Correction of Meshless FPM Interpolation Sub-domains Using Genetic Algorithms

Luis PEREZ P.[1]**, Fernando PEREZ**[2]**, Orlando DURÁN**[2]

[1] Universidad Técnica Federico Santa María,
Valparaíso, Chile,
luis.perez@usm.cl

[2] Pontificia Universidad Católica de Valparaíso,
Valparaíso, Chile,
orlando.duran@ucv.cl

Abstract: This work describes a technique that allows the correction of interpolation sub-domains, using metaheuristics for the implementation of Meshless Methods in the resolution of Partial Differential Equations. A combined strategy has been used based on a specifically developed Genetic Algorithm (GA), and searching and optimization technique. Three sample problems were solved in this work. Results show a decrease in the global error as well as in the quantity of points needed to form the domain.

1. Introduction

The meshless or meshfree methods are a family of new numerical techniques that do not require a mesh. In these methods, the body or domain is discretized by a collection of points or nodes. It is divided into local interpolation sub-domains, also called clouds, consisting of one central point, or star node, and several neighbouring points. Generally, these methods are computationally efficient and easy to implement and they have been successfully used in several problems of solid and fluid mechanics.

A detailed review of the most relevant meshless methods and their connections is presented in [1]. An analysis and classification of the most important meshless methods is presented in [2]. Advantages and disadvantages of this techniques are also discussed. A review and important aspects of computer implementation are presented in [3].

The Finite Point Method (FPM) was proposed in [4] with the initial purpose of solving convective transport and fluid flow problems. Later, its application was extended to advection-diffusion transport [5] and incompressible flow problems [6]. In the context of solid mechanics, FPM has been applied successfully in elasticity [7] and non-linear material behaviour problems [8]. The lack of dependence on a mesh or integration procedure is an important feature, which makes the FPM a truly meshless method. A crucial phase of the FPM is the definition of subdomains to perform the local approximations. This paper describes a technique that allows the formation and eventual corrections of the interpolation sub-domains, using Genetic Algorithms. The next section discusses the FPM. Then, in section 3, the Genetic Algorithms are briefly presented. Three sample problems were solved in section 4. Finally, in section 5, conclusions are depicted.

2. The Finite Point Method

The FPM approximates the local solution of a partial differential equation in each point of the discretized domain by means of a weighted least squares technique and a point collocation procedure in order to obtain the final system of discrete equations. Due to the local character of the approximation utilized by this method, it is necessary to define a sub-domain for each node that considers only the nearest nodes. This point collection is called a cloud and its referential central point is the star node. A relevant aspect in the definition of clouds is that their superposition must produce the whole domain:

$$\bigcup_{k=1}^{n_p} \Omega_k = \Omega \tag{1}$$

where n_p is the total number of nodes.

Another important aspect in the definition of clouds, which is more difficult to quantify, is that the clouds associated with nearby nodes must overlap enough to ensure the global convergence of the local least-square approximations.

Let $\hat{u}(x)$ be a local approximation of the function $u(x)$ in the cloud $\Omega_k$associated with the star node x_k, as a linear combination of known functions $p(x)$,

$$u(x) \cong \hat{u}(x) = p^T(x) \cdot \alpha, \quad \forall x \in \Omega_k \tag{2}$$

where $p(x)$ is the basis of m linearly independent functions only valid in Ω_k and α is a vector of constant parameters. The elements of the interpolation base may belong to any functions family. Nevertheless, because of simplicity the *m* first monomial polynomials are used.

Because Equation (2) is valid for each of the *n* points of the k-th sub-domain, the approximations $\hat{u}(X)$ conform a Vandermonde system given by the following relation:

$$u(X^k) \cong \hat{u}(X^k) = P(X^k) \cdot \alpha \qquad (3)$$

where

$X^k = [x_1, \dots, x_n]^T$,

$u(X^k) = [u(x_1), \dots, u(x_n)]^T$,

$\hat{u}(X^k) = [\hat{u}(x_1), \dots, \hat{u}(x_n)]^T$,

$P(X^k) = [p(x_1), \dots, p(x_n)]^T$,

$\alpha = [\alpha_1, \dots, \alpha_n]^T$

In general, the number of points *n* conforming the cloud is greater than the number of functions *m* defining the basis, hence the matrix $P(X^k)$ is usually rectangular, i.e. the interpolation property is lost and the problem is transformed into a numerical approximation. The coefficients of the vector α must be determined in such a way that the weighted sums of the square differences between the exact values $u(x)$and the approximated values $\hat{u}(x)$ of each point is minimized. This is shown in Figure 1 and established by the following expression:

$$\min\left\{\sum_{j=1}^{n} w(x_j) \cdot (u(x_j) - \hat{u}(x_j))^2\right\} \qquad (4)$$

where $w(x_j)$ is a fixed weighting function defined in Ω_k and evaluated in the node x_j.

The minimization process described by Equation (4) leads to the following expression for the vector α:

$$\alpha = A^{-1}(X^k)B(X^k)u(X^k) \qquad (5)$$

where

$A(X^k) = P(X^k)W(X^k)P^T(X^k)$

$B(X^k) = P^T(X^k)W(X^k)$

$W(X^k) = [w(x_1), \dots, w(x_n)]^T$

Finally, replacing Equation (5) in (2), the following relation is obtained:

$$\hat{u}(x) = N^T(x)\, u(X^k) \qquad (6)$$

with

$N(x) = p^T(x)C(X^k)$

$C(X^k) = A^{-1}(X^k)B(X^k)$

The weighting function is built in order to have unit values near the star node and zero values outside the Ω_k sub-domains. The common selection in the FPM is the normalized Gaussian given by:

$$w(x_j) = \begin{cases} \frac{e^{-d_j/c} - e^{-r/c}}{1 - e^{-r/c}} & \text{if } d_j \le r \\ 0 & \text{if } d_j > r \end{cases} \qquad (7)$$

with $d_j$being the distance between the star node and the point x_j, d_{max} (max. of d_j) as a reference distance and $c = \beta r$. A detailed description of the effects of the parameters q and β on the numerical approximation and

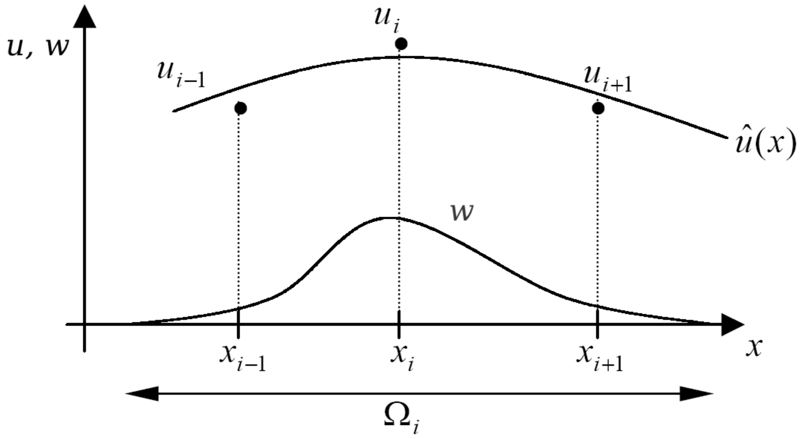


Figure 1. Concept of fixed weighted least squares interpolation.

some guidelines for their setting has been presented in [9]. Other possibilities for selecting the function $w(x_j)$ can be found in [10].

Discretization of the equations system

Let Ω_S be the solution domain of a boundary value problem and $\Gamma = \Gamma_t \cup \Gamma_u$be the boundary. Let us assume that the problem is governed by the following set of equations:

$$\mathrm{A}(\mathrm{u}) = \mathrm{b}, \text{in } \Omega_S$$

$$\mathrm{B}(\mathrm{u}) = \mathrm{t}, \text{in } \Gamma_t(\text{Neumann}'\text{s condition}) \qquad (8)$$

$$\mathrm{u} - \mathrm{u}_p = 0, \text{in } \Gamma_u(\text{Dirichlet}'\text{s condition})$$

where A and B are differential operators, u is the unknown function, b and t represent a forcing function in Ω_S and Γ_t, and u_p are prescribed values u in Γ_u. A general procedure for solving this class of problems is the Weighted Residual Method (WRM) [10]. In the FPM a point collocation scheme is used, in this way the following set of discrete equations are obtained:

$$[\mathrm{A}(\hat{\mathrm{u}})]_i - \mathrm{b}_i = 0$$

$$[\mathrm{B}(\hat{\mathrm{u}})]_j - \mathrm{t}_j = 0 \qquad (9)$$

$$\hat{\mathrm{u}}_k - \mathrm{u}_p = 0$$

where $\hat{u}_i$:

$$\hat{\mathrm{u}}_i = \sum_{j=1}^{n_p} \mathrm{N}_j^T \mathrm{u}_i \qquad (10)$$

where n_p is the total number of nodes and the shape function N_j^T is built according to the least squares formulation shown in the previous section.

A stabilized form of the governing equation is derived from the Finite Calculus (FIC) procedure. The FIC method is based on imposing the typical balance laws over a domain of finite size. The unknown fields are then approximated within the finite domain using a Taylor series expansion, retaining higher order terms than those used in the standard infinitesimal approach [4]. This introduces new terms in the governing differential equation which has stabilizing features. The stabilized forms of Equation (8) using the FIC method are [11].

$$\left[\mathrm{A}(\hat{\mathrm{u}}) - \frac{1}{2}\mathrm{h}_k \frac{\partial \mathrm{A}(\hat{\mathrm{u}})}{\partial \mathrm{x}_k}\right]_k - \mathrm{b}_k = 0$$

$$\left[\mathrm{B}(\hat{\mathrm{u}}) - \frac{1}{2}\mathrm{h}_k \mathrm{n}_k \mathrm{A}(\hat{\mathrm{u}})\right]_k - \mathrm{b}_k = 0 \qquad (11)$$

$$\hat{\mathrm{u}}_k - \mathrm{u}_p = 0$$

where n_k are the components of the unit normal to the boundary $\Gamma_t$and h_k are characteristic length parameters. The efficiency of the FIC stabilization procedure, in the context of FPM, has been shown in [4,11].

3. Genetic Algorithms

The basic concepts of Genetic Algorithms (GA) were developed by Holland [12]. GAs have been successfully used to solve a variety of optimization problems [13, 14, 15, 16, 17]. Combinatorial problems constitute a class of problems particularly difficult to solve. Every solution to the addressed problem is called a chromosome. A chromosome is a string of binary bits or numbers. The pool of solutions is called the population. New candidates are generated gradually from a set of renewed populations by applying genetic operators selected from strategies based on the survival of the fittest principle, after repeatedly using operators of crossover and mutation. GAs need only a fitness or objective function value to lead the optimization process.

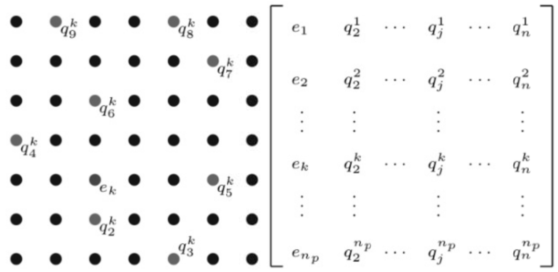
The main advantage of GAs lies in the fact that derivatives or gradients are not necessary. In addition, GAs use probabilistic transition rules to find new design points for exploration rather than deterministic rules based on gradient information to find these new points. A random initial population initiates the evolutionary process. Each chromosome is assigned a positive value called fitness proportional to the quality of the solution. Based on their fitness value, subsets of the chromosomes of the current population are selected for reproduction. The reproduction is accomplished by applying a set of genetic operators, called crossover and mutation, to the chromosomes. To promote the evolution of the population best adapting to the problem solution, it is necessary to establish the degree of goodness of each individual and compare it to the rest of the population. Therefore, a fitness function is to be defined as a means of comparison and classification between the individuals of the population.

- **Implementation**

In this work e_k represents the star node associated to the cloud Ω_k. A $n_k \times n$ matrix is used to represent the global set of sub-domains made up by n points,

$$\Omega_k = \{e_k, q_2^k, \dots, q_j^k, \dots, q_n^k\} \tag{12}$$

where q_j^k are each of the remaining points belonging to the cloud Ω_k.


q_9^k
q_8^k
q_7^k
q_6^k
q_4^k
e_k
q_5^k
q_2^k
q_3^k
e_1 q_2^1 ⋯ q_j^1 ⋯ q_n^1
e_2 q_2^2 ⋯ q_j^2 ⋯ q_n^2
e_k q_2^k ⋯ q_j^k ⋯ q_n^k
e_{n_p} $q_2^{n_p}$ ⋯ $q_j^{n_p}$ ⋯ $q_n^{n_p}$

(a) (b)

Figure 2. a) Cloud of node e_k.
b) Matrix of the domain.

- ### Codification

In Figure 2.b each row represents the elements forming a particular cloud, where the first element represents the star node of each cloud. So that, the chromosome is comprised of a string of integer numbers in the interval [1; n_p], as shown in Figure 3. The length of the string is given by the number of n_p points that conform the domain, and each one of the *n* clouds. The chromosome assembly is made by the union of the set of points that are linked to each one of the clouds (Figure 4).

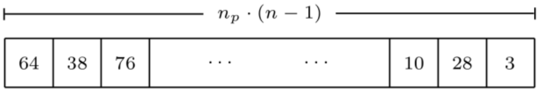


Figure 3. Chromosome structure

- ### Initial Population Generation

The initial generation is obtained based on interpolation sub-domains developed by the minimum distance method [1,2,3].

- ### Objective function

The proposed genetic algorithm has the objective of defining a set of point clouds, so that the global error L_2 among the solution is obtained by the GA and the exact solution u^e is obtained by analytical methods.

$$L_2 = \sqrt{\frac{\sum_{i=1}^{n_p}(u_i - \hat{u}_i)^2}{\sum_{i=1}^{n_p}(u_i)^2}} \tag{13}$$

- ### Fitness function

The fitness function f_k of each individual is given by:

$$f_k = \frac{1}{L_2} \tag{14}$$

- ### Selection

In this process, the number of times a specific individual is selected to reproduce itself is involved. The selected individuals for reproduction are called parents. Tournament selection is used as the selection process.

- ### Genetic Operators

In this GA a specific two-point crossover type of operator was implemented. After the crossover process was applied and a new generation of individuals was generated, a mutation operator was applied to a portion of the new individuals. Both operators act randomly, guided by a crossover probability P_c and a Mutation Probability P_m, respectively. Given the use of an integer notation, the application of crossover and mutation operators, some generated individuals may be considered violating at least one constraint of the problem. The unique constraint that can be violated after

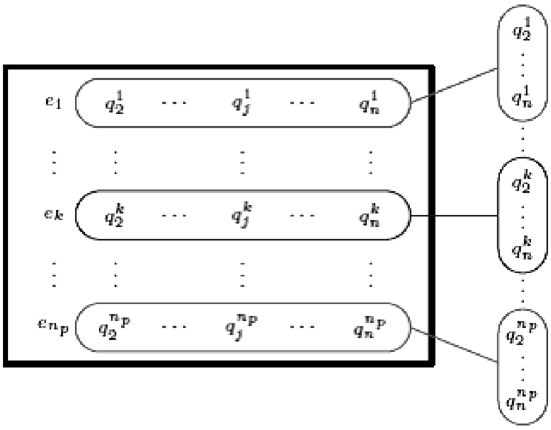

e_1 q_2^1 ⋯ q_j^1 ⋯ q_n^1
e_k q_2^k ⋯ q_j^k ⋯ q_n^k
e_{n_p} $q_2^{n_p}$ ⋯ $q_j^{n_p}$ ⋯ $q_n^{n_p}$
q_2^1 q_n^1
q_2^k q_n^k
$q_2^{n_p}$ $q_n^{n_p}$

Figure 4. Chromosome creation.

genetic operations have been performed is that which does not admit a given point or node can be repeated in a given cloud. Those unfeasible individuals have to be repaired.

- **Cloning**

Because of the use of the aforementioned genetic operators, there is no certainty the best individuals subsist from one offspring to the next. This fact may cause the loss of some desirable characteristics that well fitted individuals have presented before the application of the genetic operators. The preservation of the best individuals, from a previous generation, into the next offspring is called elitism. To avoid the loss of that important genetic information, an elitist strategy was implemented. That strategy is called cloning, which involves copying the best fitted individual of the generation into the next offspring. The whole process is summarized in Figure 5.

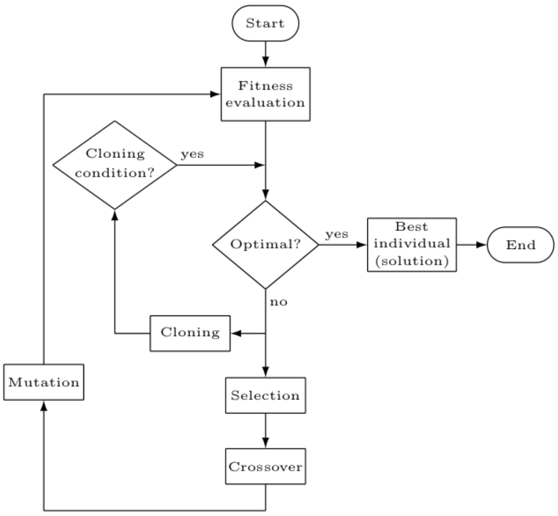


Figure 5. Proposed Genetic Algorithm flowchart

There are a number of parameters to be specified in the development and implementation of a GA. The population size (a discrete positive integer). It is well-known that a very small population may result in premature convergence, while a very large population may result in a slow convergence rate. Both the crossover and mutation rates are continuous variables within [0, 1]. These two values are important for controlling the balance between exploration and exploitation. Therefore, prior to the use of a GA, an adjustment of its parameters is to be performed (parameter tuning). This tuning is intended to obtain a better understanding of the effect that population size change, the crossover and mutation parameters may cause in the performance of the GA. To perform that tuning, a test function was used. This test function corresponds to the differential equation defined by:

$$u_{xx} + u = 2, \quad 0 \leq x \leq \frac{3\pi}{2} \qquad (15)$$

and

$$u = 2 \qquad x = 0$$

$$u = 3 \qquad x = \frac{3\pi}{2}$$

where the exact solution of this equation is (Figure 6):

$$u = -\sin(x) + 2 \qquad (16)$$

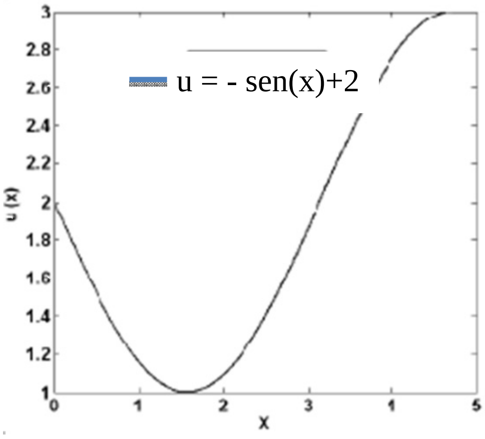


Figure 6. Exact Solution, Test (Tunning) function.

To obtain the exact solution through the use of FPM, the following parameters were used:

$x \in \Omega, \qquad \Omega = [0, 5]$

- Ω domain discretization with 15 equidistant points.
- Clouds generation using the minimum distance criteria [4].
- Clouds with 5 points each.

The global error presented by the approximated solution of the FPM is $L_2 = 0.0039$. By comparing the global errors obtained by the application of the GA with the exact solution, it was possible to determine which parameters presented a better performance. The values of each parameter used in this tuning process are listed below:

Q = [50, 100, 150]

Pc = [20, 40, 60, 80, 90, 100]

Pm = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.5, 2.0]

Considering all possible combinations, a set of 198 test problems were run (3 x 6 x 11). Table 1 shows the global results of the tuning experiment. Note the results that are better than those obtained by the FPM are named as “Number of success”.

Table 1. Summary of best results obtained from the tuning experiment.

	Q=50	Q=100	Q=150
Number of success	3	29	48
% of success	4,5	43,9	72.7
Best L2 average	0,0033	0,0022	0,0019
Best Std.Dev. of L2	0,0004	0,0004	0,0002

Optimal Parameters	Pc=90
	Pm=0,6
		Pc=90
		Pm=0,6
			Pc=90
			Pm=0,6

4. Numerical Examples

1D Poisson problem

This problem can be found in ref. [18] where is clearly shown the solution changes abruptly in the vicinity of x = 0. The following differential equation is solved:

$$-u_{xx}=6x-\left(40000x^2-200\right)e^{-100x^2}, x \in [-1,1] \quad (17)$$

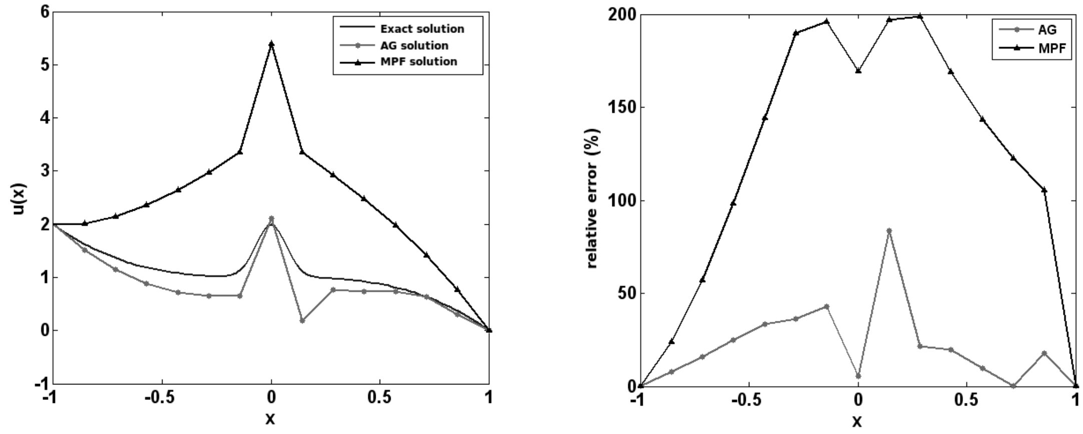
with essential boundary conditions given by

$$u(-1) = 2 + e^{-100}, \quad u(1) = e^{-100} \quad (18)$$

and the exact solution:

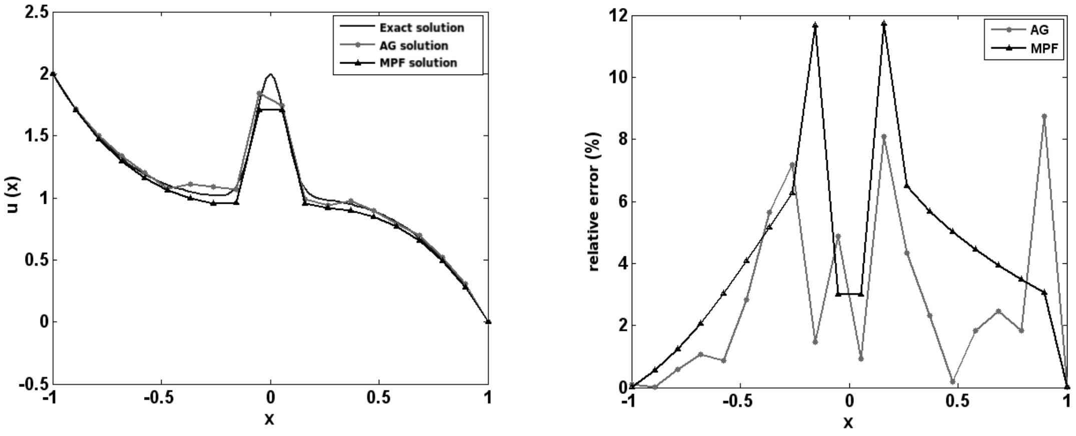
$$u^T(x) = 1 - x^3 - e^{-100x^2} \quad (19)$$

Figures 7, 8 and 9 show a Peak convergence of FPM and AG for discretization of 15, 20 and 25 points, respectively, which is explained by the use of a small quantity of points in the domain discretization. Despite the low convergence obtained through both methods, we can observe a better performance of AG with respect to the approximation by the FPM, which may be appreciated in a low absolute error value and a more uniform distribution of such parameter.



a) *b)*

Figure 7. a) Solution convergence for 1D Poison equation.
b) Convergence of the relative error for n = 15 pts.



a) *b)*

Figure 8. a) Solution convergence for the 1D Poison equation.
b) Convergence of the relative error for n = 20 pts.

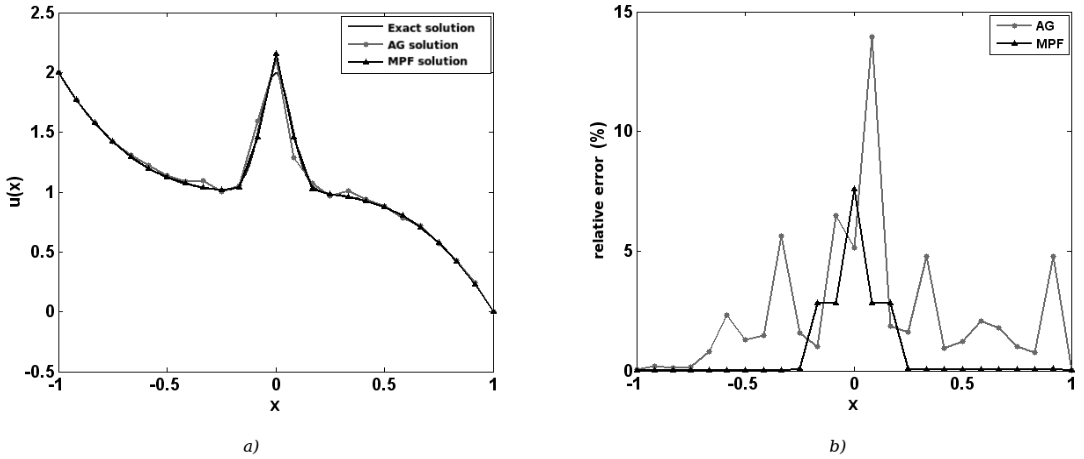


Figure 9. a) Solution convergence for the 1D Poison equation.
b) Convergence of the relative error for $n = 25$ pts.

Figures 10, 11 and 12 clearly show the reduction of L_2 as the generation number increases, which directly affects the evolutionary process of AG on the individuals conforming the population, through the association of points for the generation of better-quality clouds.

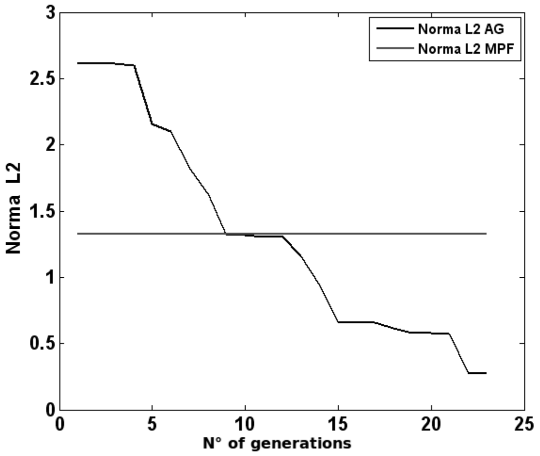


Figure 10. Convergence of the L_2 –norm for the 1D Poison equation for $n = 15$ pts.

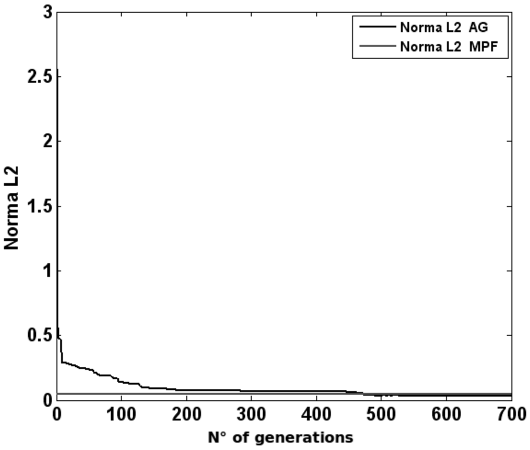


Figure 11. Convergence of the L_2 –norm for the 1D Poison equation for $n = 20$ pts.

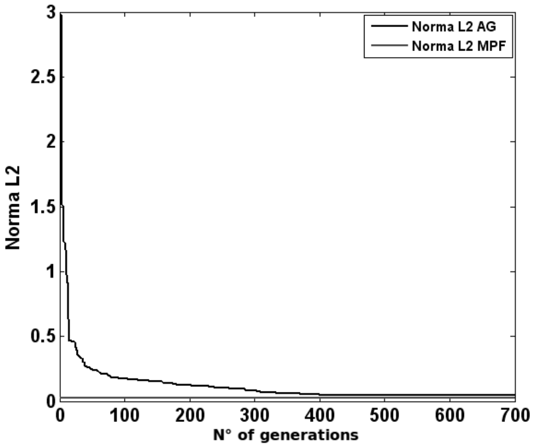


Figure 12. Convergence of the L_2 –norm for the 1D Poison equation for $n = 25$ pts.

Table 2 shows a summary of the results obtained in the resolution of the 1D Poison equation using different number of points per domain.

Table 2. Main results for the 1D Poison equation.

	Number of Points in Domain		
	15	**20**	**25**
L2(GA)	0,2733	0,0321	0,0453
L2(FPM)	1,3233	0,0462	0,0413
maxer(%)	81,1	8,75	13,96
time[s]	7,47	631,20	732,70
Pm(%)	0,6	0,8	1,0
Pc(%)	100	100	100
No.ofGener.	22	700	700
Q	50	100	100

- **Temperature distribution on an isolated boundary rectangular plate**

The temperature distribution in stable state for a rectangular plate, where the boundaries are thermally isolated from the exterior environment, is reduced to the resolution of the Laplace equation subjected to Neumann and Dirichlet-type boundary conditions.

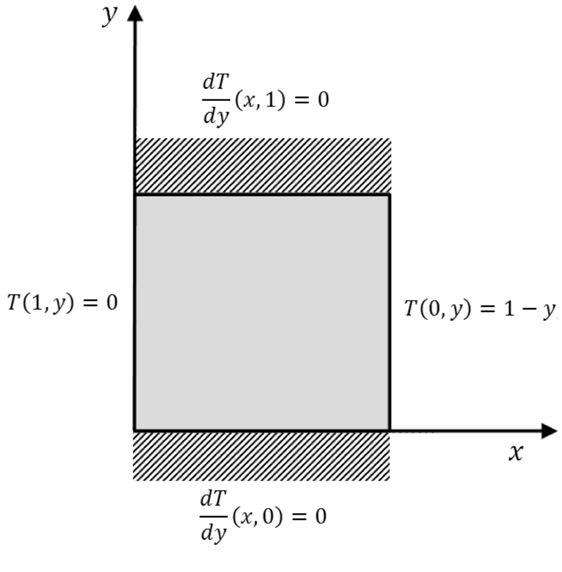


Figure 13. Boundary conditions for a rectangular plate with isolated boundaries

The example developed involves a rectangular plate (Figure 13), which temperature in stable state $\mathrm{T(x,y)}$ is described by the following equation:

$$\mathrm{T_{xx}+T_{yy}=0, \quad 0\le x\le 1,\ 0\le y\le 1} \quad (20)$$

with the following boundary conditions:

$$\begin{aligned}&\mathrm{T_x(x,0)=0,\ 0\le x\le 1}\\&\mathrm{T_y(x,1)=0,\ 0\le x\le 1}\\&\mathrm{T(0,y)=1-y,\ 0\le y\le 1}\\&\mathrm{T(1,y)=0,\ 0\le y\le 1}\end{aligned} \quad (21)$$

The analytical solution for this problem is obtained as serial development given by:

$$\mathrm{T(x,y)}=\frac{1}{2}\mathrm{x}+$$

$$+\frac{2}{\pi^2}\sum_{\mathrm{n}=1}^{\infty}\frac{1-(1)^{\mathrm{n}}}{\mathrm{n}^2\sin(\mathrm{n}\pi)}\sin(\mathrm{n}\pi\mathrm{x})\cos(\mathrm{n}\pi\mathrm{y}) \quad (22)$$

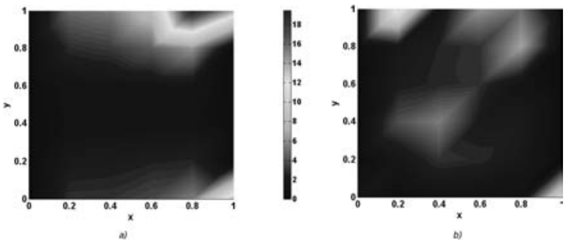


Figure 14. Relative error distribution. Rectangular plate with isolated boundaries, n = 36 pts.

Bidimensional domains on which the following problem will be analyzed will be discretized on rectangular domains of:

Pm = 0.7% and 0.4%
Pc = 100%
Cloning operator
Q = 70 individuals

Results obtained for discretization of 36, 49 and 64 points (Figures 14, 15 and 16) allow observing an erratic distribution of the AG absolute error compared to FPM. However, the maximum error values for AG (max $\mathrm{e_r}$(%) AG), Table 3, are always smaller than the maximum error values obtained for FPM (max $\mathrm{e_r}$(%) FPM), Table 3, which have a direct effect on the global error values of the problem assessed through the $\mathrm{L_2}$ norm. In each studied discretization of the FPM a chaotic growth of the local error can be observed in those areas with a high gradient as a result of the joint imposition of the Neumann and Dirichlet conditions in these areas.

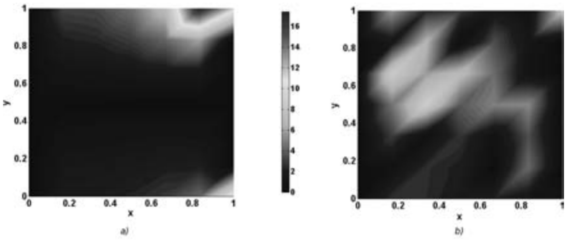


Figure 15. Relative error distribution. Rectangular plate with isolated boundaries, n = 49 pts.

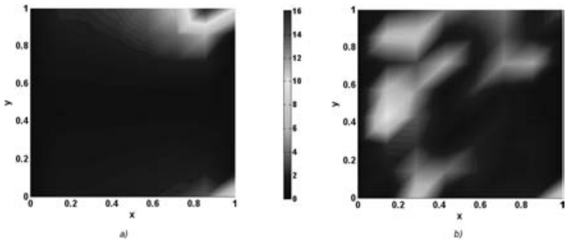


Figure 16. Relative error distribution. Rectangular plate with isolated boundaries, n = 64 pts.

Table 3. Main results for the rectangular plate with isolated boundaries.

	Number of Points in Domain		
	15	**20**	**25**
L2(GA)	0,0557	0,0387	0,0393
L2(FPM)	0,0984	0,0717	0,0547
maxer(%) GA	12,79	8,84	10,13
maxer(%) FPM	19,40	17,42	16,05
time[s]	248	845	2796
Pm(%)	0,4	0,4	0,4
Pc(%)	100	100	100
No.ofGenerat.	180	279	700
Q	60	70	70

Figure 17 shows the behavior of L_2 as the generation number varies. That variation clearly affects the evolutionary process of AG on the individuals of the population (through the association of points for the generation of better-quality clouds).

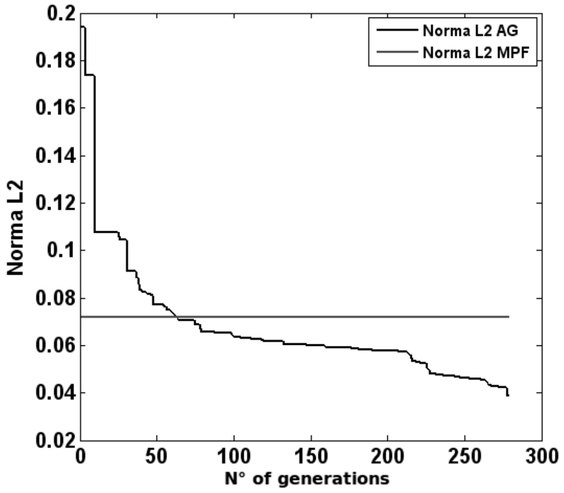


Figure 17. Convergence of the L_2norm, rectangular plate with isolated boundaries for n = 49 pts.

5. Conclusion

A methodology basedon Genetic Algorithms has been presented for the development of interpolation sub-domains for the application of meshless methods. The characteristics of an AG have been defined in order to represent efficiently the problem of cloud generation on domains regularly discretized. The performance of the suggested methodology was subsequently studied for the solution of 1D and 2D problems applied to linear elasticity, heat transfer andresolution of differential equations in partial derivatives.

In general terms, the developed strategy has shown an adequate performance in the solution of approached problems; the latter in the light of the positive results based on the analysis of both the relative and global error of the established solutions. Therefore, this work allows establishing the following conclusions:

The significance of a proper mechanism generating clouds results clear in order to improve the quality of the Solutions found through meshless methods, especially the FPM.

The determination of the mutation and crossover parameters must be studied a priori to determine the best configuration for an adequate performance of the AG.

The encoding(using an integer code) properly adjusts to the configuration presented by the phenotype in the developed problems.

The increase in the point numbers where the domain is discretized negatively affects the performance of AG, thus increasing the convergence periods of the solution and consequently the computational cost.

The generation of interpolation sub-domains for the minimal distance criteria does not provide the best solutions for the domains discretized with a low quantity of points.

The implemented technique allows the quality improvement of the solutions, especially in those areas where the Neumann and Dirichlet conditions are imposed.

Future directions of our research point to the application of other metaheuristics for defining the subdomains. Comparisons will be made between Genetic Algorithms, Particle swarm Optimization, Simulated annealing, among others.

REFERENCES

1. OÑATE, E., S. IDELSOHN, O. ZIENKIEWICZ, R. TAYLOR, **A Finite Point Methods in Computational Mechanics, Application to Convective Transport and Fluid Flow**, International Journal for Numerical Methods in Engineering, vol. 39, 1996, pp. 3839-3866.

2. LIU, G., **Mesh Free Methods: Moving Beyond the Finite Element Method,** CRC press, USA, 2002.

3. GU. Y., **Meshfree Methods and Their Comparisons,** International Journal of Computational Methods vol. 4, 2005, pp. 477-515.

4. OÑATE, E., S. IDELSOHN, O. ZIENKIEWICS, R. TAYLOR, C. SACCO, **A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems**, Computer Methods in Applied Mechanics and Engineering, vol. 139, 1996, pp. 315-346.

5. OÑATE, E., S. IDELSOHN, **A Mesh Free Finite Point Method for Adjective Diffusive Transport and Fluid Flow Problems**, Computational Mechanics, vol. 21, 1998, pp. 283-292.

6. OÑATE, E., C. SACCO, S. IDELSOHN, **A Finite Point Method for Incompressible Flow Problems**, Computing and Visualization in Science, vol. 3, 2008, pp. 67-75.

7. PERAZZO, F., J. MIQUEL, E. OÑATE, **The Finite Point Method for Solid Mechanics**. Revista Internacional de Métodos Numéricos en Ingeniería, vol. 20, 2004, pp. 235-246 (spanish).

8. PÉREZ-POZO, L., PERAZZO, F., **Non-linear Material Behaviour Analysis using Meshless Finite Point Method**, in: 2nd ECCOMAS Thematic Conference on Meshless Methods, Porto, Portugal, 2007, pp. 251-268.

9. ORTEGA, E., E. OÑATE, S. IDELSOHN, **An Improved Finite Point Method for Three-Dimensional Potential Flows**, Computational Mechanics, vol. 40, 2007, pp. 949-963.

10. ZIENKIEWICZ, O., R. TAYLOR, **The Finite Point Method** (Spanish), vol. 1, Centro Internacional de Métodos Numéricos en Ingeniería, Barcelona España, 2008.

11. OÑATE, E., **Derivation of Stabilized Equations for Advective-difusive Transport and Fluid Flow Problems**, Computer Methods in Applied Mechanics and Engineering, vol. 151, 1998, pp. 233-267.

12. HOLLAND, J., **Adaptation in Natural and Artificial Systems**, University of Michigan Press. Ann Arbor, 1975.

13. GOLDBERG, D., **Genetic Algorithms in Search, Optimization, and Machine Learning,** Addison-Wesley Publishing Company, 1989.

14. HAUPT, R., S. HAUPT, **Practical Genetic Algorithms**, Wiley-Interscience, 2004.

15. MENDOZA, J. E., L. A. VILLALEIVA, M. A. CASTRO, E. A. LÓPEZ, **Multi-objective Evolutionary Algorithms for Decision-Making in Reconfiguration Problems Applied to the Electric Distribution Networks.** Studies in Informatics and Control, vol. 18 nr. (4), 2009, pp. 325-336.

16. EL FERCHICHI, S., K. LAABIDI, S. ZIDI, **Genetic Algorithm and Tabu Search for Feature Selection.** Studies in Informatics and Control, vol. 18, nr. (2), 2009, pp. 181-187.

17. NICOARA, E. S., F. Gh. FILIP, N. PARASCHIV, **Simulation-based Optimization Using Genetic Algorithms for Multi-objective Flexible JSSP.** Studies in Informatics and Control, vol. 20, nr. (4), 2011, pp. 333-344.

18. TIMOSHENKO, S, J. GOODIER, **Theory of Elasticity**, McGraw Hill, New York, 1970.