Journal of Engineering and Applied Sciences 8 (4): 127-131, 2013
ISSN: 1816-949X

A Hybrid Method for Solving Fuzzy Differential Equations

[1]W.S. Wan Daud, [1]M.Z. Ahmad, [1]E. Sakib and [2]M.K. Hasan
[1]Institute of Engineering Mathematics, Universiti Malaysia Perlis,
Kampus Tetap Pauh Putra, 02600 Arau, Perlis, Malaysia
[2]Faculty of Information Science and Technology, School of Information Technology,
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Abstract: This study proposes a hybrid method called Particle Swarm Optimization-Fuzzy Euler Method (PSOFEM) for solving fuzzy differential equations. The PSOFEM is obtained by embedding a particle swarm optimization technique in a fuzzy Euler scheme. Results of applying the PSOFEM to several fuzzy differential equations are presented. Final results showed that the PSOFEM is capable of generating fuzzy solutions for a large class of problems.

Key words: Hybrid method, particle swarm optimisation, fuzzy euler scheme, fuzzy differential equations, Malaysia

INTRODUCTION

Differential equations are used to model real world problems that involve the change of some variables with respect to another. Most of these problems require the solution to an initial-value problem that is the solution to a differential equation that satisfies a given initial condition. The differential equation that frequently confronted is shown below:

$$x'(t) = f(t,x(t)), \quad x(t_0) = x_0 \tag{1}$$

In real-life situations, differential equations that model the problems are too complicated to be solved analytically. Generally, the structure of the equation is known which represented by a function of f but the model parameters and the initial value x_0 are not known exactly. It is due to the uncertainties of the model which can arise during experimentations, data collections and measurement processes and during selection of initial values. Hence, the fuzzy set theory is applied to solve the differential equation with fuzzy initial values.

There are several approaches proposed in approximating the solution of fuzzy differential equations. Euler's method is one of the methods that have been applied numerically in fuzzy version that has been introduced by Ma *et al.* (1999). The researchers transformed a fuzzy differential equation by two parametric ordinary differential equations and then solved it by using fuzzy Euler's Method. However, the researchers have omitted the dependency problem which arises due to multiple occurrences of the same fuzzy interval treated independently in fuzzy interval arithmetic. This problem leads to many errors in computation. Besides that the method also cannot guarantee the convexity of fuzzy solutions on the time domain. In the study published by Abbasbandy and Allah Viranloo (2004), the researchers have taken an initiative to develop a 4th order Runge-Kutta method for solving FDEs. However, their proposed method suffers the same problem as Ma *et al.* (1999). Seeing this recurrence problem, Ahmad and Hasan (2011) have improved the method by introducing a new fuzzy version of Euler's Methods which takes into account the dependency problem among fuzzy sets. The idea of introducing a new version of fuzzy Euler's Method has made, the classical Euler's Method effectively handled.

In 2010, Ahmad and Hassan have implemented an optimization technique in generating the accurate approximations of differential equations using fuzzy initial values. The study has performed the computation of the function in determining the maximum and minimum value of each α cut. For this study, researchers proposed a Hybrid Method called Particle Swarm Optimization-Fuzzy Euler Method (PSOFEM), to compute the solutions of differential equations using fuzzy initial values. This combined method will be a great tool in solving fuzzy differential equations and might be applicable to compute solutions of linear and non-linear fuzzy differential equations.

Corresponding Author: W.S. Wan Daud, Institute of Engineering Mathematics,
Universiti Malaysia Perlis, Kampus Tetap Pauh Putra, 02600 Arau, Perlis, Malaysia

PRELIMINARY NOTES

Fuzzy differential equation: In this study, researchers briefly elaborate interpretation of a fuzzy differential equation. First, researchers consider the following Ordinary Differential Equation (ODE):

$$\left\{\begin{array}{l} x'(t) = f(t, x(t)), \quad t \in [t_0, T] \\ x(t_0) = x_0 \end{array}\right\} \tag{2}$$

Where, f: $[t_0, T]\times\Re\rightarrow\Re$ is real-valued function defined on $[t_0, T]$ with $T > 0$ and $x_0 \in \Re$.

Suppose that the initial condition in Eq. 1 is uncertain and modelled by a fuzzy number then researchers have the following fuzzy differential equation (Seikkala, 1987):

$$\left\{\begin{array}{l} X'(t) = f(t, X(t)), \quad t \in [t_0, T] \\ X(t_0) = X_0 \end{array}\right\} \tag{3}$$

Where, f: $[t_0, T]\times F(\Re)\rightarrow F(\Re)$ is obtained from Zadeh (1965)'s extension principle and $X_0 \in F(\Re)$. The membership function of f (t, X (t)) is defined as:

$$f(t,X)(z) = \left\{\begin{array}{ll} \sup X(s), & \text{if } z \in \text{range}(f) \\ z = f(t,s) & \\ 0, & \text{if } z \notin \text{range}(f) \end{array}\right\}$$

It follows that:

$$\left[f(t,X)\right]^{\alpha} = \left[\begin{array}{l} \min\left\{f(t,u) \middle| u \in \left[x_1^{\alpha}, x_2^{\alpha}\right]\right\}, \\ \max\left\{f(t,u) \middle| u \in \left[x_1^{\alpha}, x_2^{\alpha}\right]\right\} \end{array}\right]$$

Let, X: $[t_0, T]\rightarrow F(\Re)$ be a fuzzy process which derivative defined as:

$$\left[X'(t)\right]^{\alpha} = [x_1^{\alpha}'(t), x_2^{\alpha}'(t)], \quad \alpha \in (0,1]$$

If the derivative of the fuzzy process exists and satisfies the following conditions:

$$x_1^{\alpha}'(t) = \min\left\{f(t,u) \middle| u \in \left[x_1^{\alpha}, x_2^{\alpha}\right]\right\}, \; x_1^{\alpha}(t_0) = x_{0,1}^{\alpha}$$

$$x_2^{\alpha}'(t) = \max\left\{f(t,u) \middle| u \in \left[x_1^{\alpha}, x_2^{\alpha}\right]\right\}, \; x_2^{\alpha}(t_0) = x_{0,2}^{\alpha}$$

Then, the fuzzy process is the solution of Eq. 2 on $t \in [t_0, T]$ with $T > 0$ and $\alpha \in [0, 1]$. If researchers, solve Eq. 2 analytically then researchers have to verify that the interval $[X(t)]^{\alpha} = [x_1^{\alpha}(t), x_2^{\alpha}(t)]$ satisfies the following theorem:

Theorem 1: If X: $[t_0, T]\rightarrow F(\Re)$ is a fuzzy solution then (Agarwal *et al*., 2005):

- $[X(t)]^{\alpha}$ is nonempty compact subset of $\Re$
- $[X(t)]^{\alpha}_2 \subseteq [X(t)]^{\alpha}_1$ for $0 \le \alpha_1 \le \alpha_2 \le 1$
- $[X(t)]^{\alpha} = \cap^{\infty}_{n=1} [X(t)]^{\alpha}_n$ for any nondecreasing sequence $\alpha n \rightarrow \alpha$ in $[0, 1]$

Particle Swarm Optimization (PSO): Basically, PSO is defined as a population based optimization method using a population of particles (swarms) to finds the optimal solution (Kennedy and Eberhart, 1995; Shi and Eberhart, 1998). Each swarm represents a candidate solution to the problem and it is defined with a position and a velocity. In any multi-dimensional search space, the position of the jth particle and its velocity at iteration k are denoted as $X_j(k)$ and $V_j(k)$, respectively. The best previous position searches by the jth particle is represented as ${}_kP_{best,j}$ where as the global best position found in the swarm during the search process is represented as ${}_kG_{best}$. Similar to other algorithms, PSO is initialized with a population of random solutions and then the velocity and position of each particle are updated according to Eq. 4 and 5:

$$V_j(k+1) = w.V_j(k) + c_1 r_1\left[{}_kP_{best,j} - X_j(k)\right] + c_2 r_2\left[{}_kG_{best} - X_j(k)\right] \tag{4}$$

$$X_j(k+1) = X_j(k) + V_j(k+1) \tag{5}$$

Where, w is the inertia weight factor given in Eq. 6, c_1 and c_2 are the acceleration constants, both of which are often set to be 2 whereas r_1 and r_2 are two random numbers within the range [0, 1]. The following equation is introduced in order to achieve a balance between global and local exploration:

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}}\right)iter \tag{6}$$

In the equation, w_{max} and w_{min} is used to represent the initial and final values of the inertia weight respectively, $iter_{max}$ is denote the maximum number of iterations used in PSO whereas iter is the current number of iteration. The values of w_{max} and w_{min} are commonly declared as 0.9 and 0.4, respectively (Gaing, 2004).

THE PARTICLE SWARM OPTIMISATION-FUZZY EULER METHOD (PSOFEM)

In this study, a hybrid method called Particle Swarm Optimization-Fuzzy Euler Method (PSOFEM) is proposed. It is obtained by embedding a particle swarm optimization technique into a fuzzy Euler scheme.

Let us recall the classical Euler Method for approximating the solution of an ordinary differential equation:

$$x_{i+1} = x_i + hf(t_i, x_i),\ i = 0,1,2,...,N-1 \quad (7)$$

First, researchers denote B (h, t_i, x_i) = x_i+hf (t_i, x_i). Researchers then have:

$$x_{i+1} = B(h, t_i, x_i),\ i = 0,1,2,...,N-1 \quad (8)$$

Where, B: [t_0, T]×$\Re$→$\Re$ is a real-valued function defined on [t_0, T] with T>0. Using Zadeh's extension principle to Eq. 8 with respect to x_i, researchers then have the following fuzzy Euler Method:

$$x_{i+1} = B(h, t_i, X_i) \quad (9)$$

Where, B: [t_0, T]×F ($\Re$)→F ($\Re$) is obtained by Zadeh's extension principle. The membership function of B (h, t_i, X_i) can be defined as follows:

$$B(h, t_i, X_i)(z_i) = \begin{cases} \sup_{z_i = B(h, t_i, s_i)} X_i(s_i), & \text{if } z_i \in \text{range}(B) \\ 0 & \text{if } z_i \notin \text{range}(B) \end{cases} \quad (10)$$

Equation 10 can be calculated as follows:

$$x^{\alpha}_{1,i+1} = \min\left\{B(h, t_i, u) \middle| u \in \left[x^{\alpha}_{1,i}, x^{\alpha}_{2,i}\right]\right\}$$
$$x^{\alpha}_{2,i+1} = \max\left\{B(h, t_i, u) \middle| u \in \left[x^{\alpha}_{1,i}, x^{\alpha}_{2,i}\right]\right\}$$

The purpose of this study is to find the best solutions of $x^{\alpha}_{1, i+1}$ and $x^{\alpha}_{2, i+1}$ at each α-cut using PSO. The proposed algorithms are as follows:

Step 1: Determine the number of particles (N) in the interval [$x^{\alpha}_{1, i}$, $x^{\alpha}_{2, i}$] for every $\alpha \in$ [0, 1].

Step 2: Generate the initial population of u in the range $x^{\alpha}_{1, i}$ and $x^{\alpha}_{2, i}$ randomly as:

$$u^{\alpha}_1(0), u^{\alpha}_2(0), ..., u^{\alpha}_N(0)$$

Step 3: Calculate the objective function corresponding to the number of particles as follows:

$$B\left[h, t_i, u^{\alpha}_1(0)\right], B\left[h, t_i, u^{\alpha}_2(0)\right], ...,$$
$$B\left[h, t_i, u^{\alpha}_N(0)\right]$$

Step 4: Set the initial velocities of each particle to be zero which is:

$$v^{\alpha}_1(0) = v^{\alpha}_2(0) = ... = v^{\alpha}_N(0) = 0$$

Step 5: Set the iteration number as k = 1. Find the value of ${}_1p^{\alpha}_{best, 1}$, ${}_1p^{\alpha}_{best, 2..., 1}$, ${}_1p^{\alpha}_{best, N}$ and compare each other in order to obtain the value of ${}_1G^{\alpha}_{best}$ which is highest value of ${}_1p_{best, j}$.

Step 6: Modify the velocity, $v_1^{\alpha}(1)$, $v_1^{\alpha}(1)$,..., $v_N^{\alpha}(1)$ of each particle according to the formula as shown in Eq. 4.

Step 7: Based on the value of velocity, calculate the new position of each particle by using Eq. 5:

$$u^{\alpha}_1(1), u^{\alpha}_2(1), ..., u^{\alpha}_N(1)$$

Step 8: Update the values:

$$B\left[h, t_i, u^{\alpha}_1(1)\right], B\left[h, t_i, u^{\alpha}_2(1)\right], ..., B\left[h, t_i, u^{\alpha}_N(1)\right]$$

And check the convergence of the current solution. If the positions of all particles converge to the same set of values, it is assumed that the maximum value of $x^{\alpha}_{2, i+1}$ is found. Otherwise, step 5 is repeated by updating the iteration number as k = k+1. The iterative process is continued until all particles converge to the same optimum solution. The minimum value of $x^{\alpha}_{1, i+1}$ can be found by nothing that max B (h, t_i, u) = −min B (h, t_i, u).

Step 9: Using linear interpolation to interpolate ($x^{\alpha}_{1, i+1}$, α) and ($x^{\alpha}_{2, I+1}$, α) in order to obtain X_{i+1}.

Step 10: Repeat these processes for each t_i for I = 0, 1, 2,..., N−1. In order to show the capability of the proposed method, researchers provide two numerical examples involving non-linear problems.

NUMERICAL EXAMPLES

Example 1: Non-linear fuzzy differential equation:

$$x'(t) = \cos(tx), t \in [0,10]$$
$$x(0) = \left(0, \frac{\pi}{2}, \pi\right) \quad (11)$$

In order to obtain the approximate solution of Eq. 11, researchers divide the interval [0, 10] into 1000 uniformly spaced subintervals. The number of particles is set to 4. The obtained result is illustrated in Fig. 1.

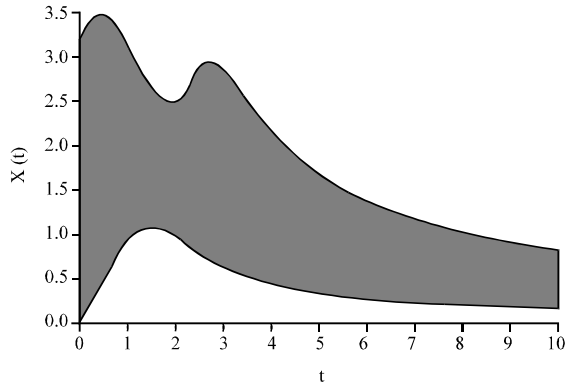

Fig. 1: The approximation solution for non-linear differential equation with fuzzy initial values

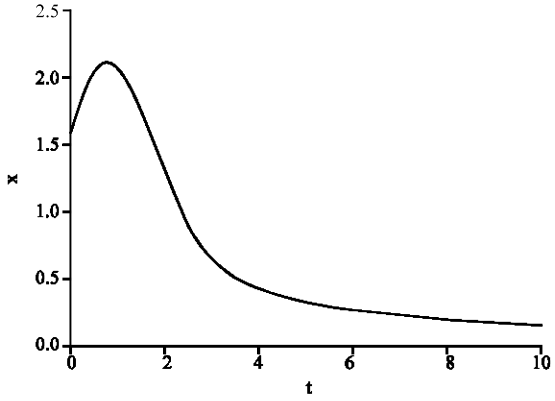

Fig. 2: The approximation solution for non-linear differential equation with crisp initial values

Based on the graph, researchers can see that the solution is non-monotone on the time domain. To verify the result, researchers provide the approximation solution for the same non-linear differential equation but with crisp initial condition, $x(0) = (\pi/2)$. The result is shown in Fig. 2. Based on Fig. 2, obviously shows that the solution is inside the fuzzy solution. Thus, researchers can say that the proposed algorithm is authentic and applicable to solve the non-linear problems.

Example 2: A fuzzy predator-prey model. In this example, researchers briefly explain the classical predator-prey model and introduce fuzzy predator-prey model.

Basically, the predator-prey model are used to describe the relationship between two different species of animals interact within the same environment or ecosystem which one of them depends on the other for food and the other one for survival. In real life problems, the model of predator-prey is always considered as an uncertainty, especially when determining the initial populations of the model. A system of fuzzy predator-prey is model can be modeled as follows:

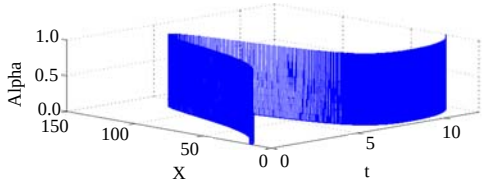

Fig. 3: The evolution of the prey population

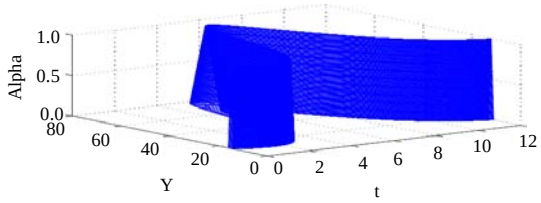

Fig. 4: The evolution of the predator population

$$X'(t) = aX - bXY,\ X(t_0) = X_0$$
$$Y'(t) = dY - cXY,\ Y(t_0) = Y_0 \quad (12)$$

Where:
X_0 and Y_0 = Fuzzy initial populations at t_0
a-d = Crisp numbers

The objective here is to show that the proposed method is, also applicable to solve a system non-linear differential equation with fuzzy initial values. Consider the following example of fuzzy predator-prey population model (Ahmad and Hasan, 2011):

$$X'(t) = X - 0.03XY,\quad X(t_0) = (14,15,16)$$
$$Y'(t) = -0.4Y + 0.01XY,\quad Y(t_0) = (14,15,16) \quad (13)$$

With N = 1000, t = 12 and h = 0.012, the solutions are plotted in Fig. 3 and 4.

The solutions from both graphs are also showing non-monotonic behavior on the time domain. The results are verified by comparing with the solutions of non-fuzzy with the initial populations are 15. The graphs are depicted in Fig. 5 and 6.

It is obviously shown in the graphs that the crisp solutions for non-fuzzy model are, also part of solutions for fuzzy model. Therefore, the numerical procedure proposed in this study is applicable to solve non-linear problems, since it produces the same solution behavior as in the crisp case.

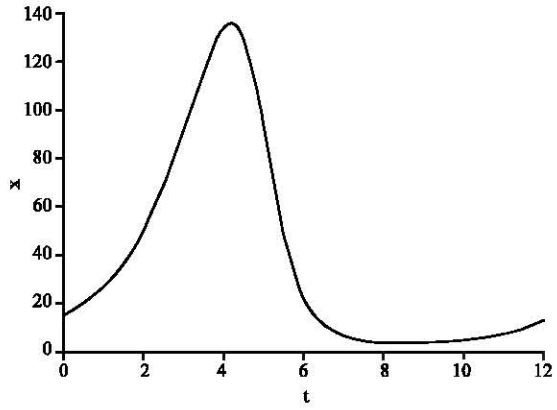

Fig. 5: The evolution of the pery population with crisp initial value

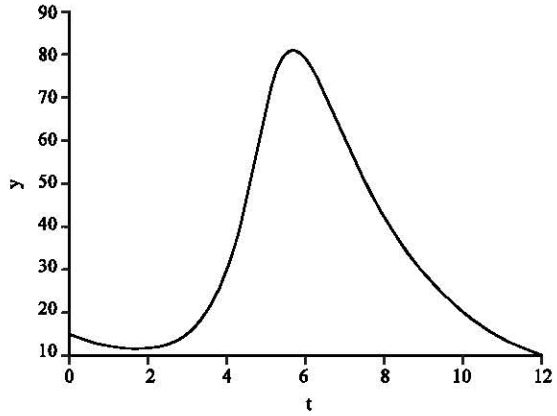

Fig. 6: The evolution of the predator population with crisp initial value

CONCLUSION

In this research, researchers have developed a new fuzzy hybrid version of Particle Swarm Optimization Fuzzy Euler's Methods for solving fuzzy differential equations. Example 1 has showed the effectiveness of the proposed algorithm in solving the non-linear fuzzy differential equation and at the same time able to produce an accurate membership function. The predator-prey model was applied as addition to show that the proposed method is also applicable for solving non-linear fuzzy system. For future research, the PSO may be combined with another numerical method such as Taylor Method in order to solve fuzzy differential equation.

ACKNOWLEDGEMENT

This research was supported by the Short Term Grant of Universiti Malaysia Perlis (UniMAP) under the project code 9001-00320.

REFERENCES

Abbasbandy, S. and T. Allah Viranloo, 2004. Numerical solution of fuzzy differential equation by Runge-Kutta method. Nonlinear Stud., 11: 117-129.

Agarwal, R.P., D. O'Regan and V. Lakshmikantham, 2005. Viability theory and fuzzy differential equations. Fuzzy Sets Syst., 151: 563-580.

Ahmad, M.Z. and M.K. Hasan, 2011. Modeling of biological populations using fuzzy differential equations. Int. J. Mod. Phys.: Conf. Ser., 9: 354-363.

Gaing, Z.L., 2004. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans. Energy Conver., 19: 384-391.

Kennedy, J. and R.C. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks, 4: 1942-1948.

Ma, M., M. Friedman and A. Kandel, 1999. Numerical solutions of fuzzy differential equations. Fuzzy Sets Syst., 105: 133-138.

Seikkala, S., 1987. On the fuzzy initial value problem. Fuzzy Sets Syst., 24: 319-330.

Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceedings of the 1998 World Congress on Computational Intelligence and IEEE International Conference on Evolutionary Computation, May 4-9, 1998, Piscataway, New Jersey, pp: 69-73.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353.