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.

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

$$u_{ij}^{k+1} = w_{ij}v_{ij}^k + c_1r_1^k[(p_{best})_{ij}^k - x_{ij}^k] + c_2r_2^k[(s_{best})_{ij}^k - x_{ij}^k]$$

(1)

$$x_{ij}^{k+1} = v_{ij}^{k+1} + x_{ij}^k$$

(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 is set equal to zero). In the relation (1), $v_{ij}^{k+1}$ is replaced by $v_{ij}^{k+1}/\sigma$, where $\sigma$ 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 $p$bests and $s$best 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 provide 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

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_{1i}$</th>
<th>$x_{2i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0110</td>
<td>10.2281</td>
</tr>
<tr>
<td>2</td>
<td>-0.2000</td>
<td>10.2402</td>
</tr>
<tr>
<td>3</td>
<td>-0.3122</td>
<td>10.2877</td>
</tr>
<tr>
<td>4</td>
<td>-0.3423</td>
<td>10.3395</td>
</tr>
<tr>
<td>5</td>
<td>-0.3832</td>
<td>10.3470</td>
</tr>
<tr>
<td>6</td>
<td>-0.4057</td>
<td>10.5138</td>
</tr>
<tr>
<td>7</td>
<td>-0.6617</td>
<td>10.5934</td>
</tr>
<tr>
<td>8</td>
<td>-0.6937</td>
<td>10.6100</td>
</tr>
<tr>
<td>9</td>
<td>-0.7023</td>
<td>10.7072</td>
</tr>
<tr>
<td>10</td>
<td>-0.9183</td>
<td>10.8737</td>
</tr>
<tr>
<td>11</td>
<td>-0.9791</td>
<td>11.1006</td>
</tr>
<tr>
<td>12</td>
<td>-1.0000</td>
<td>11.1257</td>
</tr>
</tbody>
</table>

### Table II

The new candidates for the interpolating points $x_{1}$ and $x_{2}$ generated by PSO algorithm for the examples 1 and 2 respectively. Number of points $N = 11$.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_{1i}$</th>
<th>$x_{2i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0110</td>
<td>10.2281</td>
</tr>
<tr>
<td>2</td>
<td>-0.2000</td>
<td>10.2402</td>
</tr>
<tr>
<td>3</td>
<td>-0.3122</td>
<td>10.2877</td>
</tr>
<tr>
<td>4</td>
<td>-0.3423</td>
<td>10.3395</td>
</tr>
<tr>
<td>5</td>
<td>-0.3832</td>
<td>10.3470</td>
</tr>
<tr>
<td>6</td>
<td>-0.4057</td>
<td>10.5138</td>
</tr>
<tr>
<td>7</td>
<td>-0.6617</td>
<td>10.5934</td>
</tr>
<tr>
<td>8</td>
<td>-0.6937</td>
<td>10.6100</td>
</tr>
<tr>
<td>9</td>
<td>-0.7023</td>
<td>10.7072</td>
</tr>
<tr>
<td>10</td>
<td>-0.9183</td>
<td>10.8737</td>
</tr>
<tr>
<td>11</td>
<td>-0.9791</td>
<td>11.1006</td>
</tr>
<tr>
<td>12</td>
<td>-1.0000</td>
<td>11.1257</td>
</tr>
</tbody>
</table>

### Table III

Parameters settings to generate the PSO algorithm for both examples. Case $N = 21$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>500</td>
<td>600</td>
</tr>
<tr>
<td>Acceleration Coefficients: $c_1$ and $c_2$</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Inertial Weight</td>
<td>1.2 to 0.4</td>
<td>1.2 to 0.4</td>
</tr>
<tr>
<td>Desired Accuracy</td>
<td>$10^{-6}$</td>
<td>$10^{-4}$</td>
</tr>
</tbody>
</table>

A. Example 1

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

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

$$f(x) = \varphi(x) + \lambda \int_0^x K(x, s)f(s)ds, \quad 0 \leq x \leq T$$  \hspace{1cm} (4)

where $\lambda$ 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

$$U(x) \equiv \int_0^x K(x, s)f(s)ds$$  \hspace{1cm} (5)

then $U(x)$ may be approximated by

$$U(x_m) = \sum_{j=0}^N C_{m,j}K(x_m, x_j)f(x_j), \quad m = 0, \ldots, N,$$  \hspace{1cm} (6)

$$U_h(x_m) = \sum_{j=0}^N C_{m,j}K(x_m, x_j)P_N(x_j),$$  \hspace{1cm} (7)

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

$$C_{m,j} = \frac{1}{K(x_m, x_j)} \int_0^{x_m} K(x_m, s)P_N(x_j)ds, \quad 0 \leq j \leq N$$  \hspace{1cm} (8)

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

$$C_{i,j} = \int_0^{x_m} K(x_i, s)ds - \sum_{j=0,j \neq i}^N C_{i,j}K(x_i, x_j), \quad \text{for} \ i = 0, \ldots, N,$$  \hspace{1cm} (9)
where \( \tilde{K}(x_m, x) = \frac{K(x_m, x)}{K(x_m, x_m)} \).

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

\[
 f(x) = \sum_{j=0}^{N} f(x_j) P_{N,j}(x). \tag{10}
\]

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 \( \Pi_N \) in terms of \( N \) degree polynomials \( P_{N,j}(x) \)

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

with uniform nodes \( \Pi_N = O \left( \frac{N}{N^{1/2}} \right) \). 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.,

\[
 f(x) = 1 - \int_0^x (x-s)^{-\frac{1}{2}} f(s) ds. \tag{12}
\]

The above equation has the exact solution \( f(x) = \exp(\pi x)(1-\text{erf}(\sqrt{\pi} x)) \), 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.

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

REFERENCES


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.