The particle swarm optimization against the Runge's phenomenon: Application to the generalized integral quadrature method

A. Zerarka, A. Soukeur and N. Khelil

Abstract—In the present work, we introduce the particle swarm optimization called (PSO in short) to avoid the Runge's phenomenon occurring in many numerical problems. This new approach is tested with some numerical examples including the generalized integral quadrature method in order to solve the Volterra's integral equations.

Keywords—Integral equation, particle swarm optimization, Runge's phenomenon.

I. Introduction

In recent years, much attention has been devoted to the investigation of new mathematical models and numerical approaches to evaluate the solutions of the EDP and the integral equations. Excellent surveys which contain both numerical and theoretical researches are given in [1-20].

The primary argument for the interest of the type of this problem comes naturally from its wide applications almost in any branches of science and engineering described by systems of ODEs and PDEs [21-31] which in some situations the solutions present the Runge's phenomenon in the edges of the interval. This situation can be avoided by a specific utilization of the algorithm PSO. The PSO algorithm is a parallel evolutionary computation technique proposed by Kennedy and Eberhart in 1995. The PSO has nowadays gained great importance in computer optimization.

The latest numerical approach to date is the generalized integral quadrature method introduced by Zerarka and Soukeur [32]. It was first applied to one-dimensional Volterra integral in the linear and nonlinear cases, where the solution is not completely reproduced in the domain in which strong oscillations can arise. This method studies the situation in which the unknown function is identified as the Lagrange polynomial [33] and the interpolating points of the Tchebychev type are used.

New calculations are performed for the construction of the solution by a suitable choice of the interpolating points using the particle swarm optimization (PSO) in order to avoid the Runge's phenomenon [34]. Our main purpose is to show how the Runge's phenomenon can be completely removed from the solution of interest. We examine two specific examples in which the Runge's phenomenon emerges in the evaluation of solutions.

The contents of this paper are organized as follows. In Section 2, a formulation adapted to the strategy of particle swarm optimization and the construction of an algorithm to generate the different agents in a swarm. The Section 3 gives the Runge's phenomenon for polynomial interpolation. Section 4 exposes some essential examples to show how the PSO algorithm can lead to a satisfactory result for the construction of solutions.

A. Zerarka is with the Labo of Physics and Applied Mathematics, University Med Khider, BP145, 07000 Biskra; Algeria, e-mail: abzerarka@yahoo.fr.(corresponding author)
A. Soukeur (e-mail: souka@yahoo.fr) and N. Khelil (e-mail: khelilnac@yahoo.fr) are with the Department of Mathematics, University Med Khider, BP145, 07000 Biskra; Algeria.

II. Overall description and strategy of particle swarm optimization

A new stochastic algorithm has recently appeared, called 'particle swarm optimization' PSO. The term 'particle' means any natural agent that describes the swarms behavior. The PSO model is a particle simulation concept, and was first proposed by Eberhart and Kennedy [34, 35]. Based upon a mathematical description of the social behaviors of swarms, it has been shown that this algorithm can be efficiently generated to find good solutions to a certain number of complicated situations such that for instance, the static optimization problems, the topological optimization, and others [36-40] and references contained therein. Since then, several variants of the PSO have been developed [41-48]. It has been shown that, the question of convergence of the PSO algorithm is implicitly guaranteed if the parameters are adequately selected [49, 50].

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 $X_i = (x_{i1},\, x_{i2},\, ...,\, x_{iN})$ and its corresponding velocity $V_i = (v_{i1},\, v_{i2},\, ...,\, v_{iN})$. Also a memory of its personal (previous) best position is represented by $P_i = (p_{i1},\, p_{i2},\, ...,\, p_{iN})$, 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

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

$$x_{ij}^{k+1} = v_{ij}^{k+1} + x_{ij}^{k} \tag{2}$$

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 to 4, w_j is the inertia weight with its value that ranges from 0.9 to 1.2, whereas r_1^k , r_2^k are uniform random numbers between zero and one. For more detail, the double subscript in the relations (1) and (2) means that, the first subscript for the particle i and the second one for the dimension j. The role of a suitable choice of the inertia weight w_j is important in the PSO success. In the general case, it can be initially set equal to its maximum value, and progressively we decrease it if the better solution is not reached. Too often, in the relation (1), v_{ij}^{k+1} is replaced by v_{ij}^{k+1}/σ, where σ denotes the constriction factor that controls the velocity of the particles. The following algorithm should give us the general idea how to generate the particles in the swarm:

Step 1: Set the values of the dimension space N, and the size s of the swarm (s can be taken randomly).

Step 2: Initialize the iteration number k (in the general case is set equal to zero).

Step 3: Evaluate for each agent, the velocity vector using its memory and equation (1), where pbests and sbest can be modified.

Step 4: Each agent must be updated by applying its velocity vector and its previous position using equation (2).

Step 5: Repeat the above steps (3, 4 and 5) until a convergence criterion is reached.

III. Illustration of the Runge phenomenon for polynomial interpolation

Wild oscillations can occur near the ends of the interval for large degree polynomials and can lead to the Runge's phenomenon (RP). So far the only remedy against the RP is the Chebyshev type distribution towards the end of the interval. The oscillations can be minimized by using Chebyshev nodes instead of equidistant nodes [32]. In this case the maximum error is guaranteed to diminish with increasing polynomial order. For the high degree polynomials it is suitable to use the B-spline functions which are defined in the subintervals. The PSO algorithm is more flexible and gives results with a very high accuracy, and resolves in a systematic way the oscillations phenomenon when the interpolant polynomial becomes a bad approximant as the degree increases and restores accuracy to the solutions of the problem under consideration. Thus, the suppression of Runge's phenomenon is now possible with the help of the PSO algorithm.

It is important to underline that, the main quantity to estimate the error in interpolating polynomial is expressed in terms of the $\theta(x) = \prod_{i=1}^{N}(x-x_i)$. Thus, the points of interpolation are chosen such that $\theta(x)$ differ the least possible from zero in the interval of interest.

IV. Examples

These examples can be viewed as typical cases which provides a good illustration of Runge's phenomenon. We note that, the accuracy of results depends manifestly to success of particles in the swarm to locate the best points to avoid the Runge's phenomenon. For easy interpretation, the numerical results evaluated by PSO algorithm, and those obtained by the exact formula are plotted in same graph. The new candidates for the interpolating points are displayed in Tables I and II in the cases $N = 11$ and $N = 21$ respectively. For convenience, we have presented the parameters settings to generate the PSO algorithm for both examples as Table II shows.

TABLE I
THE NEW CANDIDATES FOR THE INTERPOLATING POINTS x_{i1} AND x_{i2} GENERATED BY PSO ALGORITHM FOR THE EXAMPLES 1 AND 2 RESPECTIVELY. NUMBER OF POINTS $N = 11$.

i	x_{i1}	x_{i2}
1	-0.9909	10.0734
2	-0.1866	10.2218
3	-0.5079	10.2230
4	-0.7091	11.1317
5	-0.8578	11.2050
6	0.1704	11.4177
7	0.3139	11.4224
8	0.4184	11.8523
9	0.8566	11.8662
10	0.8661	11.9352
11	0.9100	11.9511

TABLE II
THE NEW CANDIDATES FOR THE INTERPOLATING POINTS x_{i1} AND x_{i2} GENERATED BY PSO ALGORITHM FOR THE EXAMPLES 1 AND 2 RESPECTIVELY. NUMBER OF POINTS $N = 21$

i	x_{i1}	x_{i2}	i	x_{i1}	x_{i2}
1	-0.0110	10.2281	13	0.0171	11.2639
2	-0.2000	10.2402	14	0.2172	11.6565
3	-0.3122	10.2877	15	0.2600	11.6864
4	-0.3423	10.3395	16	0.400	11.6939
5	-0.3832	10.3470	17	0.4916	11.7221
6	-0.4057	10.5138	18	0.5138	11.7522
7	-0.6617	10.5934	19	0.6836	11.8948
8	-0.6937	10.6100	20	0.8499	11.9345
9	-0.7023	10.7072	21	0.8819	11.9856
10	-0.9183	10.8737	22		
11	-0.9791	11.1006	23		
12	-1.0000	11.1257	24		

TABLE III
PARAMETERS SETTINGS TO GENERATE THE PSO ALGORITHM FOR BOTH EXAMPLES. CASE $N = 21$.

	Example 1	Example 2
Population Size	21	21
Number of Iterations	500	600
Acceleration Coefficients: c_1 and c_2	0.5	0.5
Inertial Weight	1.2 to 0.4	1.2 to 0.4
Desired Accuracy	10^{-5}	10^{-4}

A. Example 1

We now present an explicit example of calculating a specific function as

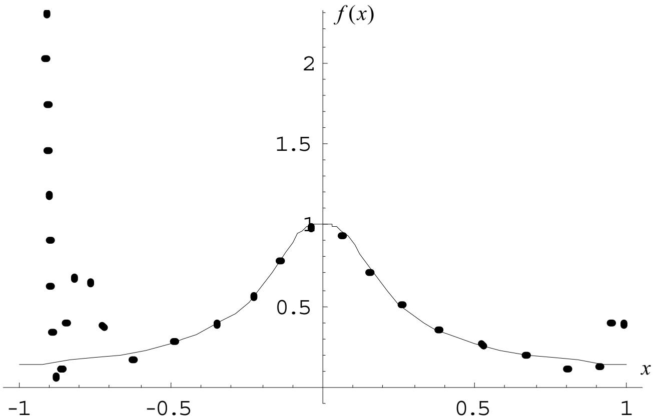

Fig. 1. Function objective (3) for the example 1. Solid line: exact solution, dots line: Lagrange interpolating polynomial of order $N = 11$.

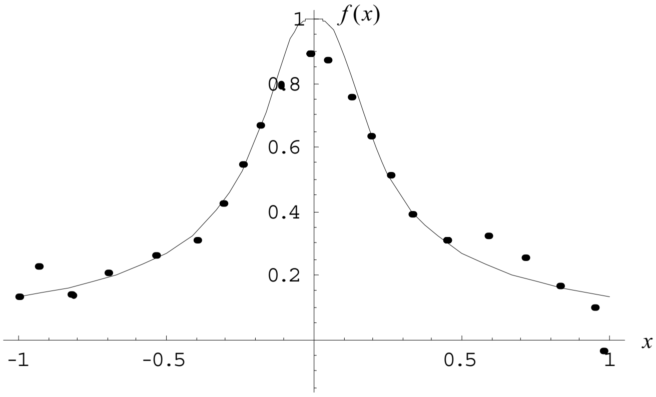

Fig. 2. Function objective (3) for the example 1. Solid line: exact solution, dots line: PSO algorithm with $N = 11$.

$$f(x) = \frac{1}{(1 + 400\,|x|^3)^{1/3}} \tag{3}$$

defined in the interval $[-1, 1]$

The Figure 1, depicts the graph of the exact function and the result obtained by Lagrange interpolating polynomial. We clearly state that the Runge's phenomenon is veritably present near the ends of the interval and must be removed by handling the PSO algorithm.

It is evident also that, the discrepancies resulting from Lagrange interpolating polynomial (Figure 1) are much apparent than their counterparts obtained by the PSO algorithm (Figure 2) with only $N = 11$. At present, the Runge's phenomenon becomes treatable and can be completely removed as expected in the Figure 3, where the number of points is taken to be $N = 21$ is sufficient to give an excellent objective function. As an important consequence of this feature is that the PSO algorithm still works even in the case where the function presents some singularities (see example 2).

B. Example 2

In the sequel, we proceed with a practical example more complicated. We first briefly introduce, the generalized integral quadrature method (GIQ), the details can be found in [32].

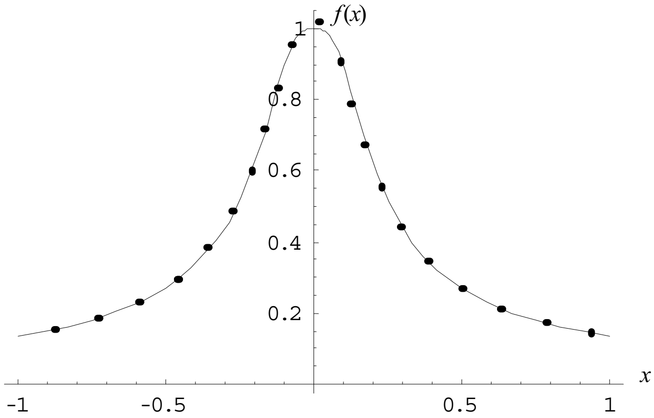

Fig. 3. Function objective (3) for the example 1. Solid line: exact solution, dots line: PSO algorithm with $N = 21$.

A brief description of generalized integral quadratic method is summarized as follows: the Volterra equation integral is written as

$$f(x) = \varphi(x) + \lambda \int_0^x K(x,s) f(s) ds, \qquad 0 \le x \le T \tag{4}$$

where λ is a parameter, $\varphi(x)$ is a given function and $K(x,s)$ is the kernel of the integral equation. It is assumed that the functions involved in (4) are sufficiently regular. In (4), the upper limit of the integral term is a variable.

If we set

$$U(x) \equiv \int_0^x K(x,s) f(s) ds \tag{5}$$

then $U(x)$ may be approximated by

$$U(x_m) = \sum_{j=0}^{N} C_{mj} K(x_m, x_j) f(x_j), \quad m = 0, ..., N, \tag{6}$$

$$U_k(x_m) = \sum_{j=0}^{N} C_{mj} K(x_m, x_j) P_{N,k}(x_j), \tag{7}$$

where $P_{N,j}(x)$ are Lagrange interpolated polynomials, and the interpolating points are taken as the points of Tchebychev of the form $x_j = \frac{1}{2} T \left[1 - \cos(\frac{2j+1}{2N+2}\pi)\right]$, $0 \le j \le N$, and

$$C_{mj} = \frac{1}{K(x_m, x_j)} \int_0^{x_m} K(x_m, s) P_{N,j}(s) ds, \tag{8}$$

In order to avoid unnecessary calculation, it is therefore more convenient to get the desired coefficients C_{ii} in the following form

$$C_{ii} = \int_0^{x_i} \bar{K}(x_i, s) ds - \sum_{j=0, j\neq i}^{N} C_{ij} \bar{K}(x_i, x_j), \quad \text{for } i = 0, ..., N, \tag{9}$$

where $\bar{K}(x_m, x) = \frac{K(x_m,x)}{K(x_m,x_m)}$.

Now the expressions (8) and (9) provide the formulae for the weighting coefficients, and the function $f(x)$ is expressed as.

$$f(x) = \sum_{j=0}^{N} f(x_j) P_{N,j}(x). \qquad (10)$$

Even with the Chebyshev-type points, in some situations, the improvements cease. To overcome this problem, it is always possible to introduce special nodes selected by PSO algorithm

The manifestation of the RP is evaluated by the Lebesgue constant Π_N in terms of N degree polynomials $P_{N,j}(x)$

$$\Pi_N = \max_{x\in[a,b]} \sum_{j=0}^{N} |P_{N,j}(x)| \qquad (11)$$

with uniform nodes $\Pi_N = O\left(\frac{2^N}{N \ln N}\right)$. We see from this that strong oscillations can emerge, whereas with the Chebyshev-type points $\Pi_N = O(\ln N)$ this situation can lead to a good improvement.

Now let us return to this model problem with a concrete exposition. As a second illustrative example, the linear integral equation taken from [51] is considered, i.e.,

$$f(x) = 1 - \int_0^x (x-s)^{-\frac{1}{2}} f(s)ds. \qquad (12)$$

The above equation has the exact solution $f(x) = \exp(\pi x)(1-\mathrm{erf}(\sqrt{\pi x}))$, and contains a weakly singular kernel $(x-s)^{-\frac{1}{2}}$. This singularity can be avoided by the following transformation: $u = \sqrt{x-s}$. As in [32], the standard numerical result on $[10, 12]$ seems to disagree with the analytic solution because in this region the oscillations are very pronounced, see Figure 4. The RP is appeared in this region because high order polynomial interpolation on equispaced grids is used. When the interpolating points using the PSO algorithm are introduced in the problem, the solution becomes more representative, and a minor difference is observed i.e., as expected on the Figure 5 with $N = 11$, and the error tolerated being 10^{-4} . The good result is then achieved by using an optimal set of interpolation points $N = 21$. The result is displayed in Figure 6, on which the solution is now almost identical with the exact one.

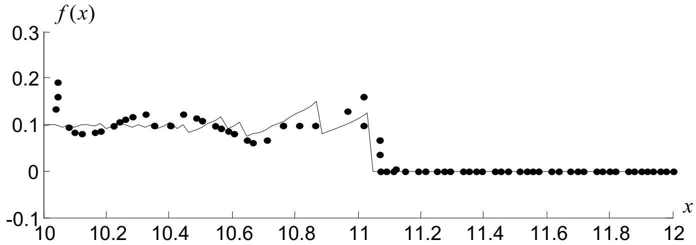

Fig. 4. Solutions of the equation (4) for the kernel: $K(x, y) = (x - y)^{-\frac{1}{2}}$ and $\varphi(x) = 1$, $10 \le x \le 12$. Solid line: exact solution, dots line: Lagrange interpolating polynomial of order $N = 11$.

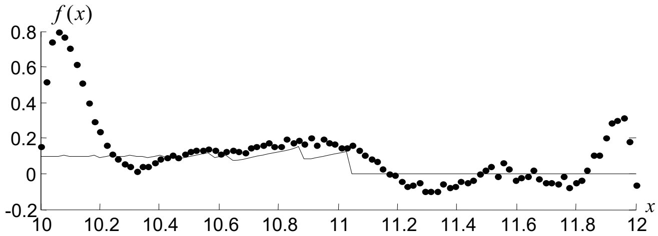

Fig. 5. Solutions of the equation (4) for the kernel: $K(x, y) = (x - y)^{-\frac{1}{2}}$ and $\varphi(x) = 1$, $10 \le x \le 12$. Solid line: exact solution, dots line: PSO algorithm with $N = 11$.

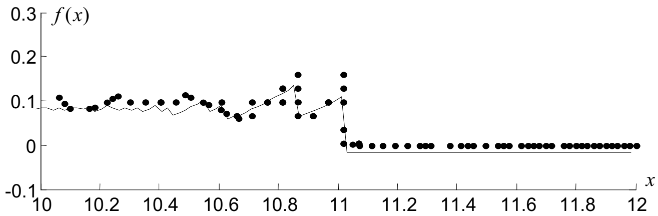

Fig. 6. Solutions of the equation (4) for the kernel: $K(x, y) = (x - y)^{-\frac{1}{2}}$ and $\varphi(x) = 1$, $10 \le x \le 12$. Solid line: exact solution, dots line: PSO algorithm with $N = 21$.

V. Comments and conclusions

We presented a formulation that uses the PSO algorithm in order to avoid the Runge's phenomenon which emerges for large degree polynomials. The preliminary results, obtained through the use of the PSO method, show that the Runge's phenomenons can be always removed from the problem under consideration and the comparison with the exact solutions is spectacular.

In this work the particle swarm optimization is introduced to improve the solutions of the Volterra integral equation. We have shown that the PSO procedure provides substantially better accuracy than the conventional Tchebychev's interpolating points which are always known to be the only best points which permits a good approach of the interpolating function. For instance, the Figures 3 and 6 show graphically the best solutions for both examples. It is shown that, in some problems, which contain more complexity, the PSO algorithm can also lead to results with a high effectiveness [48-50]. As seen from the numerical results, the best interpolating points are attained in a surprisingly short time with error tolerances of 10^{-5} and 10^{-4} for Examples 1 and 2 respectively.

The most important remark is that, the PSO algorithm is readily applicable to both conventional and complex applications and can provide good results even for a great number of the interpolating points.

Acknowledgment

The authors gratefully acknowledge helpful conversations with Prof. W. Cramer and Prof. V. G. Foester. One of the authors wishes to express its sincere thanks to Dr. Dyn Keit for bringing his attention to the PSO method.This work was sponsored in part by the M.E.R.S (Ministère de l'Enseignement et de la Recherche Scientifique): Under contract No. D0701/01/07.

References

[1] Brunner H. and van der Houwen P. J. The Numerical Solution of Volterra Equations, CWI Monographs, North-Holland, Amsterdam (1986).
[2] Baker C. T. H. The Numerical Treatment of Integral Equations, Oxford University Press (1977).
[3] Delves L. M. and Walsh J. Numerical Solution of Integral Equations, Clarendon Press, Oxford (1974).
[4] Cochran J. A. The Analysis of Linear Integral Equations, McGraw-Hill (1972).
[5] Atkinson K. E. A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind SIAM, Philadelphia (1976).
[6] Wolkenfelt P. H. M. The construction of reducible quadrature rules for Volterra integral and integro-differential equations, IMA J. Numer. Anal. 2 (1982) 131-152.
[7] Dobner H.-J., Bounds of high quality for first kind Volterra integral equations. Reliable Computing 2(1), (1996) 35-45,.
[8] Bugajewski D. On the Volterra integral equation in locally convex spaces, Demonstratio Math., 25, (1992) 747-754.
[9] Delves L.M. and Mohamed J.L. Computational methods for integral equations. Cambridge University Press, Cambridge (1985).
[10] H. Brunner, A. Pedas, and G. Vainikko. The piecewise polynomial collocation method for nonlinear weakly singular Volterra equations. Mathematics of Computation Volume 68 (1999) 1079 - 1095
[11] J. P. Kauthen, H. Brunner. Continuous collocation approximations to solutions of first kind Volterra equations. Mathematics of Computation Volume 66 (1997) 1441 - 1459
[12] Q. Hu . Multilevel correction for discrete collocation solutions of Volterra integral equations with delay arguments Applied Numerical Mathematics Volume 31 (1999) 159 - 171
[13] H. Brunner, Q. Hu and Q. Lin . Geometric meshes in collocation methods for Volterra integral equations with proportional delays. IMA Journal of Numerical Analysis Volume 21 (2001) 783-798.
[14] V. Karlin, V. G. Maz'ya, A. B. Movchan , J. R. Willis, and R. Bullough . Numerical solution of nonlinear hypersingular integral equations of the Peierls type in dislocation theory. SIAM Journal on Applied Mathematics, 60 (2000) 664–678.
[15] M. H. Fahmy, M. A. Abdou and M. A. Darwish. Integral equations and potential-theoretic type integrals of orthogonal polynomials. J. Comput. Appl. Math. 106 (2) (1999) 245-254.
[16] T. Diogo and P. M. Lima. Numerical solution of a non-uniquely solvable Volterra integral equation using extrapolation methods, Journal of Computational and Applied Mathematics, 140 (2002) 537-557.
[17] Fabio Fagnani and Luciano Pandolfi. A recursive algorithm for the approximate solution of Volterra integral equations of the first kind of convolution type. Inverse Problems 19 (2003) 23-47
[18] Peeter Oja, Darja Saveljeva. Cubic spline collocation for Volterra integral equations. Computing Volume 69 (2002) 319 - 337
[19] Igor Bock, Jan Lovisek. On a reliable solution of a Volterra integral equation in a Hilbert space . APPLICATIONS OF MATHEMATICS, Vol. 48, No. 6 (2003) 469-486,
[20] M. Federson, R. Bianconi and L. Barbanti, Linear Volterra integral equations, Acta Math Appl. Sinica (English Series), 18(4), (2002) 553-560.
[21] M. Federson, R. Bianconi and L. Barbanti. Linear Volterra integral equations as the limit of discrete systems. CADERNOS DE MATEMÁTICA 4, (2003) 331-352
[22] Linz, P., Analytical and Numerical Methods for Volterra Equations. SIAM, Philadelphia, (1985).
[23] Kulisch, U., Miranker, W.L., A new Approach to Scientific Computation. Academic Press, New York, (1983).
[24] Hammer, R., Hocks, M., Kulisch, U., Ratz, D., Numerical Toolbox for Verified Computing I. Springer Verlag, Berlin, (1995).
[25] Bugajewski D., On differential and integral equations in locally convex spaces, Demonstratio Math., 28, (1995) 961-966.
[26] Dobner, H.-J., Bounds for the Solution of Hyperbolic Problems. Computing 38, (1987) 209-218, .
[27] Copson, E.T., Partial Differential Equations. Cambridge University Press, Cambridge (1975).
[28] Bugajewska D., Topological properties of solution sets of some problems for differential equations, Ph. D.Thesis, Poznań, (1999).
[29] Constantin A., On the unicity of solution for the differential equation $x^{(n)} = f(t, x)$, Rend. Circ. Mat. Palermo, Serie II, 42, (1991) 59-64.
[30] Bugajewski D., Szu.a S., Kneser's theorem for weak solutions of the Darboux problem in Banach spaces, Nonlinear Analysis, 20, No 2, (1993) 169-173.
[31] Maron, M. and Lopez, R., Numerical Analysis. Wadsworth Publishing Company, Belmont California, (1991).
[32] Zerarka A. and Soukeur A. A generalized integral quadratic method: I. an efficient solution for one-dimensional Volterra integral. Communication in Nonlinear Science and Numerical Simulation, 10 (2005)653 –663.
[33] Zerarka A, Hassouni S, Saidi H and Boumedjane Y. Energy spectra of the Schrödinger equation and the differential quadrature method. Communication in Nonlinear Science and Numerical Simulation, 10 (2005)737 –745.
[34] J. Kennedy, R.C. Eberhart, Particle Swarm Optimization, Proc. IEEE Int. Conf. Neural Networks, Piscataway, NJ, USA, 1942-1948, (1995).
[35] Eberhart, R. C. and Kennedy, J. A New Optimizer Using Particles Swarm Theory. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, (1995), pp. 39-43.
[36] K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Objective function stretching to alleviate convergence to local minima, Nonlinear Analysis TMA 47, 3419-3424, (2001).
[37] K.E. Parsopoulos, V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Stretching technique for obtaining global minimizers through Particle Swarm Optimization, Proceedings of the PSO Workshop, Indianapolis, USA, 22-29, (2001).
[38] K.E. Parsopoulos, M.N. Vrahatis, Modification of the Particle Swarm Optimizer for locating all the global minima, Artificial Neural Networks and Genetic Algorithms, V. Kurkova et al. (Eds.), Springer, 324-327, (2001).
[39] Fourie, P. C. and Groenwold, A. A. Particle Swarms in Topology Optimization. Extended Abstracts of the Fourth World Congress of Structural and Multidisciplinary Optimization, Dalian, China, June 4-8, (2001), pp. 52-53.
[40] Fourie, P. C. and Groenwold, A. A. Particle Swarms in Size and Shape Optimization. Proceedings of the International Workshop on Multidisciplinary Design Optimization, Pretoria, South Africa, August 7-10 (2000), pp. 97-106.
[41] Eberhart, R.C., Simpson, P.K., Dobbins, R.W.: Computational Intelligence PC Tools. Academic Press Professional, Boston (1996).
[42] Kennedy, J. The Behavior of Particles. Evol. Progr. VII (1998) 581-587.
[43] Kennedy, J., Eberhart, R.C. Swarm Intelligence. Morgan Kaufmann (2001).
[44] Shi Y H, Eberhart R C. Fuzzy adaptive particle swarm optimization. IEEE Int. Conf. on Evolutionary Computation, (2001) 101-106.
[45] Kennedy, J. and Spears, W. M. Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator. Proceedings of the (1998) IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9 (1998).
[46] Shi, Y. and Eberhart, R. C. A Modified Particle Swarm Optimizer. Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9 (1998).
[47] Shi, Y. H. and Eberhart, R. C. Parameter Selection in Particle Swarm Optimization. Evolutionary Programming VII, Lecture Notes in Computer Science, (1998), pp. 591-600.
[48] Clerc, M. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. Proceedings of the 1999 IEEE Congress on Evolutionary Computation, Washington D.C., (1999), pp. 1951-1957.
[49] Eberhart, R. C., Shi, Y. Parameter Selection in Particle Swarm Optimization. Lecture Notes in Computer Science-Evolutionary Programming VII, Porto, V. W., Saravanan, N. Waagen, D., Eiben, A. E., 1447, 591-600, Springer, (1998).
[50] Cristian T. I. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, (2003), 85(6): 317-325.
[51] R. K. Miller and A. Feldstein. Smoothness of solutions of Volterra integral equations with weakly singular kernels. SIAM J. Math. Anal. 2 (1971), 242-258.

A. Zerarka is a Professor of Physics and Applied Mathematics at the University Med Khider at Biskra, Algeria. He obtained his PhD from the University Bordeaux, France. His domain of interest is physics and applied mathematics. He has been teaching and conducting research since 1981 and has widely published in international journals and the reviewer of numerous papers. He is an active member of the Academy of Science, New York.

A. Soukeur is a PhD student at Department of Applied Mathematics, University Med Khider at Biskra, Algeria. He holds a Graduate Diploma, DES in Mathematics and Post Graduate Diploma: Magister. His domain of interest is: applied mathematics.

N. Khelil is a Doctor at Department of Applied Mathematics, University Med Khider at Biskra, Algeria. He holds a Graduate Diploma, DES in Mathematics and Post Graduate Diploma: Magister. He obtained his Doctorat from the University Biskra. His domain of interest is: numerical analysis and applied mathematics.