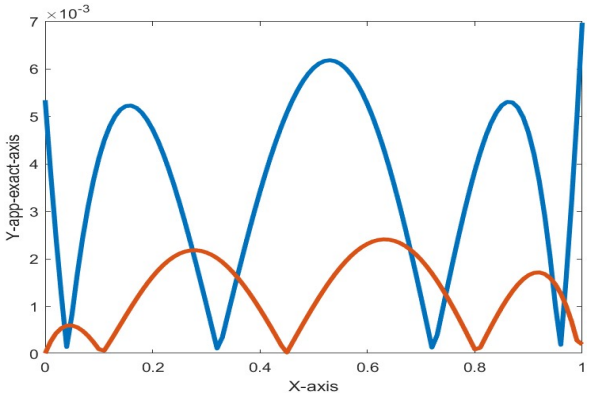
(a)LSVIEs1



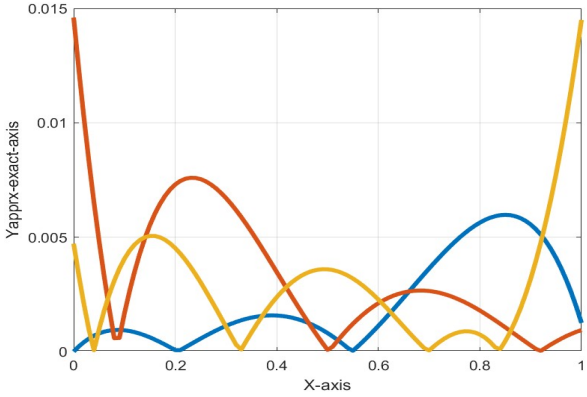
(b)LSVIEs2



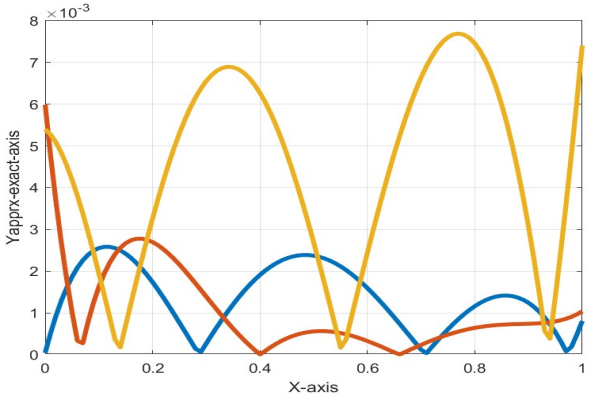
(c)NLSVIEs4



(d)NLSVIEs5



(e)LSVIEs3



(f)NLSVIEs6

Figure 2. Mean Absolute Error

Figure 3. It is shown that SVIEs-IWO converges quickly over 200 iterations for all samples. We observed that the solution stabilizes at 150 iterations in most examples.

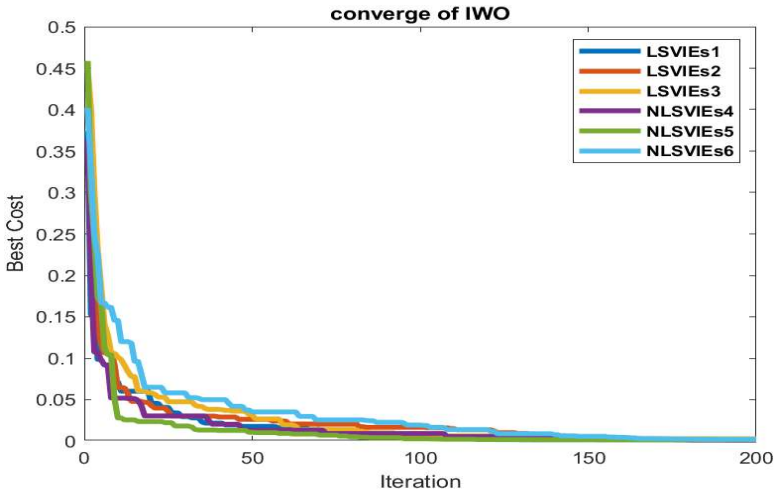


Figure 3. SVIEs-IWO converges

5. CONCLUSIONS

A combination of the Invasive Weed Optimization Algorithm with Padé expansion approximation is applied to solve systems of Volterra integral equations. The system is converted into an unconstrained optimization problem. This study reveals that the SVIEs-IWO algorithm is effective in solving systems of Volterra integral equations of the second kind. Moreover, the convergence and stability of the algorithm is also confirmed. Furthermore, it is recommended that this approach can be utilized to solve systems of Volterra integral-differential equations.

6. ACKNOWLEDGMENT

The authors write to appreciate the University of Mosul and the Special thanks to the College of Education for Pure Sciences for their assistance in completing this work.

7. REFERENCES

[1] S. Banerjea and B. N. Mandal, *Integral Equations and Integral Transforms*. Singapore: Springer Nature Singapore, 2023. doi: 10.1007/978-981-99-6360-7.

[2] K. M. V. Balakumar and Department, "Numerical solution of systems of linear Volterra integral equations using block-pulse functions," *Malaya J. Mat.*, no. 1, pp. 77–84, Sep. 2013.

[3] T. Allahviranloo and A. Esfandiari, *A Course on Integral Equations with Numerical Analysis*. in Mathematical Engineering. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-85350-1.

[4] A. M. Wazwaz, "A first course in integral equations, second edition," *A First Course in Integral Equations, Second Edition*. pp. 1–312, 2015. doi: 10.1142/9570.

[5] M. Rahman, *Integral equations and their applications*, vol. 45, no. 08. 2008. doi: 10.5860/choice.45-4442.

[6] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 7, pp. 1501–1529, Jul. 2020, doi: 10.1007/s13042-019-01053-x.

[7] F. Mirzaee, "Numerical computational solution of the linear Volterra integral equations system via rationalized Haar functions," *J. King Saud Univ. - Sci.*, vol. 22, no. 4, pp. 265–268, Oct. 2010, doi: 10.1016/j.jksus.2010.05.010.

[8] D. A. Maturi and H. M. Malaikah, "Numerical Solution of System of Three Nonlinear Volterra Integral Equations Using Implicit Trapezoidal," *Journal of Mathematics Research*, vol. 10, no. 1. p. 44, 2017. doi: 10.5539/jmr.v10n1p44.

[9] A. Tahmasbi and O. S. Fard, "Numerical solution of linear Volterra integral equations system of the second kind," *Applied Mathematics and Computation*, vol. 201, no. 1–2. pp. 547–552, 2008. doi: 10.1016/j.amc.2007.12.041.

[10] M. M. Mustafa, "Numerical Solutions for System of Volterra Integral Equations Using Spline Functions," 2020. doi: 10.13140/RG.2.2.30590.82243.

[11] Ó. Oliveira, T. Matos, and D. Gamboa, *Learning and Intelligent Optimization*, vol. 11968. in Lecture Notes in Computer Science, vol. 11968. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-38629-0.

[12] S. Mirjalili, *Evolutionary Algorithms and Neural Networks*, vol. 780. in Studies in Computational Intelligence, vol. 780.

Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-319-93025-1.

[13] D. Simon, *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*. 2013.

[14] B. Zolghadr-Asli, *Computational Intelligence-based Optimization Algorithms*. Boca Raton: CRC Press, 2023. doi: 10.1201/9781003424765.

[15] D. Gutierrez-Navarro and S. Lopez-Aguayo, "Solving ordinary differential equations using genetic algorithms and the Taylor series matrix method," *J. Phys. Commun.*, vol. 2, no. 11, p. 115010, Nov. 2018, doi: 10.1088/2399-6528/aaedd2.

[16] 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.

[17] Reham Alazzawi, "Solving Systems of Ordinary Differential Equations Using Particle Swarm Optimization and Genetic algorithm Based on Padé Approximants." pp. 53–68, 2022.

[18] 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.

[19] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4. pp. 355–366, 2006. doi: 10.1016/j.ecoinf.2006.07.003.

[20] A.-M. Wazwaz, *Linear and Nonlinear Integral Equations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-21449-3.

[21] T. Jayabarathi, A. Yazdani, and V. Ramesh, "Application of the invasive weed optimization algorithm to economic dispatch problems," *Frontiers in Energy*, vol. 6, no. 3. pp. 255–259, 2012. doi: 10.1007/s11708-012-0202-1.

[22] W. Abdulelah Qasim and B. Ahmed Mitras, "A Hybrid Algorithm Based on Invasive Weed Optimization Algorithm and Grey Wolf Optimization Algorithm," *Int. J. Artif. Intell. Appl.*, vol. 11, no. 1, pp. 31–44, Jan. 2020, doi: 10.5121/ijaia.2020.11103.

[23] S. Mekni and B. C. Fayech, "A Modified Invasive Weed Optimization Algorithm for MultiObjective Flexible Jobe Shop Scheduling Problems." pp. 51–60, 2014. doi: 10.5121/csit.2014.41106.

[24] W. A. Qasim and B. A. Mitras, "A Suggestion Algorithm Instituted on Invasive Weed Optimization Algorithm and Bat Optimization Algorithm," *OALib*, vol. 07, no. 06. pp. 1–11, 2020. doi: 10.4236/oalib.1106438.

[25] B. Xing and W.-J. Gao, *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, vol. 62. in Intelligent Systems Reference Library, vol. 62. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-03404-1.

[26] J. P. G.-M. GEORGE A. BAKER, *ENCYCLOPEDIA OF MATHEMATICS AND ITS APPLICATIONS Pade Approximants, second edition*. 1996.

[27] A. Cohen and G. Migliorati, "SMAI-JCM SMAI Journal of Computational Mathematics Optimal weighted least-squares methods Optimal weighted least-squares methods," *SMAI Journal of Computational Mathematics*, vol. 3. pp. 181–203, 2017.

[28] Israa I. M. Sheekhoo & Azzam S. Y. Aladool, "Application of Cultural Algorithm and Padé Approximants to Solve Volterra Integral Equation." 2023.

[29] M. I. Berenguer, D. Gmez, A. I. Garralda-Guillem, M. Ruiz Galn, and M. C. Serrano Prez, "Biorthogonal systems for solving Volterra integral equation systems of the second kind," *Journal of Computational and Applied Mathematics*, vol. 235, no. 7. pp. 1875–1883, 2011. doi: 10.1016/j.cam.2010.07.011.

[30] A. R. Yaghoobnia and R. Ezzati, "Using Bernstein multi-scaling polynomials to obtain numerical solution of Volterra integral equations system," *Comput. Appl. Math.*, vol. 39, no. 3, p. 170, Sep. 2020, doi: 10.1007/s40314-020-01198-4.

استخدام خوارزمية تحسين الأعشاب الغازية مع تقريب بادي لحل نظام من معادلات فولتيرا التكاملية

محمد جمال حسن المولى [1]* ، عزام صلاح الدين يونس العدول [2]

[1,2] قسم الرياضيات، كلية التربية للعلوم الصرفة، جامعة الموصل، الموصل، العراق

المستخلص

توصف تطبيقات الهندسة والفيزياء رياضيًا كنظام من معادلات فولتيرا التكاملية . وعلى الرغم من وجود العديد من التقنيات العددية لحل معادلات فولتيرا التكاملية، إلا أن تطوير خوارزميات التي تكون أكثر استقرارًا وكفاءة لا يزال يمثل تحديًا امام الباحثين. ان في هذا العمل، تم اقتراح خوارزمية لحل نظام معادلات فولتيرا التكاملية الخطية وغير الخطية من النوع الثاني. و قد تم دمج خوارزمية تحسين الأعشاب الضارة (IWO) مع التوسع التقريبي لبادي. و نظرًا لأن الحل يتم تمثيله كدوال ذات انواع مختلفة، يتم استخدام تقريب بادي والذي يعرف كتوسيع كسري و الذي بدوره يمثل حل نظام من معادلات فولتيرا التكاملية و يعطي نتائج عالية الدقة. و من ثم يتم تحويل معادلات فولتيرا التكاملية إلى مشكلة تحسين غير مقيدة. بعد ذلك، يتم تقدير دالة وزن المربعات الصغرة المتقطعة لتقليل قيمة دالة اللياقة. حيث يتم تطبيق هذه الخوارزمية لحل مجموعة متنوعة من الأمثلة الخطية وغير الخطية ومقارنة حلولها بالحلول الدقيقة. يكون أداء الخوارزمية اكثر كفاءة و يحقق نتائج دقيقة من حيث التقارب والاستقرار.