

Research Article

An Approach to Refinement of Irregular Clouds of Points Using Generalized Finite Differences

Luis Gavete,¹ María Lucía Gavete,¹ Francisco Ureña,² and Juan José Benito³

¹UPM, ETSIME, Alenza 4, 28003 Madrid, Spain

²UCLM, IMACI, Avenida Camilo José Cela, s/n, 13071 Ciudad Real, Spain

³UNED, ETSII, Apdo 60149, 28080 Madrid, Spain

Correspondence should be addressed to Luis Gavete; lu.gavete@upm.es

Received 23 March 2015; Revised 5 June 2015; Accepted 9 June 2015

Academic Editor: Salvatore Caddemi

Copyright © 2015 Luis Gavete et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper describes a fully automatic adaptive refinement procedure performed in conjunction with the generalized finite difference method (GFDM) for solving a second-order partial differential equation (PDE) frequently encountered in engineering practice. A quadtree structure is used to organize the clouds of points. The adaptive procedure is based on the existing differences between the gradients of close points. The relative error between two successive adaptation processes is taken as the criterion to stop the algorithm. Using this adaptive procedure, the high gradient regions are easily detected and refined. Some benchmark examples are presented showing the accuracy of the method.

1. Introduction

Considerable research in computational mechanics has been devoted to the development of meshless methods (see Liu [1], Li and Liu [2], and Nguyen et al. [3]). In these methods, the domain of interest is discretized by a scattered set of points. Meshless or meshfree methods can be classified according to the method of solution as weak, local weak, or strong form.

Some meshfree methods have used the weak form. The diffuse element method developed by Nayroles et al. [4] was the first meshless method based on the Galerkin method. Belytschko et al. [5] developed an alternative implementation using moving least squares approximation as defined by Lancaster and Salkauskas [6]. Several other meshless methods such as Partition of Unity Finite Element Method (PUFEM) by Babuška and Melenk [7], h-p cloud by Duarte and Oden [8], and natural element method (NEM) by Sukumar et al. [9] and Cueto et al. [10] have also been published.

Another possibility is to integrate the weak local formulation as in the meshless Petrov-Galerkin method (MLPG), reported by Atluri and Zhu [11, 12].

Another solution is to use the strong form. One of the earliest developments was the SPH method, introduced by

Monaghan [13, 14]. Liu et al. [15] proposed a different kind of gridless multiple scale method (RPKM) to improve the accuracy of the SPH method for finite domain problems. Another possibility is to solve the strong form using collocation method as in Oñate et al. [16].

Another important path in the evolution of the strong form has been the development of the generalized finite difference method (GFDM), also called meshless finite difference method. The bases of the GFDM were published by Jensen [17] and Perrone and Kao [18]. Liszka and Orkisz [19] and Orkisz [20] used moving least squares (MLS) in arbitrary irregular grids. Benito et al. [21] developed an implementation of the GFDM showing that the results obtained depend on the number of nodes in the cloud, the relative coordinates of the nodes with respect to the star node, and the weight function employed. Gavete et al. [22] reported improvements of GFDM and compared it with other meshless methods.

Adaptive methods based on the point density increase have been described in Benito et al. [23] and Benito et al. [24, 25] using a posteriori error indicators. A higher order approximation technique based on a modified multipoint approach has also been considered by Jaworska and Orkisz [26].

A totally different adaptive and fully automatic method is proposed in this paper. By combining the GFD and an adaptive numerical procedure the method can be used to analyze more complicated problems. The method proposed is based on the use of quadtree and on the computations of the gradients, using the differences between the gradients in each quadtree as a simple error indicator. The present approach provides certain restrictions on the location of the nodes due to the use of a quadtree structure. However, the clouds of points generated have an irregular shape. Different partial differential equations of second order are solved in the restrictive case of considering functions with high gradients.

The paper is organized as follows. Section 1 is Introduction. Section 2 describes the GFD. Section 3 describes the quadtree method used to organize the clouds of points. Section 4 describes the adaptive algorithm and shows the a posteriori error indicator used. Section 5 illustrates the performance of the adaptive method presented for solving second-order different partial differential equations with known analytical solutions. Finally, in Section 6 some conclusions are given.

2. Generalized Finite Difference Method

For any sufficiently differentiable function $U(x, y)$, in a given domain, the aim is to obtain explicit linear expressions for the approximation of partial derivatives at the points of the domain. First of all, an irregular grid or cloud of points is generated in the domain $\Omega \cup \Gamma$, where Γ is the boundary of the domain. On defining the central node with a set of nodes surrounding that node, the star then refers to a group of established nodes in relation to a central node. Each node in the domain has an associated star assigned to it. Following [21, 23–25], the explicit finite difference formulae for the first and second spatial derivatives with second-order approximation for the derivatives are obtained:

$$\begin{aligned} \frac{\partial^2 U_0}{\partial x^2} &= -m_{10}u_0 + \sum_{j=1}^N m_{1j}u_j + \Theta(h_j^2, k_j^2), \\ \frac{\partial^2 U_0}{\partial y^2} &= -m_{20}u_0 + \sum_{j=1}^N m_{2j}u_j + \Theta(h_j^2, k_j^2), \\ \frac{\partial^2 U_0}{\partial x \partial y} &= -m_{30}u_0 + \sum_{j=1}^N m_{3j}u_j + \Theta(h_j^2, k_j^2), \\ \frac{\partial U_0}{\partial x} &= -m_{40}u_0 + \sum_{j=1}^N m_{4j}u_j + \Theta(h_j^2, k_j^2), \\ \frac{\partial U_0}{\partial y} &= -m_{50}u_0 + \sum_{j=1}^N m_{5j}u_j + \Theta(h_j^2, k_j^2). \end{aligned} \quad (1)$$

By including the explicit expressions for the values of the partial derivatives $\partial^2 u_0 / \partial x^2$, $\partial^2 u_0 / \partial y^2$ in the equation

$$\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u_0}{\partial y^2} - f(x_0, y_0) = 0 \quad (2)$$

the star equation is obtained

$$\lambda_0 u_0 + \sum_{i=1}^N \lambda_i u_i - f(x_0, y_0) = 0 \quad (3)$$

and then

$$u_0 = \frac{f(x_0, y_0)}{\lambda_0} - \sum_{i=1}^N \frac{\lambda_i}{\lambda_0} u_i \quad (4)$$

with

$$\lambda_0 + \sum_{i=1}^N \lambda_i = 0. \quad (5)$$

Equation (4) is formed at each point, including the explicit expressions of these derivatives (1) in the Poisson partial differential equation (3). All points in the control scheme are called a *star* of nodes. The number and the position of nodes in each star i ($i = 1, \dots, N$) are the decisive factors affecting GFD formula approximation. The choice of these supporting nodes is constrained as particular patterns lead to degenerated solutions (Syczewski and Tribillo [27]).

As star selection criterium we follow the denominated cross criterium: the area around the central nodal point, 0, is divided into four sectors corresponding to quadrants of the Cartesian coordinates system originating at the central node. In each sector two or more nodes are selected, the closest to the origin. If this is not possible, for example, at the boundary, missing nodes can be added to provide the total number of nodes necessary in each star.

Having calculated the values of u_i in the nodes of the domain, we calculate derivatives using formula (1). It is possible to control the precision of GFD solutions by calculating the residual at each point of the interior of the domain using (1) and (2). In order to provide the required and controlled precision of the GFD method, residuals of (2) may be very small and with smoothed distribution over the entire domain. The existence of ill-conditioned stars of nodes depends on the weighting function w_i employed and on the number of nodes by quadrant in each star of nodes, but it can be avoided using the quadtree method shown in the next section.

3. Quadtree Method

A quadtree is a method of organizing data in four quadrants. It is a hierarchical data structure based on the principle of recursive decomposition. A quadtree structure requires each internal node to have exactly four children nodes. When illustrating most quadtree structures, a node that has four child nodes hanging from it is shown, with lines connecting the parent node with its children nodes. The illustration can continue, with four more child nodes hanging from each of the original four child nodes.

The GFD is a meshfree method; however, it is easy to create an initial simple quadrilateral data structure, as shown in Figure 1(a), as an initial quadtree for the cloud of points.

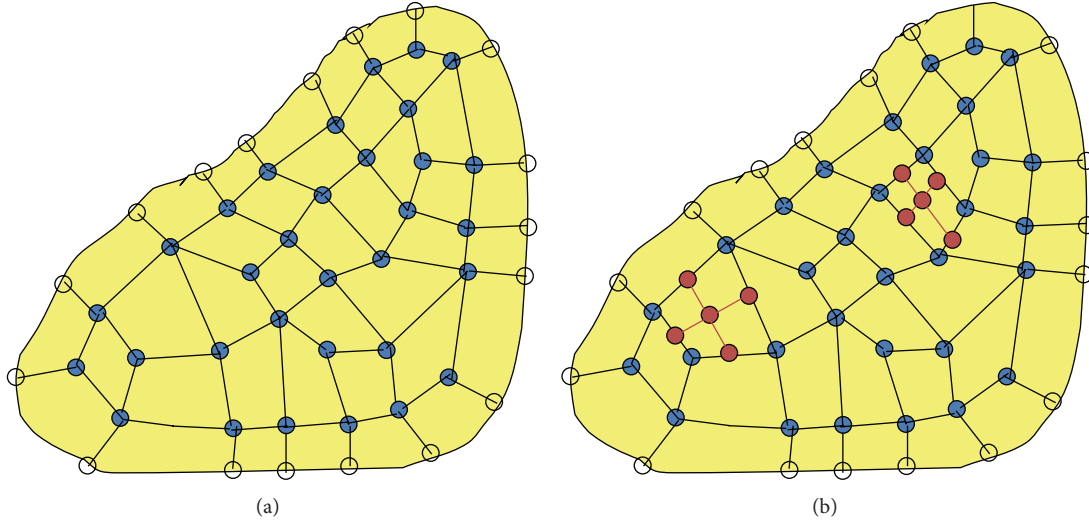


FIGURE 1: ((a) and (b)) The quadtree method with quadrilaterals.

Note that the only restriction used in this paper is that all the quadrilaterals used are convex.

An adaptive method using a quadtree can be implemented using an error indicator. Note that the GFDM uses the quadtree structure only to refine the cloud of points; see Figures 1(a) and 1(b) where two quadtree cells are refined. The GFDM with quadtree structure is a fully meshless method. The nodes of each star are selected using the previously defined cross criterion, so there is no relation between the nodes in the quadtree and the nodes in a star of GFDM.

The reason for combining the GFDM and an adaptive numerical procedure is to ameliorate the accuracy of this meshless method for analyzing problems governed by second-order partial differential equations in 2D.

It is interesting to note that not only the number but also the location of the nodes influences the FD operators precision. At the same time it is necessary to take into account that increasing the number of nodes of a cloud of points may decrease the solution error, but it does not change precision of the FD operators.

This organization of a cloud of points using a quadtree structure forces the stars of points used in the GFDM to be in equilibrium; therefore the existence of ill-conditioned stars of nodes is restricted. Also, it all depends on the PDE to be solved.

We can decrease the error by increasing the number of nodes in the cloud. It is well known that the error decreases when the number of nodes is increased in a regular mesh. However, in the case of an irregular mesh, the addition of nodes must be selective, and a solution to this problem is to use the adaptive algorithm described in the following paragraph.

4. Adaptivity Strategy

As is well known, mesh refinement is important in order to capture an accurate solution of a problem. In meshless

integral-free methods, a cloud refinement procedure based on gradients can be important and suitable for implementation. Previous gradient based methods have been used by Stein and Rust [28] for finite elements and by Luo and Häussler-Combe [29] for the element-free Galerkin method. The basic idea of the gradient based methods is simple: a larger variation of the gradients needs a richer cloud of nodes.

In this paper a procedure like the one used by Luo and Häussler-Combe [29] has been adapted to the GFDM, but with the following differences:

(1) In each one of the points of the domain, the first derivatives are calculated using (1) and then the gradients at the point are calculated:

$$\sqrt{\left(\frac{\partial u_0}{\partial x}\right)^2 + \left(\frac{\partial u_0}{\partial y}\right)^2}. \quad (6)$$

(2) In each quadrilateral cell, we calculate the average of the differences of the gradients between the nodes of each quadtree cell (taking into account the distances between the nodes) and we multiply the result by the area of the corresponding cell, to obtain a measure of the gradient intensity in each cell:

$$\text{Grad. int. cell}_k = \text{cell area}$$

$$\cdot \frac{1}{6} \frac{\sum_{i=1}^3 \sum_{j=i+1}^4 |\text{Grad}_i - \text{Grad}_j|}{l_{ij}}, \quad (7)$$

where

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (8)$$

(3) Then we obtain the maximum (max) and minimum (min) values of the cell intensities for the entire domain.

(4) Cells with intensity values greater than α ($0.75 \text{ max intensity} + 0.25 \text{ min intensity}$) are selected to be refined, where

$$\begin{aligned} \alpha_{i+1} &= \alpha_i + \beta \times i; \quad (i = 1, \dots, \text{nas}); \\ \alpha_0 &= 1 \end{aligned} \quad (9)$$

with $\beta \in [0.05, 0.2]$ and $\text{nas} = \text{number of adaptation stages}$. Parameter β is related to the number of adaptation stages. Note that when using (9) α_i increases slowly in each refinement step. The quadrilateral cells to be refined are divided into four new cells. A new node is introduced in the midpoint of each nodeless corner of the new cell, respectively (see Figures 1(a) and 1(b), e.g.). The successive clouds of nodes obtained in the refinement process present a smooth nodal density transition.

(5) In the refinement process the algorithm stops when new cells to be refined are not detected. Also the relative error (%) calculated between two successive adaptation processes (10) can be taken as a criterium to stop the algorithm:

$$\frac{\sqrt{\sum_{i=1}^N (u_i^{j+1} - u_i^j)^2 / N}}{|u_{\max}^{j+1}|} \times 100, \quad (10)$$

where u_i^j is the GFDM solution in node i at the adaptive step j , u_{\max}^{j+1} is the maximum of the values in the cloud of nodes at the adaptive step $j + 1$, and N is the total number of interior nodes of the domain at the adaptive step j .

By using this adaptive algorithm in the GFDM, it is possible to decrease the error in the domain by selectively increasing the number of nodes. A fundamental point in using the adaptive procedure is the detection and refinement of the regions with high gradients; to illustrate the procedure some examples have been solved in cases with high gradients in solution.

5. Numerical Results

In this section the previously described h-adaptive method is used to solve various second-order partial differential equations, with the objective of checking the quality of the algorithm. The efficiency of the algorithm has been illustrated analyzing the reduction in the error value of the second-order PDEs with constant or variable coefficients as nodes are added.

The examples have been chosen so that high gradients are present. The potential weighting function used is

$$w(h_i, k_i) = \frac{1}{(\sqrt{h_i^2 + k_i^2})^3}. \quad (11)$$

Parameter $\beta = 0.05$ has been used in Examples 1–4 and $\beta = 0.2$ in Example 5.

As the idea is to analyze the efficiency of the algorithm, it shows that the high gradient regions are well detected and that the adaptive algorithm then adds new nodes selectively, if necessary.

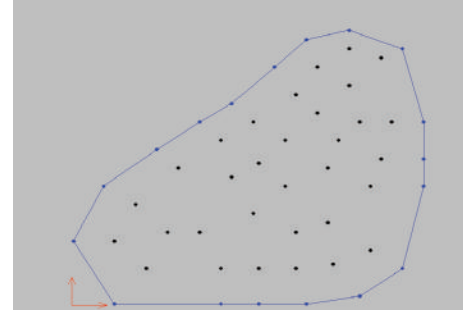


FIGURE 2: Initial cloud of 50 nodes.

The global exact error is evaluated using the following global error formula:

$$\frac{\sqrt{\sum_{i=1}^N (u_i^{\text{ex}} - u_i^j)^2 / N}}{|u_{\max}^{\text{ex}}|} \times 100, \quad (12)$$

where u_i^j is the GFDM solution in node i at the adaptive step j , u_i^{ex} is the exact value of the solution in the i , u_{\max}^{ex} is the maximum value of the exact values in the cloud of nodes considered, and N is the total number of interior nodes of the domain considered.

The following examples show how the h-adaptive method is applied several times, progressively adding new nodes. The nodes added in each step serve as a reference for the following step, in order to obtain more balanced clouds of points at the end of the process. All the examples correspond to different cases with high gradients corresponding to partial differential equations frequently encountered in engineering problems.

5.1. Example 1. For application to solve Poisson equation (13), with Dirichlet boundary conditions being the exact solution (14),

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = g(x, y), \quad (13)$$

$$U(x, y) = \frac{1}{xy} \quad (14)$$

$g(x, y)$ can be calculated by substituting exact solution (14) in the partial differential equation (13).

The domain Ω is given in Figure 2 (initial cloud with 50 nodes (32 interior nodes)). After seven refinement stages the final cloud in Figure 3 has 276 nodes (210 interior nodes). The limit used to stop the refinement steps using (10) is 1.5%.

In Figure 4 we can see the successive % global error calculated using the formula in (12) versus the number of interior nodes.

The error displayed in the logarithmic scale is shown in Figure 5.

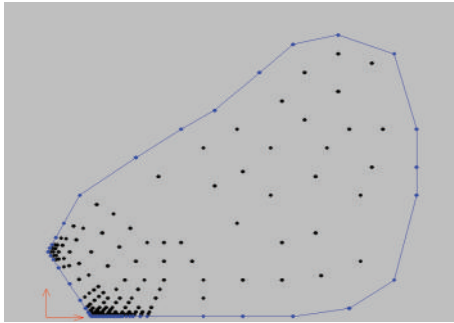


FIGURE 3: Final cloud of 276 nodes.

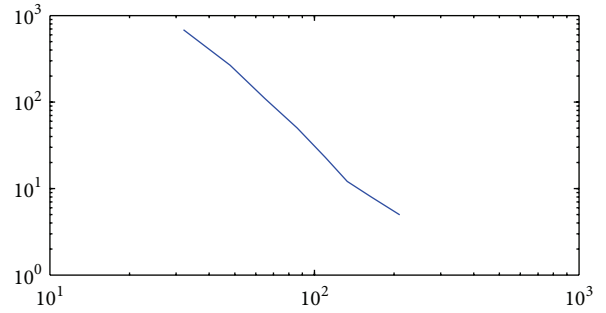


FIGURE 5: log(% global error) versus log(number of interior nodes).

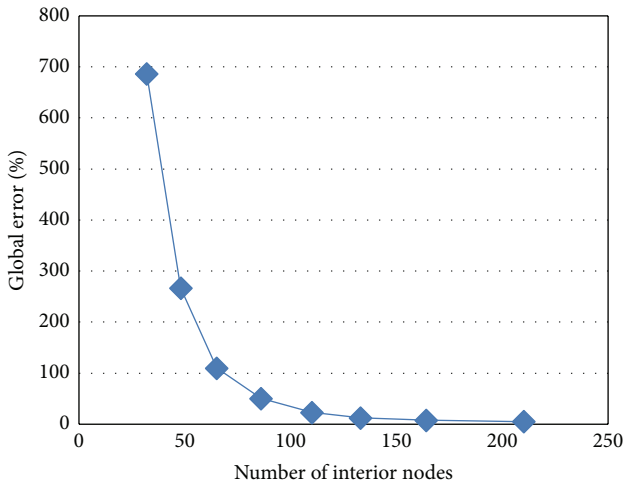


FIGURE 4: % global error versus the number of interior nodes.

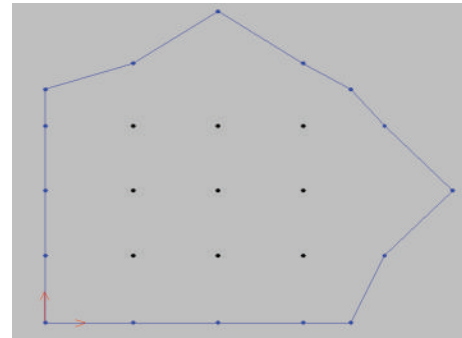


FIGURE 6: Initial cloud of 25 nodes.

5.2. Example 2. The adaptive GFDM is tested in the case of the following partial differential equation:

$$\frac{\partial^2 U}{\partial x^2} - 2 \frac{\partial^2 U}{\partial x \partial y} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial U}{\partial x} - \frac{\partial U}{\partial y} = 0 \quad (15)$$

with Dirichlet boundary conditions.

The exact solution is given by

$$U(x, y) = \frac{1}{x + y}. \quad (16)$$

The domain is defined by Figure 6 (initial cloud with 25 nodes (9 interior nodes)). Note that in (16) there is a singular point in (0, 0), not included in the domain. After eight refinement stages the final cloud in Figure 7 has 65 nodes (33 interior nodes). The limit used to stop the refinement steps using (10) has been 0.5%.

We can see in Figure 7 how the region with high gradient is detected and refined.

In Figure 8 we can see the successive global errors calculated using formula in (12). The analytical solution corresponds to the case of a function with high gradients in the function and the derivatives.

The error displayed in the logarithmic scale is shown in Figure 9.

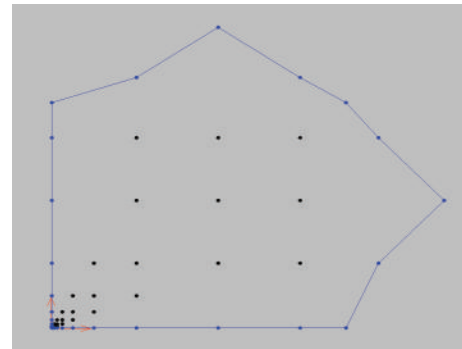


FIGURE 7: Final cloud of 65 nodes.

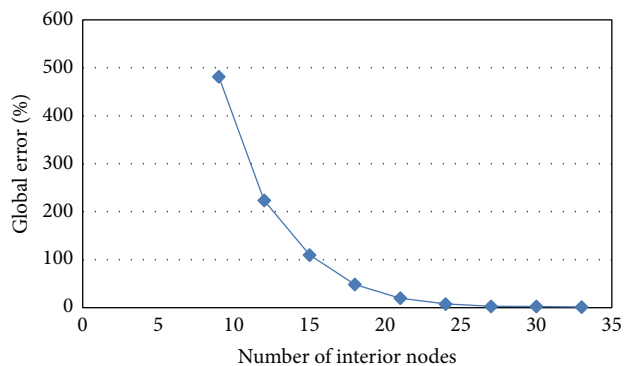


FIGURE 8: % global error versus the number of interior nodes.

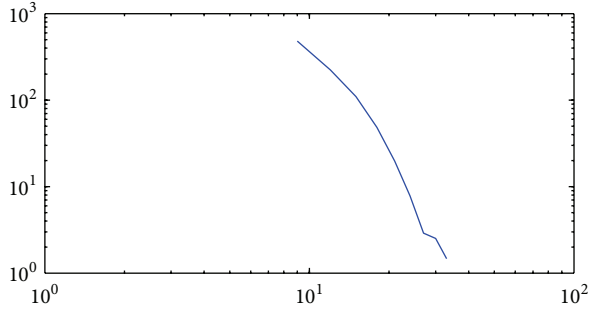


FIGURE 9: log (% global error) versus log (number of interior nodes).

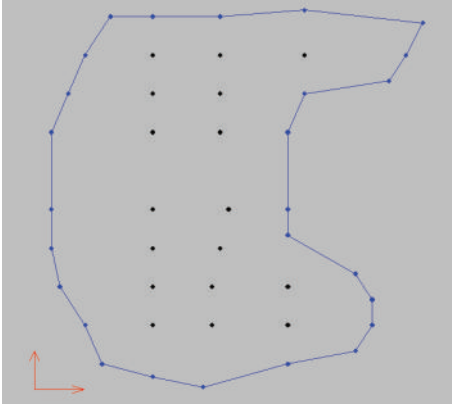


FIGURE 10: Initial cloud of 43 nodes.

5.3. *Example 3.* The adaptive GFDM is tested with the following partial differential equation with variable coefficients:

$$x \frac{\partial^2 U}{\partial x^2} + y \frac{\partial^2 U}{\partial y^2} = g(x, y) \quad (17)$$

with Dirichlet boundary conditions.

The exact solution is given by (16) and $g(x, y)$ can be calculated by substituting the exact solution (16) in the partial differential equation (17).

The domain Ω is given in Figure 10 (initial cloud with 43 nodes (17 interior nodes)). Then, after six adaptation stages the refined final cloud with 137 nodes (88 interior nodes) has been obtained; see Figure 11. The limit used to stop the refinement steps using (10) has been 0.75%.

In Figure 12 we can see the successive global errors calculated using formula in (12). We can see how the region with high gradient is detected and refined.

The error displayed in the logarithmic scale is shown in Figure 13.

5.4. *Example 4.* The adaptive GFDM is tested with the following solution of the Poisson equation (18) with Dirichlet boundary conditions.

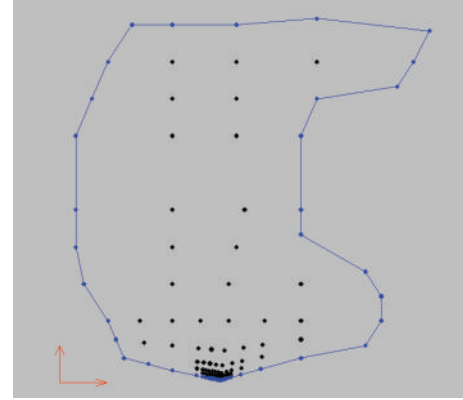


FIGURE 11: Final cloud of 137 nodes.

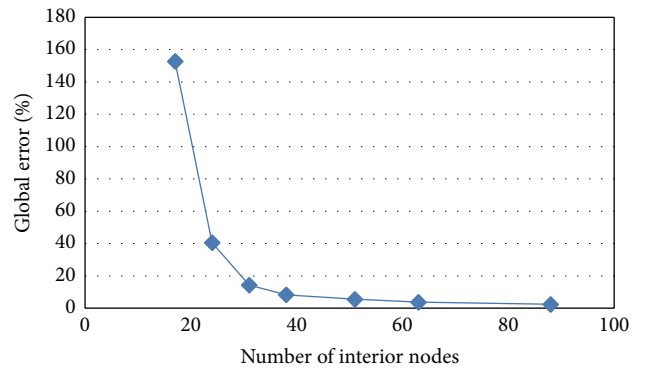


FIGURE 12: % global error versus the number of interior nodes.

The exact solution is

$$U(x, y) = \frac{1}{2000} x^2 (1 - x^2) (e^{10x^2} - 1) y^2 (1 - y^2) \cdot (e^{10y^2} - 1) \quad (18)$$

and $g(x, y)$ can be calculated by substituting the exact solution (18) in the partial differential equation (13).

The domain Ω is given in Figure 14 (initial cloud with 55 nodes (35 interior nodes)). Then, after three adaptation stages, the refined final cloud with 82 nodes (56 interior nodes) is obtained; see Figure 15. The limit used to stop the refinement steps using (10) has been 3.5%.

We can see in Figure 15 how the region with high gradients is detected and refined in three adaptation steps.

In Figure 16 we can see the successive global errors calculated using formula in (12). The analytical solution corresponds to the case of a function with high gradients.

The error displayed in the logarithmic scale is shown in Figure 17.

5.5. *Example 5.* For application to solve a problem taken from the engineering context, a torsion problem defined by the PDE,

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = -4 \quad (19)$$

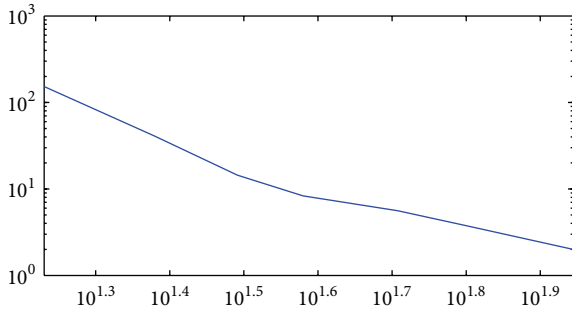


FIGURE 13: log (% global error) versus log (number of interior nodes).

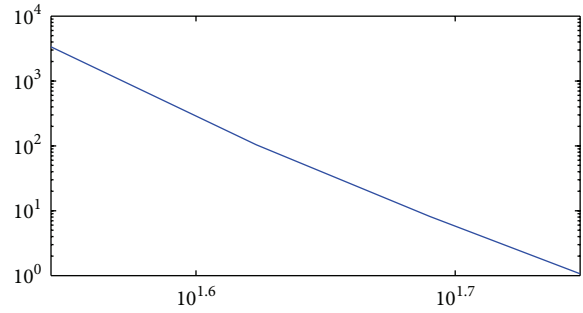


FIGURE 17: log (% global error) versus log (number of interior nodes).

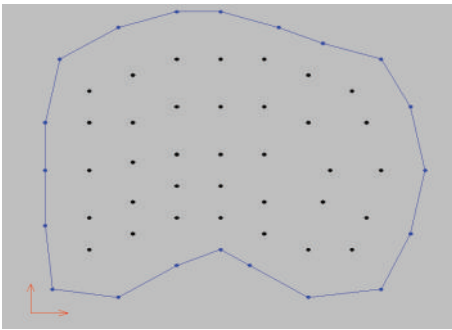


FIGURE 14: Initial cloud of 55 nodes.

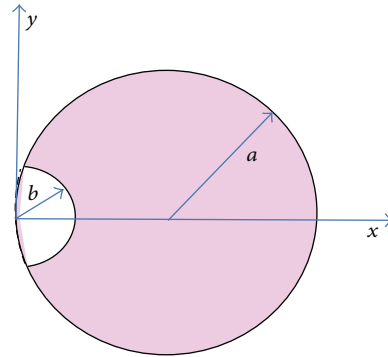


FIGURE 18: Circular shaft with a circular keyway.

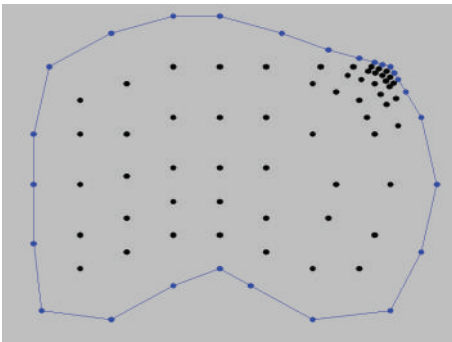


FIGURE 15: Final cloud of 82 nodes.

to calculate the Prandtl stress function Φ in the case of a circular shaft with a circular keyway, see Figure 18.

The exact solution of this problem as given by Saad [30] is

$$\Phi = (b^2 - r^2) \left(1 - \frac{2a \cos \theta}{r} \right). \quad (20)$$

The domain Ω with $a = 12, b = 4$ is shown in Figure 19 (initial cloud with 86 nodes (60 interior nodes)). After four refinement stages the final cloud of Figure 20 has 1481 nodes (1297 interior nodes). The limit used to stop the refinement steps using (10) is 0.075%.

This case is different from the other four above mentioned since the singular strength is lower but it is more widespread in the domain. β parameter has been taken as 0.2.

In Figure 21 we can see the successive % global error calculated using the formula in (12) versus the number of interior nodes.

The error displayed in the logarithmic scale is shown in Figure 22.

All the examples have been solved using a low initial number of nodes. The h-adaptive method is applied several times progressively adding new nodes, so a smooth transition area is created between zones with different nodal density. As has been shown, the numerical results obtained (see Figures 5, 9, 13, 17, and 22) show a good convergence in a few refinement steps.

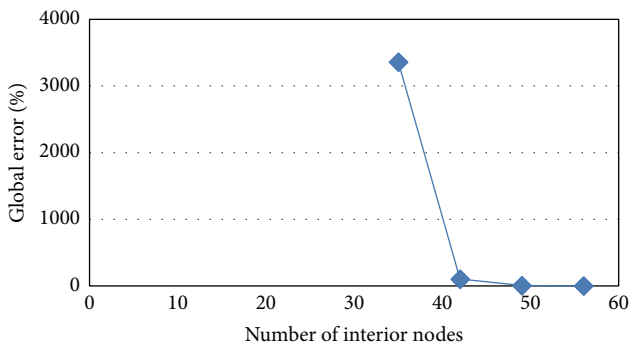


FIGURE 16: % global error versus the number of interior nodes.

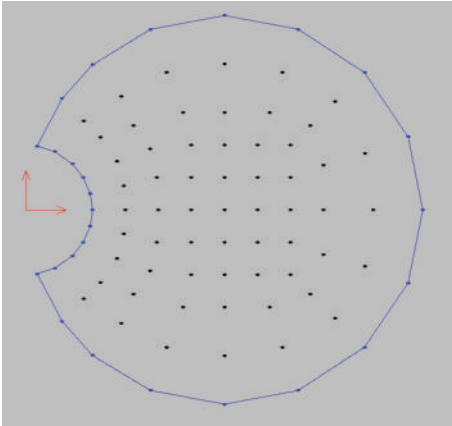


FIGURE 19: Initial cloud of 86 nodes.

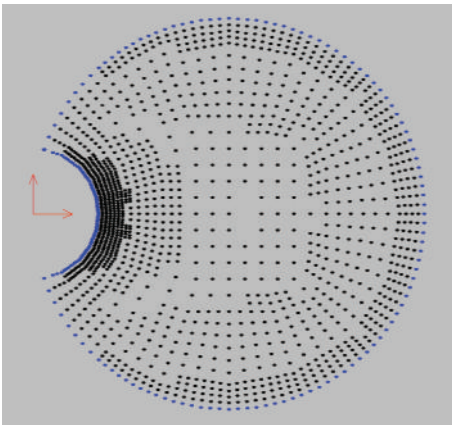


FIGURE 20: Final cloud of 1481 nodes

6. Conclusions

The use of the GFDM using irregular clouds of points is an interesting way of solving a partial differential equation. This paper shows how a fully automatic adaptive algorithm can be used with the GFDM. The h-adaptive algorithm consists of selectively adding nodes around the areas where the greater errors are located, using a quadtree method to refine the successive clouds of nodes. The algorithm efficiency has been checked analysing the reduction in the solution error value as well as the situation of the added nodes for different partial differential equations, defined in several domains including the cases of a PDE with variable coefficients and a torsion problem.

The computer program developed automatically gives the global estimated error between two successive refinement steps, the number of quadrilaterals with a gradient intensity over the limit, and the number of nodes to be added in each adaptive step. It is then possible to adjust the gradient intensities in the new adaptive steps in order to decrease the error.

Several cases of different second-order partial differential equations, with high gradients in their solutions, have been solved. This shows how the algorithm can detect the high

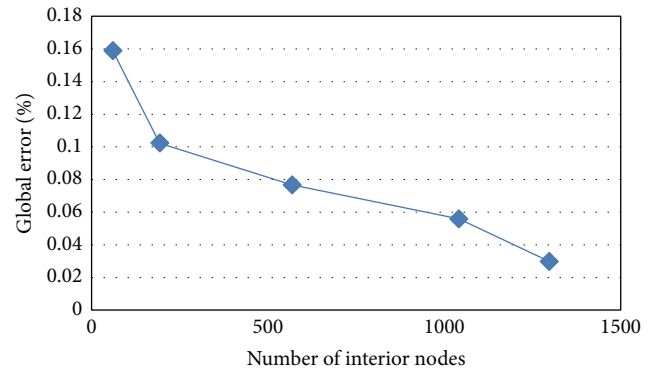


FIGURE 21: % global error versus the number of interior nodes.

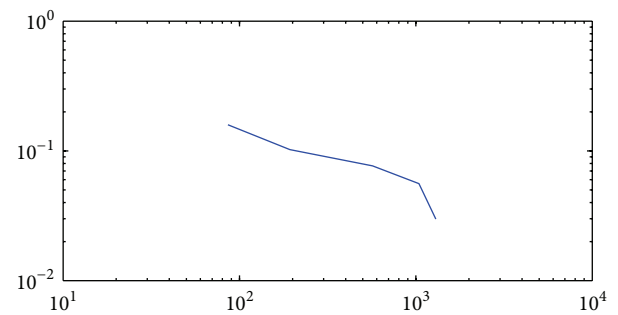


FIGURE 22: log (% global error) versus log (number of interior nodes).

gradient regions and refine them to decrease the error; thus a very good improvement of the accuracy can be obtained by adding a few nodes in each refinement step. All of this can improve the efficiency of the GFDM in future applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors acknowledge the support of the Escuela Técnica Superior de Ingenieros Industriales (UNED) of Spain, Project 2014-IFC02.

References

- [1] G. R. Liu, *Meshfree Methods*, CRC Press, 2003.
- [2] S. Li and W. K. Liu, *Meshfree Particle Methods*, Springer, Berlin, Germany, 2004.
- [3] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot, "Meshless methods: a review and computer implementation aspects," *Mathematics and Computers in Simulation*, vol. 79, no. 3, pp. 763–813, 2008.
- [4] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, no. 5, pp. 307–318, 1992.

- [5] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.
- [6] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158, 1981.
- [7] I. Babuška and J. M. Melenk, "The partition of unity method," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 4, pp. 727–758, 1997.
- [8] A. Duarte and J. T. Oden, "H-p clouds—an h-p meshless method," *Numerical Methods for Partial Differential Equations*, vol. 12, no. 6, pp. 673–705, 1996.
- [9] N. Sukumar, B. Moran, and T. Belytschko, "The natural element method in solid mechanics," *International Journal for Numerical Methods in Engineering*, vol. 43, no. 5, pp. 839–887, 1998.
- [10] E. Cueto, N. Sukumar, B. Calvo, M. A. Martínez, J. Cegoñino, and M. Doblaré, "Overview and recent advances in natural neighbour galerkin methods," *Archives of Computational Methods in Engineering*, vol. 10, no. 4, pp. 307–384, 2003.
- [11] S. N. Atluri and T. Zhu, "A new meshless local Petrov-Galerkin (MLPG) approach in modelling and simulation," *Computer Modeling and Simulation in Engineering*, vol. 3, pp. 187–196, 1998.
- [12] S. N. Atluri and T. Zhu, "A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics," *Computational Mechanics*, vol. 22, no. 2, pp. 117–127, 1998.
- [13] J. J. Monaghan, "Why particle methods work," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 4, pp. 422–433, 1982.
- [14] J. J. Monaghan, "An introduction to SPH," *Computer Physics Communications*, vol. 48, no. 1, pp. 89–96, 1988.
- [15] W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko, "Reproducing kernel particle methods for structural dynamics," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 10, pp. 1655–1679, 1995.
- [16] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor, "A finite point method in computational mechanics. Applications to convective transport and fluid flow," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 22, pp. 3839–3866, 1996.
- [17] P. S. Jensen, "Finite difference techniques for variable grids," *Computers and Structures*, vol. 2, no. 1-2, pp. 17–29, 1972.
- [18] N. Perrone and R. Kao, "A general finite difference method for arbitrary meshes," *Computers & Structures*, vol. 5, no. 1, pp. 45–58, 1975.
- [19] T. Liszka and J. Orkisz, "The finite difference method at arbitrary irregular grids and its application in applied mechanics," *Computers & Structures*, vol. 11, no. 1-2, pp. 83–95, 1980.
- [20] J. Orkisz, "Finite difference method (part, III)," in *Handbook of Computational Solid Mechanics*, M. Kleiber, Ed., Springer, Berlin, Germany, 1998.
- [21] J. J. Benito, F. Ureña, and L. Gavete, "Influence of several factors in the generalized finite difference method," *Applied Mathematical Modelling*, vol. 25, no. 12, pp. 1039–1053, 2001.
- [22] L. Gavete, M. L. Gavete, and J. J. Benito, "Improvements of generalized finite difference method and comparison with other meshless method," *Applied Mathematical Modelling*, vol. 27, no. 10, pp. 831–847, 2003.
- [23] J. J. Benito, F. Ureña, L. Gavete, and R. Alvarez, "An h-adaptive method in the generalized finite differences," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 5-6, pp. 735–759, 2003.
- [24] J. J. Benito, F. Ureña, L. Gavete, and B. Alonso, "A posteriori error estimator and indicator in generalized finite differences. Application to improve the approximated solution of elliptic PDEs," *International Journal of Computer Mathematics*, vol. 85, no. 3-4, pp. 359–370, 2008.
- [25] J. J. Benito, F. Ureña, L. Gavete, and B. Alonso, "Application to improve the approximated solution of pdes," *Computer Modeling in Engineering & Sciences*, vol. 38, no. 1, pp. 39–58, 2008.
- [26] I. Jaworska and J. Orkisz, "Higher order multipoint method—from Collatz to meshless FDM," *Engineering Analysis with Boundary Elements*, vol. 50, pp. 341–351, 2015.
- [27] M. Syczewski and R. Tribillo, "Singularities of sets used in the mesh method," *Computer & Structures*, vol. 14, no. 5-6, pp. 509–511, 1981.
- [28] E. Stein and W. Rust, "Mesh adaptations for linear 2D finite-element discretizations in structural mechanics, especially in thin shell analysis," *Journal of Computational and Applied Mathematics*, vol. 36, no. 1, pp. 107–129, 1991.
- [29] Y. Luo and U. Häussler-Combe, "A gradient-based adaptation procedure and its implementation in the element-free Galerkin method," *International Journal for Numerical Methods in Engineering*, vol. 56, no. 9, pp. 1335–1354, 2003.
- [30] M. H. Saad, *Elasticity*, Elsevier Butterworth-Heinemann, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

