ORIGINAL RESEARCH PAPER

Numerical analysis of the Volterra differential equations via a combination of evolutionary algorithms with moving least squares and finite difference methods

Mohammad Javad Mahmoodabadi

Abstract

In this paper, the Volterra differential equations governing on the predator-prey dynamics is numerically solved through the combination of evolutionary algorithms with Moving Least Squares as well as the Finite Difference Method. Among the numerous evolutionary algorithms, the Genetic Algorithms and Particle Swarm Optimization, as two powerful and effective approaches, are selected. The penalty method is chosen to impose the initial conditions on the objective function, while the Moving Least Squares is implemented to interpolate the derivatives in the differential equations. In order to illustrate the effectiveness of the Moving Least Squares, the results are compared with those obtained via the Finite Difference Method based evolutionary algorithms. Moreover, the results are associated with those of a numerical method and an analytical scheme to evaluate the obtained populations of the prey and the predator. These comparisons clearly illustrate the reliability and efficiency of the proposed methodology to solve nonlinear differential equations.

1 INTRODUCTION

Most governing equations of real world phenomena are enormously nonlinear; and it is challenging to find their solutions [1–5]. Specially, Volterra differential equations governed on the predator-prey dynamics belong to these drawbacks that many researchers have been attracted to solve them [6–16]. Some notable studies on Volterra differential equations are as follows. The stability analysis and convergence studies of Volterra integro-differential equations were conducted based on the convolution test equation and theoretical results and using the corresponding recurrence relation and stability matrix [17]. The nonlinear stability properties of the discontinuous Galerkin method were investigated for stiff Volterra functional differential equations [18]. Spectral collocation methods were developed for weakly singular Volterra integro-differential equations, and convergence analysis results illustrated that the global convergence order depends on the regularities of the kernels functions [19]. Lyapunov functions combined with the Laplace transform were utilized to gain boundedness results regarding the solutions of the nonlinear Volterra integro-differential equations [20]. The multi-wavelets Galerkin method was used for linear and nonlinear second order Volterra integro-differential equations, and the problem was reduced to a set of algebraic equations using the operational matrices of integration and the wavelet transform matrix [21].

On the other hand, the moving least squares approach is implemented to reconstruct continuous functions from a set of unorganized point samples [22–26]. As another discretization scheme, the finite difference method converts linear/nonlinear differential equations into a system of linear/non-linear algebraic equations [27–35].

Besides, evolutionary algorithms inspired by biological evolution are primarily suited for numerical optimization problems because they often found well approximating solutions for all types of challenges. In the recent years, numerous types of these schemes have been introduced by scientists and researchers, to name but a few; genetic algorithm [39], particle swarm optimization [40], ant colony [36], artificial bee colony [37], bat algorithm [38], firefly algorithm [46], grey wolf optimizer [47], invasive...
weed optimization [48], shuffled frog leaping algorithm [49],
spider algorithm [50], whale optimization algorithm [51] and
team game algorithm [52].

The motivation of this paper is to extend two evolution-
ary algorithms, namely the genetic algorithm [39] and particle
swarm optimization [40], to solve the predator-prey dynamics.
To reach this goal, the penalty method is used to impose the
boundary conditions, and the MLS and FDM are implemented
to approximate the derivatives appeared in the equation. The
success and effectiveness of the introduced methodology are
shown via comparing its results with reported ones in literature.

The rest of the paper is organized as follows. The governing
equations are briefly represented in Section 2. The moving least
squares approach is formulated in Section 3. The PSO and GA
are described Sections 4 and 5, respectively. Discretization of
the governing equations to provide a proper objective function
is explained in Section 6. Simulation results and discussions are
given in Section 7. Finally, Section 8 concludes the findings of
this paper.

2 GOVERNING EQUATIONS

There are some rabbits and foxes living together that foxes eat
the rabbits and rabbits eat clover, and there is an increase and
decline in their number. The governing relations to this prob-
lem known as Volterra differential equation could be stated as

de

\[ \frac{d\mu (\tau)}{d\tau} = \mu (\tau) (\alpha - \beta \delta (\tau)) \quad \text{that} \quad \mu (0) = \mu_0, \quad (1) \]

\[ \frac{d\delta (\tau)}{d\tau} = -\delta (\tau) (\gamma - \theta \mu (\tau)) \quad \text{that} \quad \delta (0) = \delta_0, \quad (2) \]

where, \( \mu (\tau) \) and \( \delta (\tau) \) respectively represent the populations
of the rabbits and foxes at time \( \tau \). Further, \( \alpha, \beta, \gamma \) and \( \theta \) are
constant parameters dependent to the environment conditions.

3 MOVING LEAST SQUARES

The moving least squares approximation is formulated in this
section. At first, the shape function \( \chi (\zeta_i) \) for a selected point
\( \zeta_i = [\mu (\tau), \delta (\tau)] \in \Psi \) is defined by:

\[ \chi (\zeta_i) = \omega^T (\zeta_i) \Gamma^{-1} (\zeta) \mathbf{E} (\zeta) \quad \text{for} \quad \Psi_d, \quad (3) \]

where, \( \omega^T (\zeta_i) = [1, \zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_m] \) denotes the monomial
basis vector, and \( m \) is its order. Matrices \( \Gamma (\zeta) \) and \( \mathbf{E} (\zeta) \) are
calculated by the following relations.

\[ \Gamma (\zeta) = \Omega^T \Lambda \Omega \quad \text{and} \quad \mathbf{E} (\zeta) = \Omega^T \Lambda, \quad (4) \]

where, \( \Omega = [\omega (\zeta_1), \omega (\zeta_2), \ldots, \omega (\zeta_n)]^T \) signifies the monomial
basis matrix; \( n \) represents to the number of nodes in the sup-
porting domain \( \Psi_d \) of the global domain \( \Psi \); and \( \Lambda \) defines the
diagonal weight function matrix whose \( j^{th} \) diagonal member is
calculated by the following equation:

\[ \lambda_j (\zeta_i) = \begin{cases} 1 - 6 \left( \frac{\rho_j}{\rho} \right)^2 + 8 \left( \frac{\rho_j}{\rho} \right)^3 - 3 \left( \frac{\rho_j}{\rho} \right)^4 & 0 \leq \rho_j \leq \rho, \\ 0 & \rho_j > \rho \end{cases} \quad (5) \]

where, \( j = 1, 2, \ldots, n, \rho_j \) denotes the distance between points
\( \zeta_i \) and \( \zeta_j \); and \( \rho \) represents the size of the supporting domain.
After calculation of the shape function \( \chi (\zeta_i) \), the unknown value \( F^0 (\zeta_i) \) is approximated from the known nodal values \( \hat{F} (\zeta_i) \)
as bellow [42].

\[ F^0 (\zeta_i) = \chi (\zeta_i) \hat{F} (\zeta_i). \quad (6) \]

4 PARTICLE SWARM OPTIMIZATION

In the particle swarm optimization algorithm, the position of
each particle is enhanced with respect to its previous best pos-
tion and the position of the best particle among population. The
mathematical equations of the PSO are generally represented as
follows:

\[ \vec{V}_i (t_1 + 1) = \mathbf{G} \vec{V}_i (t_1) + \epsilon_1 \vec{R}_1 \left( \vec{p}_{best} (t_1) - \vec{V}_i (t_1) \right) + \epsilon_2 \vec{R}_2 \left( \vec{g}_{best} (t_1) - \vec{V}_i (t_1) \right), \quad (7) \]

\[ \vec{V}_i (t_1 + 1) = \vec{V}_i (t_1) + \vec{V}_i (t_1 + 1), \quad (8) \]

where, \( \vec{p}_{best} (t_1) \) and \( \vec{V}_i (t_1) \) describe the position and velocity of particle
\( i \) at certain iteration \( t_1 \), respectively. Moreover, \( \epsilon_1 \) and \( \epsilon_2 \) are
coefficients that express the tendency of the particle to the per-
sonal and social successes, respectively. \( \vec{R}_1 \) and \( \vec{R}_2 \) are random
vectors in interval [0,1], and \( \mathbf{G} \) namely inertia weight illustrates
the effect of the particle velocity at the previous iteration on the
next one. Furthermore, \( \vec{g}_{best} (t_1) \) and \( \vec{p}_{best} (t_1) \) are personal
and global best positions of particle \( i \) and the swarm, respec-
tively [43]. In this paper, the tendency coefficients are regarded as \( \epsilon_1 = \epsilon_2 = 3 \); and the inertia weight is calculated through the
following adaptive relation [44].

\[ \mathbf{G} (r) = \frac{1}{1 + 1.5r^{-2.6r}}, \quad (9) \]

where, \( r \) presents the evolutionary factor and could be
calculated by the following equation.

\[ r = \frac{s_{global} - s_{min}}{s_{max} - s_{min}}, \quad s_i = \frac{1}{N - 1} \sum_{j=1}^{N} \sqrt{\sum_{k=1}^{D} \left( \zeta_{ik} - \bar{\zeta}_{ik} \right)^2}, \quad (10) \]

where, \( s_i \) is the mean distance of particle \( i \) to all other particles.
\( N \) and \( D \) are the population size and the number of dimensions,
respectively. \( s_i \) for the global best particle is shown as \( s_{global} \). It
is clear that by comparing \( s_1 \), the maximum and minimum distances \( (s_{\text{max}} \text{ and } s_{\text{min}}) \) would be determined.

5 | GENETIC ALGORITHM

The genetic algorithm is considered as an approach for solving the optimization problems based on the biological evolution via modification of a population of individual solutions, repeatedly. At each iteration, individuals are chosen randomly from the current population (as parents) and employed to produce the children for the next generation. The main operators of the real coded genetic algorithm utilized in this work could be described as follows.

- Crossover: This operator selects two chromosomes from the mating pool, randomly. Then, it creates two offspring from the selected parents, and the generated offspring are replaced by the parents. Let \( \vec{\xi}_i (\vec{\delta}) \) and \( \vec{\xi}_j (\vec{\delta}) \) represent two random selected chromosomes, and \( \vec{\xi}_i (\vec{\delta}) \) has the smaller fitness value, the crossover formula could be written as follows:

\[
\vec{\xi}_i (\vec{\delta}+1) = \vec{\xi}_i (\vec{\delta}) + R_1 (\vec{\xi}_i (\vec{\delta}) - \vec{\xi}_j (\vec{\delta})) \quad (11)
\]
\[
\vec{\xi}_j (\vec{\delta}+1) = \vec{\xi}_j (\vec{\delta}) + R_2 (\vec{\xi}_j (\vec{\delta}) - \vec{\xi}_i (\vec{\delta})) \quad (12)
\]

where, \( R_1 \) and \( R_2 \in [0, 1] \) are random vectors. After calculation of Equations (11) and (12), the superior offspring between \( \vec{\xi}_i (\vec{\delta}) \) and \( \vec{\xi}_i (\vec{\delta}+1) \) should be selected.

- Mutation: The mutation operation causes some varieties in the population and decreases the probability of convergence towards local optima. Let \( \vec{\xi}_i (\vec{\delta}) \) represents a randomly selected chromosome, then the mutation formula used in this algorithm is defined as:

\[
\vec{\xi}_i (\vec{\delta}+1) = \vec{\xi}_i (\vec{\delta}) + R (\vec{\xi}_{\text{max}} - \vec{\xi}_{\text{min}}) \quad (13)
\]

where, \( \vec{\xi}_{\text{max}} \) and \( \vec{\xi}_{\text{min}} \) are upper and lower bounds related to the search domain, respectively. Moreover, \( R \in [0, 1] \) represents a random vector \( [39] \).

6 | DISCRETIZATION OF THE GOVERNING EQUATIONS AS THE OBJECTIVE FUNCTION

In this section, the objective function of the problem is identified for minimization by the detailed PSO algorithm. Suppose that \( \tau_i \) denotes nodal position \( i \) on the problem domain, and \( \mu_{\alpha}(\tau_i) \) and \( \delta_{\beta}(\tau_i) \) respectively indicate the numerical values of the populations of the rabbits and the foxes at \( \tau_i \), computed via the regarded approaches. Using Equation (6), Equations (1) and (2) could be rewritten for node \( i \) as follows.

\[
\frac{d\chi(\tau_i)}{d\tau} \mu_{\alpha}(\tau_i) - \mu_{\alpha}(\tau_i) (\alpha - \beta \delta_{\beta}(\tau_i)) = 0 \quad (14)
\]
\[
\frac{d\chi(\tau_i)}{d\tau} \delta_{\beta}(\tau_i) + \delta_{\beta}(\tau_i) (\gamma - \theta \mu_{\alpha}(\tau_i)) = 0 \quad (15)
\]

that, \( \mu(0) = \mu_0 \) and \( \delta(0) = \delta_0 \).

The summation of absolute values of these equations for all nodal points generates the first part of the regarded objective function.

\[
\sum_{i=1}^{M} \left\{ \left| \frac{d\chi(\tau_i)}{d\tau} \mu_{\alpha}(\tau_i) - \mu_{\alpha}(\tau_i) (\alpha - \beta \delta_{\beta}(\tau_i)) \right| + \left| \frac{d\chi(\tau_i)}{d\tau} \delta_{\beta}(\tau_i) + \delta_{\beta}(\tau_i) (\gamma - \theta \mu_{\alpha}(\tau_i)) \right| \right\} = 0, \quad (16)
\]

where, \( M \) represents the total number of the nodes. If the penalty method is utilized to impose the initial conditions, then:

\[
\sum_{i=1}^{M} \left\{ \left| \frac{d\chi(\tau_i)}{d\tau} \mu_{\alpha}(\tau_i) - \mu_{\alpha}(\tau_i) (\alpha - \beta \delta_{\beta}(\tau_i)) \right| + \left| \frac{d\chi(\tau_i)}{d\tau} \delta_{\beta}(\tau_i) + \delta_{\beta}(\tau_i) (\gamma - \theta \mu_{\alpha}(\tau_i)) \right| + \Pi_1 |\mu_{\alpha}(\tau_1) - \mu_0| + \Pi_2 |\delta_{\beta}(\tau_1) - \delta_0| = 0 \quad (17)
\]

where, \( \Pi_1, \Pi_2 \gg 1 \) are the penalty parameters. Besides, if the backward FDM is implemented to calculate the derivative terms, the following formulation for the objective function is obtained.

\[
\sum_{i=1}^{M} \left\{ \left| \mu_{\alpha}(\tau_i) - \mu_{\alpha}(\tau_{i-1}) \right| \left( \tau_i - \tau_{i-1} \right) - \mu_{\alpha}(\tau_i) (\alpha - \beta \delta_{\beta}(\tau_i)) \right| + \left| \delta_{\beta}(\tau_i) - \delta_{\beta}(\tau_{i-1}) \right| \left( \tau_i - \tau_{i-1} \right) + \delta_{\beta}(\tau_i) (\gamma - \theta \mu_{\alpha}(\tau_i)) \right| + \Pi_1 |\mu_{\alpha}(\tau_1) - \mu_0| + \Pi_2 |\delta_{\beta}(\tau_1) - \delta_0| = 0 \quad (18)
\]

7 | NUMERICAL RESULTS AND COMPARISONS

In this section, the introduced strategy based on the PSO/GA and MLS/FDM is applied to analyse the nonlinear predator-prey dynamics in several cases. The initial population size for the PSO and GA is considered as 100; their maximum number of iterations is set as 10,000; the radius of the supporting domain is regarded as \( \rho = (s_{\text{max}} - s_{\text{min}}) + 0.1 \); 6 nodes are regularly distributed on the defined domains; and the penalty parameters are set at \( \Pi_1 = \Pi_2 = 100 \).
**FIGURE 1** Convergence trajectories of the MLS based PSO, FDM based PSO, MLS based GA and FDM based GA methods for Example 1

**FIGURE 2** Prey ($\mu(\tau)$) and predator ($\delta(\tau)$) populations obtained by the exact solution, RKM [45], MLS based PSO, FDM based PSO, MLS based GA and FDM based GA for Example 1

### 7.1 Example 1

In this example, the constant parameters in the governing equations are considered as $\alpha = \gamma = 1$ and $\beta = \theta = 0$; and the initial conditions are respected as $\mu_0 = \delta_0 = 10$. Therefore, the exact solution could be easily obtained as follows:

$$
\mu(\tau) = 10e^\tau \quad \text{and} \quad \delta(\tau) = 10e^{-\tau}
$$

The global domain is considered as a line geometry from $\tau_{\min} = 0$ to $\tau_{\max} = 1$. The convergence trajectories of the MLS based PSO, MLS based GA, and FDM based GA methods are depicted in Figure 1. For this case, the found numerical solutions by the four mentioned methods are compared together and with those of the Runge–Kutta Method (RKM) [45] and the exact solutions in Figure 2. These comparisons obviously demonstrate that the MLS based PSO and
RKM have more accuracy to solve the nonlinear Volterra differential equation.

### 7.2 Example 2

In this case, in order to have a reasonable comparison with the method presented in Ref. [8], the fixed values in the governing equations are regarded similar to this reference as \( \alpha = \beta = \theta = 1 \) and \( \gamma = 0.1 \); and the initial conditions are appreciated as \( \mu_0 = 14 \) and \( \delta_0 = 18 \). In Ref. [8], the problem has been solved through Homotopy Perturbation Method (HPM) and the global domain is considered as a line geometry from \( \tau_{\text{min}} = 0 \) to \( \tau_{\text{max}} = 0.125 \). The convergence diagrams of the evolutionary algorithms are depicted in Figure 3. The numerical results obtained by these methods are compared together and with those of the RKM [45] and HPM [8] in Figure 4. This figure shows that MLS based PSO has most agreement with RKM for the prey population, while FDM based PSO has less error for the predator population. These comparisons obviously display the ability and competence of the methods based PSO algorithm to solve the considered nonlinear differential equation.
FIGURE 5  Convergence trajectories of the MLS based PSO, FDM based PSO, MLS based GA and FDM based GA methods for Example 3

FIGURE 6  Prey ($\mu(\tau)$) and predator ($\delta(\tau)$) populations obtained by the RKM [45], HPM [8], MLS based PSO, FDM based PSO, MLS based GA and FDM based GA for Example 3

7.3  Example 3

To prepare another comparison, the fixed parameters in the governing equations are regarded similar to Ref. [8] as $\gamma = \beta = \theta = 1$ and $\alpha = 0.1$; and the initial conditions are fixed at $\mu_0 = 14$ and $\delta_0 = 18$. The convergence charts of the MLS based PSO, MLS based GA, FDM based PSO and FDM based GA approaches are portrayed in Figure 5. The explanations obtained by these methods are compared based schemes with those of the RKM [45] and HPM [8] in Figure 6 that clearly verifies the ability of the PSO algorithm to find the answers of the prey and predator population problem.

7.4  Example 4

For this example, the constant parameters and the initial conditions are regarded as $\gamma = \beta = \theta = 1$, $\alpha = 0.1$, $\mu_0 = 14$ and
δ₀ = 18. The problem domain is measured as a line geometry from \( τ_{min} = 0 \) to \( τ_{max} = 0.2 \). The convergence illustrations of the different approaches based on the GA and PSO are exhibited in Figure 7. The solutions obtained by these methods are associated with those of the RKM [45] and HPM [8] in Figure 8. These assessments visibly confirm the effectiveness and efficiency of the MLS based PSO method to solve the differential equation governed to the considered nonlinear problem.

### 7.5 Example 5

In this case, in order to prepare another fair comparison, the parameters in the Volterra equations are considered similar to Ref. [8] as \( \alpha = \beta = \vartheta = 1 \) and \( \gamma = 0.1 \) with the initial conditions as \( \mu_0 = 14 \) and \( \delta_0 = 18 \). As it was mentioned earlier, in Ref. [8], these equations have been analysed by the HPM having a global domain as a line geometry from \( τ_{min} = 0 \) to \( τ_{max} = 0.2 \). The trajectories of the MLS based PSO, MLS based
GA, FDM based PSO and FDM based GA techniques are plotted in Figure 9. The populations of the preys and predators are compared with those of the RKM [45] and HPM [8] in Figure 10. These comparisons demonstrate that the MLS based PSO method could find the nearest solutions to the RKM for the prey population while the FDM based PSO approach has found the closest solutions to the RKM for the predator population.

7.6 Example 6

In this instance, according to Ref. [8], the constants, initial conditions and problem domain are fixed at $\alpha = \beta = \gamma = \theta = 1$, $\mu_0 = 3$, $\delta_0 = 2$, $\tau_{\min} = 0$, $\tau_{\max} = 1$. The convergences of the GA and PSO based methodologies are signified in Figure 11. The numerical findings achieved by these methods are contrasted together and with those of the RKM [45] and HPM [8]
FIGURE 11  Convergence trajectories of the MLS based PSO, FDM based PSO, MLS based GA and FDM based GA methods for Example 6

FIGURE 12  Prey ($\mu(\tau)$) and predator ($\delta(\tau)$) populations obtained by the RKM [45], HPM [8], MLS based PSO, FDM based PSO, MLS based GA and FDM based GA for Example 6

in Figure 12. These comparisons obviously verify the success of the MLS based PSO method to solve the considered Volterra differential equation.

8  CONCLUSIONS

In this paper, new combinations of the evolutionary algorithms with the approximation methods were described to analyse the nonlinear Volterra differential equation based predator-prey dynamics. The mathematical modelling of the objective functions were displayed by applying both the FDM and MLS approximation to interpolate the derivatives and the penalty method to impose the initial conditions. Simulation results were illuminated for several cases with different values for the constants and initial conditions of the problem to challenge the introduced strategies. The convergence trajectories of the MLS based PSO, MLS based GA, FDM based PSO and FDM based GA methods were plotted for each case. The diagrams of the prey and predator populations evidently expressed that the MLS
based PSO method is competent for solving this class of nonlinear differential equations with an acceptable convergent speed and high accuracy without any restrictive assumptions.

CONFLICT OF INTEREST
The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT
The data used to support the findings of this study are available from the corresponding author upon request.

ORCID
Mohammad Javad Mahmoodabadi https://orcid.org/0000-0002-4249-8623

REFERENCES
1. Abdel-Salam, E.A.B., Yousif, E.A., El-Aasser, M.A.: Analytical solution of the space-time fractional nonlinear Schrödinger equation. Rep. Math. Phys. 77(1), 19–34 (2016)
2. Tala-Tebue, E., et al.: Solitons and other solutions of the nonlinear fractional Zoomeron equation. Chin. J. Phys. 56(3), 1232–1246 (2018)
3. Alhakim, L.A., Moussa, A.A.: The double auxiliary equations method and its application to space-time fractional nonlinear equations. J. Ocean. Eng. Sci. 4(1), 7–13 (2019)
4. Cullen, A.C., Clarke, S.R.: A fast, spectrally accurate homotopy based numerical method for solving nonlinear differential equations. J. Comput. Phys. 385, 106–118 (2019)
5. Mahmoodabadi, M.J.: Epidemic model analyzed via particle swarm optimization based homotopy perturbation methods. Inf. Med. Unlocked 18, 100293 (2020)
6. Biazar, J., Ilie, M., Khoshkenar, A.: A new approach to the solution of the prey and predator problem and comparison of the results with the Adomian method. Appl. Math. Comput. 171(1), 486–491 (2005)
7. Biazar, J., Montazeri, A.: A computational method for solution of the prey and predator problem. Appl. Math. Comput. 163(2), 841–847 (2005)
8. Rafie, M., et al.: Solution of the prey and predator problem by homotopy perturbation method. Appl. Math. Comput. 188, 1419–1425 (2007)
9. Chowdhury, M.S.H., Hashim, I., Mawa, S.: Solution of prey–predator problem by numeric–analytic technique. Commun. Nonlinear Sci. Numer. Simul. 14(4), 1008–1012 (2009)
10. Yang, W.: Analysis on existence of bifurcation solutions for a predator–prey model with herd behavior. Appl. Math. Modell. 53, 433–446 (2018)
11. Yang, R., Ma, J.: Analysis of a diffusive predator-prey system with anti-predator behaviour and maturation delay. Chaos, Solitons Fractals 109, 128–139 (2018)
12. Yuan, H., Wu, J.: Positive solutions of a predator-prey model with cross-diffusion. Appl. Math. Comput. 331, 232–250 (2018)
13. Wang, Y.X., Li, W.T.: Uniqueness and global stability of positive stationary solution for a predator–prey system. J. Math. Anal. Appl. 462(1), 577–589 (2018)
14. Wang, Q.: Qualitative analysis of a Lotka–Volterra predator–prey system with migration. J. Math. Anal. Appl. 472(1), 421–431 (2019)
15. Wang, J., Wang, M.: Global solution of a diffusive predator–prey model with prey-taxis. Comput. Math. Appl. 77(10), 2676–2694 (2019)
16. Negreanu, M., Tello, J.I.: Global existence and asymptotic behavior of solutions to a Predator–Prey chemotaxis system with two chemicals. J. Math. Anal. Appl. 474(2), 1116–1131 (2019)
17. Mahdi, H., Hojjati, G., Abdi, A.: On the numerical stability of the general linear methods for Volterra integro-differential equations. Appl. Numer. Math. 142, 139–150 (2019)
18. Zhang, G., He, G., Dai, X.: Stability analysis of discontinuous Galerkin method for stiff Volterra functional differential equations. J. Math. Anal. Appl. 475(2), 1293–1303 (2019)
19. Gu, Z.: Spectral collocation method for weakly singular Volterra integro-differential equations. Appl. Numer. Math. 143, 263–275 (2019)
20. Haji, M.E.: Boundedness and asymptotic stability of nonlinear Volterra integro-differential equations using Lyapunov functional. J. King Saud Univ. Sci. 31(4), 1516–1521 (2019)
21. Saray, B.N.: An efficient algorithm for solving Volterra integro-differential equations based on Alpert’s multi-wavelets Galerkin method. J. Comput. Appl. Math. 348, 453–465 (2019)
22. Cao, F., Li, M.: Spherical data fitting by multiscale moving least squares. Appl. Math. Modell. 39, 3448–3458 (2015)
23. Spandan, V., et al.: A fast moving least squares approximation with adaptive Lagrangian mesh refinement for large scale immersed boundary simulations. J. Comput. Phys. 375, 228–239 (2018)
24. Dehghan, M., Abbaszadeh, M.: Interpolating stabilized moving least squares (MLS) approximation for 2D elliptic interface problems. Comput. Methods Appl. Mech. Eng. 328, 775–803 (2018)
25. Bang, S., et al.: Estimation of electron-photon coupling via moving least squares averaging: A method for fast-screening potential thermoelectric materials. Mater. Today Physics 6, 22–30 (2018)
26. Li, W., Zheng, H., Sun, G.: The moving least squares based numerical manifold method for vibration and impact analysis of cracked bodies. Eng. Fract. Mech. 190, 410–434 (2018)
27. Rodmiane, M.B., Temimi, H., Baccouch, M.: An iterative finite difference method for approximating the two-branched solution of Bratu’s problem. Appl. Numer. Math. 139, 62–76 (2019)
28. Lei, J., et al.: The generalized finite difference method for in-plane crack problems. Eng. Anal. Boundary Elem. 98, 147–156 (2019)
29. Sosa Jones, G., Arteaga, J., Jiménez, O.: A study of mimetic and finite difference methods for the static diffusion equation. Comput. Math. Appl. 76, 633–648 (2018)
30. Patil, H.M., Maniery, R.: Finite difference method based analysis of bioheat transfer in human breast cyst. Therm. Sci. Eng. Prog. 10, 42–47 (2019)
31. Fan, C.M., et al.: Numerical solutions of waves-current interactions by generalized finite difference method. Eng. Anal. Boundary Elem. 100, 150–163 (2019)
32. Ureña, F., et al.: Solving second order non-linear parabolic PDEs using generalized finite difference method (GFDM). J. Comput. Appl. Math. 354, 221–241 (2019)
33. Hu, W., et al.: The generalized finite difference method for an inverse boundary value problem in three-dimensional thermo-elasticity. Adv. Eng. Software 131, 1–11 (2019)
34. Gu, Y., et al.: The generalized finite difference method for long-time transient heat conduction in 3D anisotropic composite materials. Appl. Math. Modell. 71, 316–330 (2019)
35. Roul, P., Goura, V.M.K.P., Agarwal, R.: A compact finite difference scheme for a general class of nonlinear singular boundary value problems with Neumann and Robin boundary conditions. Appl. Math. Comput. 350, 283–304 (2019)
36. Bouamama, S., Blam, C., Fages, J.-G.: An algorithm based on ant colony optimization for the minimum connected dominating set problem. Appl. Soft Comput. 80, 672–686 (2019)
37. Zhang, M., et al.: Modeling and simulation of improved artificial bee colony algorithm with data-driven optimization. Simul. Model. Pract. Theory 93, 305–321 (2019)
38. Talbi, N.: Design of Fuzzy Controller rule base using Bat Algorithm. Energy Procedia 162, 241–250 (2019)
39. Whitley, D., Vose, M.D.: Foundations of Genetic Algorithms, Elsevier, Amsterdam (1995)
40. Lindfield, G., Penn, J.: Introduction to Nature-Inspired Optimization, Elsevier, Amsterdam (2017)
41. Simmons, G.F.: Differential Equations with Applications and Historical Notes, McGraw-Hill, New York (1972)
42. Mahmoodabadi, M.J., Mahmoodabadi, F., Arashafroz, M.: Development of the meshless local Petrov-Galerkin method to analyze three-dimensional transient incompressible laminar fluid flow. J. Serbian Soc. Comput. Mech. 12(2), 128–152 (2018)
43. Mahmoodabadi, M.J., Salahshoor Mottaghi, Z., Bagheri, A.: HEPSO: High dimensional zoomeron equation. Chin. J. Phys. 56(3), 1232–1246 (2018)
44. Biazar, J., Ilie, M., Khoshkenar, A.: A new approach to the solution of the prey and predator problem and comparison of the results with the Adomian method. Appl. Math. Comput. 171(1), 486–491 (2005)
45. Biazar, J., Montazeri, A.: A computational method for solution of the prey and predator problem. Appl. Math. Comput. 163(2), 841–847 (2005)
46. Rafie, M., et al.: Solution of the prey and predator problem by homotopy perturbation method. Appl. Math. Comput. 188, 1419–1425 (2007)
47. Chowdhury, M.S.H., Hashim, I., Mawa, S.: Solution of prey–predator problem by numeric–analytic technique. Commun. Nonlinear Sci. Numer. Simul. 14(4), 1008–1012 (2009)
45. Devries, P.L., Hasbun, J.E.: A First Course in Computational Physics, 2nd ed., Jones and Bartlett Publishers, Burlington, Massachusetts (2011)
46. Wang, C.-F., Song, W.-X.: A novel firefly algorithm based on gender difference and its convergence. Appl. Soft Comput. 80, 107–124 (2019)
47. Luo, K.: Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. Appl. Soft Comput. 77, 225–235 (2019)
48. Misaghi, M., Yaghoobi, M.: Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller. J. Comput. Des. Eng. 6(3), 283–295 (2019)
49. Dash, R., Dash, R., Rautray, R.: An evolutionary framework based microarray gene selection and classification approach using binary shuffled frog leaping algorithm. J. King Saud Univ. Comput. Inf. Sci. (2019). https://doi.org/10.1016/j.jksuci.2019.04.002
50. Elsayed, W.T., et al.: Modified social spider algorithm for solving the economic dispatch problem. Engineering Science and Technology. Eng. Sci. Tech. Int. J. 19(4), 1672–1681 (2016)
51. Bozorgi, S.M., Yazdani, S.: IWOA: An Improved whale optimization algorithm for optimization problems. J. Comput. Des. Eng. 6(3), 243–259 (2019)
52. Mahmoodabadi, M.J., Rashekh, M., Zoharib, T.: TGA: Team game algorithm. Future Comput. Inf. J. 3(2), 191–199 (2018)

How to cite this article: Taherkhorsandi M, Mahmoodabadi MJ. Numerical analysis of the Volterra differential equations via a combination of evolutionary algorithms with moving least squares and finite difference methods. J Eng. 2021;2021:177–187. https://doi.org/10.1049/tje2.12024