# On the application of Particle Swarm Optimization to Diffferential Equations

N. Khelil[1], L. Djerou[2], N. Rahmani[3] and H. Dakhia[4]

[1]LMA Laboratory, UMKBiskra, Algeria
n.khelil@univ-biskra.dz
[2] LESIA Laboratory, UMKBiskra, Algeria
ldjerou@yahoo.fr
[3] LMA Laboratory, UMKBiskra, Algeria
r_nacer@yahoo.com
[4]LMA Laboratory, UMKBiskra, Algeria
dakhia_ha@yahoo.fr

## 1 Introduction

Many problems in applied mathematics lead to ordinary differential equation. This work presents a novel numerical Differential Equation method based en Particle Swarm Optimization (PSO). Let $f = f(x, y)$ be a real-valued function of two real variables defined for $a \leq x \leq b$ , where $a$ and $b$ are finite, and for all real values of $y$ . The equations

$$y'(x) = f(x, y) \quad with \quad y(a) = y_0 \tag{1}$$

is called an initial-value problem (IVP); it symbolizes the following problem: To find a function $y(x)$, continuous and differentiable for $x \in [a, b]$ such that $y' = f(x, y)$ from $y(a) = y_0$ for all $x \in [a, b]$ [1]. One does not construct a closed-form expression for the desired solution $y(x)$ this is not even possible, in general, but in correspondence to certain discrete abscissa $x_k, k = 0, ..., n$, one determines approximate values $y_k$ for the exact value $y(x_k)$.

## 2 Problem Formulation

The main idea behind the algorithm is to use the following approximate formula for the derivative: $y'(x_i) \approx y'(x_{i-1} + \frac{h}{2})$, and from equation (1) we obtain, $\frac{y(x_i)-y(x_{i-1})}{h} \approx f(x_{i-1}, y(x_{i-1} + \frac{h}{2}))$, or $y(x_{i-1} + \frac{h}{2}) \approx y(x_{i-1}) + \frac{h}{2} f(x_{i-1}, y_{i-1})$, Thus, $\frac{y_i-y_{i-1}}{h} \approx f\Bigg(x_{i-1}, y_{i-1} + \frac{h}{2} f(x_{i-1}, y_{i-1})\Bigg)$.
Consequently, we have to consider the error formula: $\left[\frac{y_i-y_{i-1}}{h} - f\Bigg(x_{i-1}, y_{i-1} + \frac{h}{2} f(x_{i-1}, y_{i-1})\Bigg)\right]^2$
The performance function, associated to an individual $y = (y_1, y_2, ..., y_n)$ will be:

$$E(y) = \sum_{i=1}^{n} \left[\frac{y_i - y_{i-1}}{h} - f\Bigg(x_{i-1}, y_{i-1} + \frac{h}{2} f(x_{i-1}, y_{i-1})\Bigg)\right]^2 \tag{2}$$

The proposed method consists to calculate the minimum of equation (2) with Particle Swarm Optimization, and then permits to avoid accumulated errors.

The particle swarm treatment supposes a population of individuals designed as real valued vectorsparticles, and some iterative sequences of their domain of adaptation must be established. It is assumed that these individuals have a social behavior, which implies that the ability of social conditions, for instance, the interaction with the neighborhood, is an important process in successfully finding good solutions to agiven problem.
The strategy of the PSO algorithm is summarized as follows: We assume that each agent (particle) i can be represented in a N dimension space by its current position $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ and its corresponding velocity $v_i = (v_{i1}, v_{i2}, \cdots, v_{iN})$. Also a memory of its personal (previous) best position is represented by $p_i = (p_{i1}, p_{i2}, \cdots, p_{iN})$, called (pbest), the subscript i range from 1 to s, where s indicates the size of the swarm. Commonly, each particle localizes its best value so far

(pbest) and its position and consequently identifies its best value in the group (swarm), called also (sbest) among the set of values (pbest). The velocity and position are updated as:

$$v_{ij}^{k+1} = w_{ij}v_{ij}^{k} + c_1 r_1^k[(pbest)_{ij}^k - x_{ij}^k] + c_2 r_2^k[(sbest)_{ij}^k - x_{ij}^k] \ . \tag{3}$$

$$x_{ij}^{k+1} = v_{ij}^{k+1} + x_{ij}^{k} \ . \tag{4}$$

where $x_i^{k+1}, v_i^{k+1}$ are the position and the velocity vector of particle $i$ respectively at iteration $k+1$, $c_1$ and $c_2$ are acceleration coefficients for each term exclusively situated in the range of $2--4$, $w_j$ is the inertia weight with its value that ranges from 0.9 to 1.2, where as $r_1, r_2$ are uniform random numbers between zero and one. For more details, the double subscript in the relations ( 3) and ( 4) means that the first subscript is for the particle $i$ and the second one is for the dimension $j$.

We used the above algorithm for the equation $y' = 1 - y(x)$ from $y(0) = 0$, in order to find an approximate solution $y : [0 \quad 4] \mapsto R$.

The figure 1 illustrate the PSO solution obtained with $h = 1.0$ is much more accurate than the Euler solution with the same $h$. For $h$ smaler, we obtain impressive results.

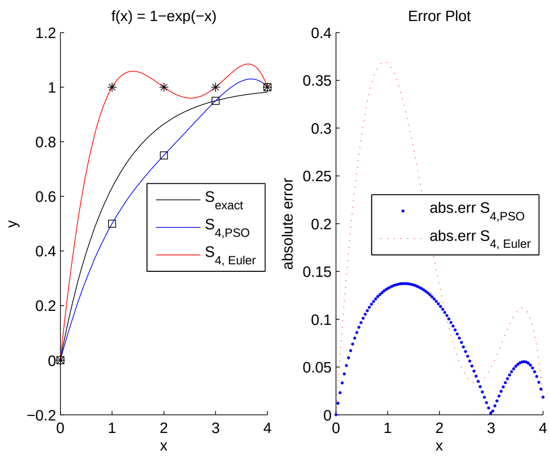


**Fig. 1.** Analytical/Numerical solutions by Euler and PSO method, with comparison

## References


1. Henrici, P.: Elements of Numerical Analysis. McGraw-Hill, New York (1964)
2. Parsopoulos K. E. and M. N. Vrahatis : Modification of the Particle Swarm Optimizer for Locating all the Global Minima.
3. J.Kennedy and R.C.Eberhart, Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, (2001).
4. Y. H.Shi and R.C.Eberhart, Fuzzy adaptive particle swarm optimization.IEEE Int. Conf. on Evolutionary Computation, pp. 101-106, (2001).
5. Y. H.Shi and R.C.Eberhart, A modified particle swarm optimizer, Proc. of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9 (1998).
6. Y. H.Shi and R.C.Eberhart, Parameter selection in particle swarm optimization. Evolutionary Programming VII, Lecture Notes in Computer Science, pp. 591-600, (1998).
7. Zerarka A. and N. Khelil, A generalized integral quadratic method: improvement of the solution for one dimensional Volterra integral equation using particle swarm optimization. Int. J. Simulation and Process Modelling 2(1-2), 152-163, (2006).
8. Djerou L.et al. Numerical Integration Method Based on Particle Swarm Optimization.Y. Tan et al. (Eds.) : Part I, LNCS 6728, Springer-Verlag Berlin Heidelberg, pp. 221-226, (2011).