Copyright © 2008 Tech Science Press CMES, vol.38, no.1, pp.39-58, 2008

Application of the Generalized Finite Difference Method to improve the approximated solution of pdes

J.J. Benito[1], F. Ureňa[2], L. Gavete[3] and B. Alonso[3]

Abstract: One of the most universal and effective methods, in wide use today, for solving equations of mathematical physics approximately is the finite difference method (FDM). The Generalized finite difference method (GFDM) is evolved fron classical (FDM), which can be applied over general or irregular clouds of points.
This paper starts by showing the GFDM. In this paper, this meshless method is used for solving second-order partial (pde's) with constant coefficients in any type of domain. The method gives the values of derivatives in the nodes using the direct application of the formulae in differences obtained.
The following points describe an a posteriori error estimator. This serves as a starting point for an h-adaptive method to improve the solution of pde's by selectively adding nodes to the domain.

1 Introduction

During recent years, meshless methods have emerged as a class of effective numerical methods which are capable of avoiding the difficulties encountered in conventional computational mesh based methods. Considerable research in computational mechanics has been devoted to the development of meshless methods (see Atluri and Shen (2002), Liu (2003), Li and Liu (2004)). In these methods, the domain of interest is discretized by a scattered set of points. One of the earliest developments in meshless methods was the SPH method. The foundation of the SPH method is the kernel estimate introduced by Monaghan (1982) and Monaghan (1988). In this method, partial differential equations (pde's), such as conservation laws, are transformed into integral equations, and the kernel estimate then provides the approximation to estimate field variables at discrete points. Liu, Jun, Li, Adee and Belytschko (1995) proposed a different kind of "gridless" multiple scale method based on reproducing kernel and wavelet analysis (RPKM method), to improve the

[1] UNED, Madrid, SPAIN.
[2] UCLM, CR, Castilla-La Mancha, SPAIN.
[3] UPM, Madrid, SPAIN.

accuracy of the SPH method for finite domain problems. In this method, the kernel function is modified by introducing a correction function to meet the reproducing conditions.

The diffuse element method developed by Nayroles, Touzot and Villon (1992) was the first meshless method based on the Galerkin method. They proposed a Diffuse Element Method that employs moving least-squares interpolants in conjunction with the Galerkin method to provide a mesh-free computational formulation. Belytschko, Lu and Gu (1994) developed an alternative implementation using moving least squares approximation as defined by Lancaster and Salkauskas (1981). They called their approach the Element Free Galerkin (EFG) method. Several other meshless methods as Partition of Unity Finite Element method (PUFEM) by Babuska and Melenk (1997); h-p cloud by Duarte and Oden (1996); Natural element method (NEM) by Sukumar, Moran, and Belytschko (1998) have also been reported in the literature. These methods employ the Galerkin procedure by using a shadow mesh of elements for integrating the weak formulation.

Other possibility to integrate the weak formulation is to use the meshless Petrov-Galerkin method (MLPG) as reported by Atluri and Zhu (1998), Atluri and Zhu (1998). The MLPG is a truly meshless method because all integrations are performed over overlapping and regularly shaped subdomains. A mixed approach was introduced to improve the MLPG method using finite difference method by Atluri, Liu and Han (2006), Atluri, Liu and Han (2006).

Another important path in the evolution of meshless methods has been the development of the Generalized Finite Difference Method (GFDM), also called meshless finite difference method. The bases of the GFD were published in the early seventies. Jensen (1972) was the first to introduce fully arbitrary mesh. He considered Taylor‘s series expansions interpolated on six-node stars in order to derive the finite difference (FD) formulae approximating derivatives of up to the second order. Perrone and Kao (1975) suggested that additional nodes in the six-point scheme should be considered and an averaging process for the generalization of finite difference coefficients applied. The idea of using an eight node star and weighting functions to obtain finite difference formulae for irregular meshes, was first put forward by Liszka and Orkisz (1980) using moving least squares (MLS) interpolation. and an advanced version of the GFDM was given by Orkisz (1998). Benito, Urena and Gavete (2001) reported that the solution of the generalized finite difference method depends on the number of nodes in the cloud, the relative coordinates of the nodes with respect to the star node, and on the weight function employed.

GFD methods have been used in many engineering applications and in papers where irregular geometries and free-moving boundaries are involved by Tang, Chen, Yang Kobayashi and Ku (2002), Tang, Yang Kobayashi and Ku (2001), Yang, Tang, Yuan, Kerwin, Liu Canton, Hatsukami and Atluri (2008).

An h-adaptive method in GFDM is described in Benito, Urena, Gavete and Alvarez (2003) and Urena, Benito, Gavete and Alvarez (2005). Gavete and Benito (2003) reported improvements of GFDM and comparison with other meshless method.

This paper describes how the GFDM can be applied for the improve solution of different partial differential equations (pde's).

The paper is organized as follows. Section 1 is an introduction. Section 2 describes the GFDM obtaining the explicit formulae. Section 3 describes an adaptive algorithm and shows the a posteriori error indicator in the GFDM. In Section 4 some numerical results are included to illustrate the efficiency of the h-adaptive algorithm. Finally, in Section 5 some conclusions are given.

2 Generalized finite difference approximation

Let us assume a problem governed by the following second-order pde:

$$L_2[U] = 0 \quad in \quad \Omega \tag{1}$$

with boundary conditions

$$L_1[U] = 0 \quad in \quad \Gamma \tag{2}$$

where U is a function at least twice differentiable in $\Omega \subset R^2$ with boundary Γ. L_2 and L_1 are linear partial differential second and first order, respectively.

On defining the composition central node with a set of N points surrounding it (henceforth referred as nodes), the star then refers to the group of established nodes in relation to a central node. Each node in the domain have an associated star assigned.

If U_0 is the value of the function at the central node of the star, with coordinates (x_0, y_0) and U_i the value of the function at the rest of nodes, of coordinates (x_i, y_i) with $i = 1, \cdots, N$, then, according to the Taylor series expansion we know that an approximation , u_i, of the second order for the U_i, is:

$$u_i = u_0 + h_i \frac{\partial u_0}{\partial x} + k_i \frac{\partial u_0}{\partial y} + \frac{1}{2}(h_i^2 \frac{\partial^2 u_0}{\partial x^2} + k_i^2 \frac{\partial^2 u_0}{\partial y^2} + 2h_i k_i \frac{\partial^2 u_0}{\partial x \partial y}) \tag{3}$$

where $h_i = x_i - x_0; k_i = y_i - y_0$.

Minimizing the function

$$B[u] = \sum_{i=1}^{N} [(u_0 - u_i + h_i \frac{\partial u_0}{\partial x} + k_i \frac{\partial u_0}{\partial y} + \tfrac{1}{2}(h_i^2 \tfrac{\partial^2 u_0}{\partial x^2} + k_i^2 \tfrac{\partial^2 u_0}{\partial y^2} + 2h_i k_i \tfrac{\partial^2 u_0}{\partial x \partial y}))w(h_i, k_i)]^2 \tag{4}$$

with respect the partial derivatives, and using $w(h_i,k_i)$ which is a weighting function, a linear equations system is obtained

$$AD_u = b \tag{5}$$

where

$$D_u = \left\{ \begin{array}{ccccc} \dfrac{\partial u_0}{\partial x} & \dfrac{\partial u_0}{\partial y} & \dfrac{\partial^2 u_0}{\partial x^2} & \dfrac{\partial^2 u_0}{\partial y^2} & \dfrac{\partial^2 u_0}{\partial x \partial y} \end{array} \right\}^T \tag{6}$$

$$b = \left\{ \begin{array}{c} \sum\limits_{i=1}^{N}(-u_0+u_i)h_i w^2 \\ \sum\limits_{i=1}^{N}(-u_0+u_i)k_i w^2 \\ \sum\limits_{i=1}^{N}(-u_0+u_i)\dfrac{h_i^2}{2}w^2 \\ \sum\limits_{i=1}^{N}(-u_0+u_i)\dfrac{k_i^2}{2}w^2 \\ \sum\limits_{i=1}^{N}(-u_0+u_i)h_i k_i w^2 \end{array} \right\} \tag{7}$$

$$A = \begin{pmatrix} \sum\limits_{i=1}^{N} h_i^2 w^2 & \sum\limits_{i=1}^{N} h_i k_i w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i^3}{2} w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i k_i^2}{2} w^2 & \sum\limits_{i=1}^{N} h_i^2 k_i w^2 \\ & \sum\limits_{i=1}^{N} k_i^2 w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i^2 k_i}{2} w^2 & \sum\limits_{i=1}^{N} \dfrac{k_i^3}{2} w^2 & \sum\limits_{i=1}^{N} h_i k_i^2 w^2 \\ & & \sum\limits_{i=1}^{N} \dfrac{h_i^4}{4} w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i^2 k_i^2}{4} w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i^3 k_1}{2} w^2 \\ & SYM & & \sum\limits_{i=1}^{N} \dfrac{k_i^4}{4} w^2 & \sum\limits_{i=1}^{N} \dfrac{h_i k_i^3}{2} w^2 \\ & & & & \sum\limits_{i=1}^{N} h_i^2 k_i^2 w^2 \end{pmatrix} \tag{8}$$

The explicit expressions of the vector D_u depend on the number of nodes, the selection and placing of the nodes (star) and the weight function.

From the previously obtained matrix equation Eq. 5 and as the matrix of coefficients A_p is symmetrical, it is then possible to use the Cholesky method in order to solve the same. The aim is to obtain the decomposition of A_p in the product of an upper and a lower triangular matrices QQ^T. Then, the equation Eq. 5 can be written as:

$$QQ^T D_u = b \tag{9}$$

This is then solved in two stages, the first of which is:

$$Q^T D_u = Y \tag{10}$$

Which provides the vector D_u, after solving:

$$QY = b \tag{11}$$

On solving the system Eq. 11 in descending order, the Y values are obtained. Once the vector Y has been established, it is then easy to solve the system Eq. 10 and to obtain the following difference formulae:

$$\begin{cases} Y(k) = -u_0 \sum_{i=1}^{5} M_{ki} c_i + \sum_{j=1}^{N} u_j (\sum_{i=1}^{5} M_{ki} d_{ji}), k = 1, \cdots, 5 \\ D_u(k) = \frac{1}{q_{kk}} (Y(k) - \sum_{i=1}^{5-k} q_{(k+1)k} D_u(k+i)), k = 1, \cdots, 5 \end{cases} \tag{12}$$

where

$$M_{ij} = \begin{cases} (-1)^{1-\delta_{ij}} \frac{1}{q_{ij}} \sum_{k=j}^{i-1} q(ik) M(kj) & for \quad j < i \\ \frac{1}{q_{ij}} & for \quad j = i \\ 0 & for \quad j > i \end{cases}$$

with $i, j = 1, \cdots, 5$, and where q_{ij} are elements of Q, δ_{ij} the Kronecker delta function,

$$c_i = \sum_{j=1}^{N} d_{ji}$$

$$d_{j1} = h_j w^2; \quad d_{j2} = k_j w^2; \quad d_{j3} = \frac{h_j^2}{2} w^2$$

$$d_{j4} = \frac{k_j^2}{2} w^2; \quad d_{j5} = h_j k_j w^2$$

If in a node of the domain the partial derivatives of the equations are substituted in Eq. 1 and Eq. 2 by the explicit differences formulae Eq. 11, the star equation is obtained as

$$u_0 = \sum_{i=1}^{N} m_i u_i, \quad with \quad \sum_{i=1}^{N} m_i = 1 \tag{13}$$

If this process is carried out for each node of the domain a linear equations system is obtained, where the unknowns are the values u_i. On solving this system, the approximated values of the function in the nodes of the domain are obtained and the partial derivatives may easily be calculated from the aforementioned Eq. 12.

From the above it may be seen that the star equation depends on the following factors:

- *The number of nodes of the star (N).*
  The results improve as the number of nodes in the star increases, although over a certain number of nodes ($\geq$ 8 for 2-D) this improvement does not compensate the effort of the calculation process involved.

- *The relative coordinates of the nodes of the star with regards to the central node (h_i, k_i).*
  When selecting the nodes of the star, Jensen (1972) only considered the distance of the nodes to the central node, and selected those closest to the same. This method shall subsequently referred to as the distance criterion. This criterion may produce distorted stars with an uneven distribution of nodes around the central node which is, subsequently, reflected by more imprecise results (see Benito, Urena and Gavete (2001)).
  Perrone and Kao (1975) suggested a second method, which may be referred to as the eight segment criterion, consisting of the selection of the nearest node in each octant of a system of cartesian axes around the central node of the star. Given the irregular density of the node this method produces further calculation error. A viable alternative would be to correct this last method in terms of distance, assuring that none of the nodes are set at greater distances to those indicated, and thereby correcting the effect of node density irregularity.
  A third method, proposed by Liszka and Orkisz (1980), Orkisz (1998), which denominated the four quadrants criterion, consists of the selection of the two nearest nodes per quadrant. The mentioned quadrant is established in the same manner as that indicated above by cartesian axes around the central node of the star. This method corrects the problems of the octant method.
  The results improve when using the four quadrants criterion and the distance criterion of selection of nodes of the star (see Benito, Urena and Gavete (2001)).

- *The weighting function w.*
  The factor m_i in Eq. 13 depends on weighting function w. Then , the influence of the weight function is clear, the value of the function in a node of the domain is the weighting sum of the values of the function in the rest of

the nodes of the star (Eq. 13), with more influence of the nodes closest to the central node.

Different weighting functions can be used as potential, exponential or splines.

Potentials.

$$\omega = \frac{1}{(dist)^n}, n = 2,3,4 \tag{14}$$

Exponentials

$$\omega = exp^{-k(dist)^2}, k > 0 \tag{15}$$

Quartic Spline.

$$\omega = \begin{cases} 1-6(\frac{dist}{RP})^2+8(\frac{dist}{RP})^3-3(\frac{dist}{RP})^4 & if \quad dist \leq RP \\ 0 & if \quad dist > RP \end{cases} \tag{16}$$

Cubic Spline.

$$\omega = \begin{cases} \frac{2}{3}-4(\frac{dist}{RP})^2+4(\frac{dist}{RP})^3 & if \quad 0 \leq dist \leq \frac{RP}{2} \\ \frac{4}{3}-4(\frac{dist}{RP})+4(\frac{dist}{RP})^2-\frac{4}{3}(\frac{dist}{RP})^3 & if \quad \frac{RP}{2} \leq dist \leq RP \\ 0 & if \quad dist > RP \end{cases} \tag{17}$$

where *RP* is a parameter.

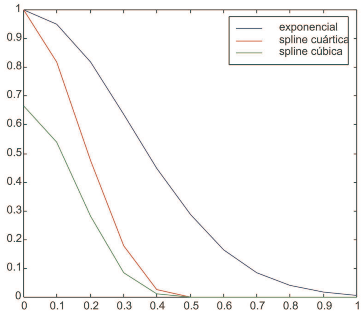

Figure 1: Exponential and Splines

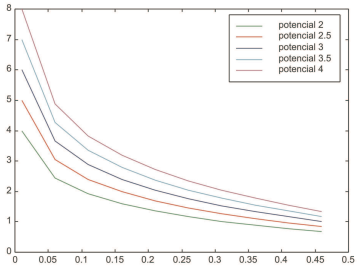

Figure 2: Potentials

The weighting functions are plotted in rectangular coordinates and there are two graphs, weighting magnitude versus distance central node for exponential and splines weighting functions (Fig. 1) and logarithm of weighting magnitude versus distance in the case of potential functions (Fig. 2).

The potential functions give more influence to the star nodes the nearer they are to the central node (greater m_i values.

3 Influence of the weighting functions.

The issue of influence of weighting function is examined in the context of two numerical examples. In both cases a Laplace equation is solved using the four quadrants criterion to select the star nodes and different weighting functions described in the previous section.

The global exact error can be calculated as

$$Global \quad exact \quad error = \frac{\sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}}}{exac_{max}} \tag{18}$$

where N is the number of nodes in the domain, $exac_{max}$ is the maximum exact value of function in the domain, e_i is the exact error in the node i.

The nodal absolute exact error is,

$$nodal \quad error = |e_i| = |sol(i) - exac(i)| \tag{19}$$

and it will be drawn by vertical lines in figures.

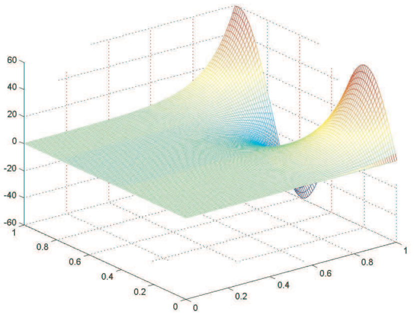

Figure 3: $U(x,y) = \frac{exp(8x)\sin(8x)}{50}$

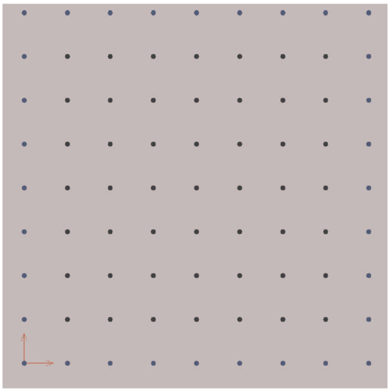

Figure 4: Cloud of 81 nodes

Example I.- The exact solution is (Fig. 3)

$$U(x,y) = \frac{exp(8x)\sin(8x)}{50} \tag{20}$$

The domain is defined by Fig. 4 (mesh with 81 nodes).

Tab. 1 shows the global exact error, Eq. 18, values. The best results are obtained using the potential function as it can be appreciated in this Tab. 1.

Table 1: Influence of the weight function (Example I).

Example	Weight function	Global error
I	Pot. $(n = 3)$	0.002281
I	Exp. $(k = 5)$	1.639
I	Exp. $(k = 0.5)$	1.741
I	Quartic $(RP = 0.1)$	1.746
I	Quartic $(RP = 0.05)$	1.746
I	Cubic $(RP = 0.5)$	2.264
I	Cubic $(RP = 0.1)$	1.746
I	$\omega = 1$	1.746

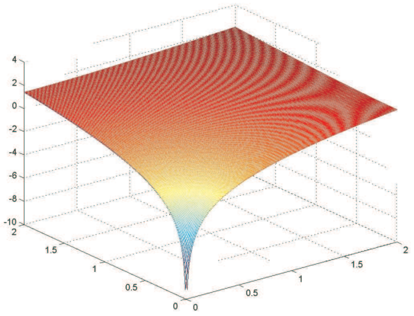

Figure 5: $U(x,y) = \log(x^2 + y^2)$

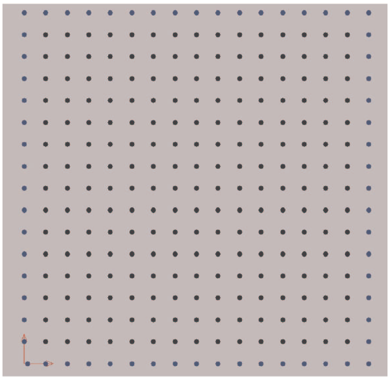

Figure 6: Cloud of 289 nodes

Example II.- The exact solution is (Fig. 5)

$$U(x,y) = \log(x^2 + y^2) \tag{21}$$

The domain is defined by Fig. 6 (mesh with 289 nodes).

Tab. 2 shows the global exact error, Eq. 18, values as in the previous case the best results are obtained when the potential weighting function is used.

4 Comparison with the standard Finite Difference Method

In order to analyse the potential of GFDM a comparison with the standard FDM is carried out in this section.

Table 2: Influence of the weight function (Example II).

Example	Weight function	Global error
II	Pot. $(n = 3)$	0.03136
II	Exp. $(k = 5)$	0.258
II	Exp. $(k = 0.5)$	0.2614
II	Quartic $(RP = 0.1)$	0.1434
II	Quartic $(RP = 0.05)$	0.2615
II	Cubic $(RP = 0.5)$	0.3648
II	Cubic $(RP = 0.1)$	0.1799
II	$\omega = 1$	0.2615

Tab. 3 and Tab. 4 show the global exact errors (Eq. 18) obtained solving the examples I and II of the previous section applying the standard Finite Difference Method (SFDM)(five point schemes) and the GFDM for stars with eight and five nodes (GFD8 and GFD5 respectively).

Table 3: Comparison with SFDM.Example I

Cloud of nodes	Methods	Global error
Regular (81)	GFD8	0.002281
Regular (81)	GFD5	1.094
Regular (81)	SFD	1.094

5 An h-adaptive method in the GFD

In this section an h-adaptive method in the GFD is shown. This adaptive method proposes adding nodes selectively in the domain to improve the approximated solution.

The a posteriori error indicator, $Ind(u_0)$, used in this paper is defined by the weighted addition of the absolute values of the difference between the fourth order and the

Table 4: Comparison with SFDM. Example II

Cloud of nodes	Methods	Global error
Regular (289)	GFD8	0.03136
Regular (289)	GFD5	0.1507
Regular (289)	SFD	0.1507

second order approximations, obtained using Taylor serial expansions

$$Ind(u_0) = \sum_{i=1}^{N} \left| m_i \left(\frac{1}{6}(h_i^3 \frac{\partial^3 u_0}{\partial x^3} + k_i^3 \frac{\partial^3 u_0}{\partial y^3} + 3h_i^2 k_i \frac{\partial^3 u_0}{\partial x^2 \partial y} + 3h_i k_i^2 \frac{\partial^3 u_0}{\partial x \partial^2 y}) \right.\right.$$
$$\left.\left. + \frac{1}{24}(h_i^4 \frac{\partial^4 u_0}{\partial x^4} + k_i^4 \frac{\partial^4 u_0}{\partial y^4} + 4h_i^3 k_i \frac{\partial^4 u_0}{\partial x^3 \partial y} + 6h_i^2 k_i^2 \frac{\partial^4 u_0}{\partial x^2 \partial y^2} + 4h_i k_i^3 \frac{\partial^4 u_0}{\partial x \partial y^3}) \right) \right| \qquad (22)$$

The 3rd and 4th order partial derivatives are calculated using the values of 1st and 2nd order derivatives and the explicit differences formulae.

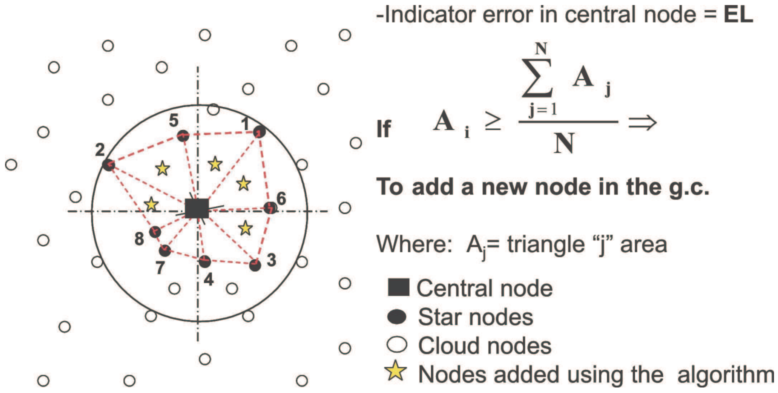
1
2
3
4
5
6
7
8
-Indicator error in central node = EL
If $A_i \geq \frac{\sum_{j=1}^{N} A_j}{N} \Rightarrow$
To add a new node in the g.c.
Where: A_j= triangle "j" area
■ Central node
● Star nodes
○ Cloud nodes
☆ Nodes added using the algorithm

Figure 7: Example of h-adaptive algorithm. I) Nodes of the star and triangles. II) New nodes

If the value given by the a posteriori error indicator in a node is greater than a fixed value (first parameter=Error limit (EL)) then new nodes are added at the centre of gravity of triangles made using this node and the other nodes of the star. If the area of a triangle is smaller than the mean area (second parameter),of the triangles of a star (eight in the example of Fig. 7), then the new node should not be added.

Fig. 7 show, an example, the application of the h-adaptive algorithm. A star whose central node is represented by a square, and the other nodes of star, eight (in this case), by circles and the triangles that can be built using the star nodes avoiding overlapping.

Fig. 7 shows the new nodes added, five in this case (because the triangles made using the central node and the nodes whose numbers are: 8,7,4, have an area smaller than the mean area).

6 Numerical results

The efficiency of the algorithm has been illustrated analyzing the reduction in the solution error value of pde's with constant coefficients as well as nodes are added. This section includes three examples of the application of the h-adaptive algorithm to improve the approximated solution of second order pde's with constant coefficients. It is interesting to see how not only the global error (Eq. 18) but also the nodal errors (Eq. 19) are significantly reduced.

The domains are irregular and the solutions are chosen so that important gradients are presented.

The potential weighting function (Eq. 14) has been used in every example.

As the idea is to analyse the efficiency of the algorithm, the clouds of nodes of these academic examples have been generated randomly. In real applications it would logical to choose the most regular position of nodes possible, considering the shape of the domain, the number of star nodes, the shape of boundary, the boundary conditions, etc. Below, the h-adaptive algorithm will add new nodes selectively if necessary.

6.1 Case elliptic

Application to solve Laplace equation, with Dirichlet boundary condition and the exact solution is Eq. 21(Fig. 5). The domain is defined by Fig. 8 (mesh with 125 nodes).

The first cloud, Fig. 8, has 125 nodes (38 nodes in the boundary) and after two adaptive steps ($EL = 0.01$ and $EL = 0.0055$) we obtain the cloud showed in Fig. 9, which has 145 nodes (41 nodes in the boundary, three nodes more that the first clouds, included in the area with greater error).

In Tab. 5 we can see a summary of the global exact error (Eq. 18) obtained using the adaptive method.

Fig. 10 and Fig. 11 show the nodal errors (Eq. 19) for the step 1 (cloud of 125 nodes) and for the last step (cloud of 145 nodes) respectively, as it is shown in

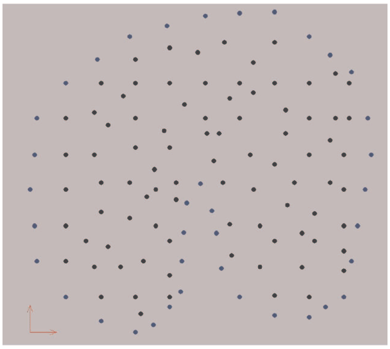

Figure 8: First cloud of 125 nodes

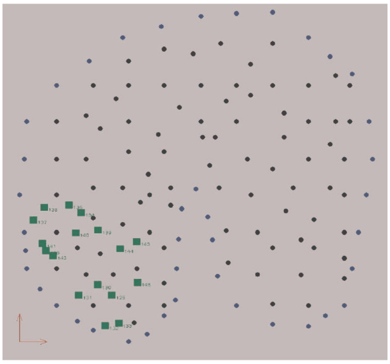

Figure 9: Last cloud of 145 nodes

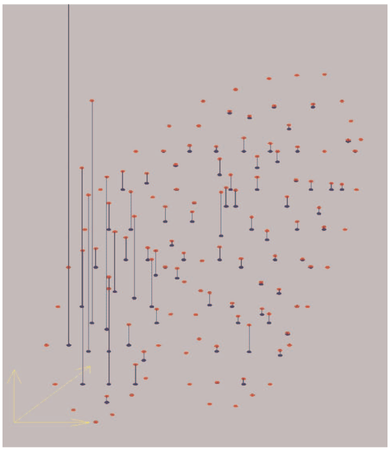

Figure 10: First cloud. Nodal errors (125 nodes)

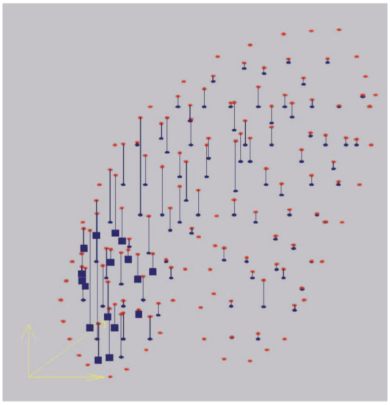

Figure 11: Last cloud. Nodal errors (145 nodes)

Fig. 11 the nodal errors decrease.

6.2 Case hyperbolic

Application to solve the equation

$$\frac{\partial^2 U}{\partial x^2} - 3\frac{\partial^2 U}{\partial y^2} = 0 \tag{23}$$

with Dirichlet boundary condition and the exact solution is (see Fig. 12)

$$U(x,y) = x^3 + xy^2 \tag{24}$$

Table 5: Adaptive Method (case 1). Global error.

Number of nodes	Error %
125	0.05665
133	0.04617
145	0.02796

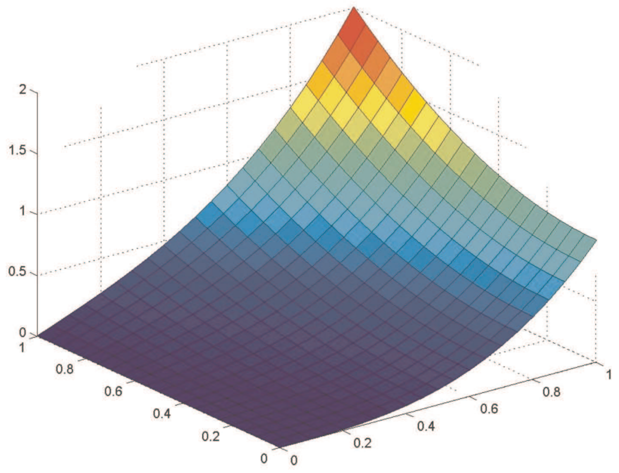

Figure 12: $U(x,y) = x^3 + xy^2$

The domain is defined by Fig. 13 (mesh with 76 nodes).

The first cloud, Fig. 13, has 76 nodes (36 nodes in the boundary) and after four steps using the adaptive algorithm we obtain the cloud showed in Fig. 14, which has 134 nodes (56 nodes in the boundary, twenty nodes more that the first clouds, included in the side with a greater error).

In Tab. 6 we can see a summary of the global errors (Eq. 18) obtained using the adaptive method.

Fig. 15 and Fig. 16 show the nodal errors (Eq. 19) for step 1 (cloud of 76 nodes) and for the last step (cloud of 134 nodes) respectively.

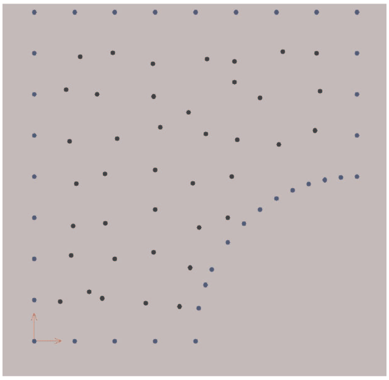

Figure 13: First cloud of 76 nodes

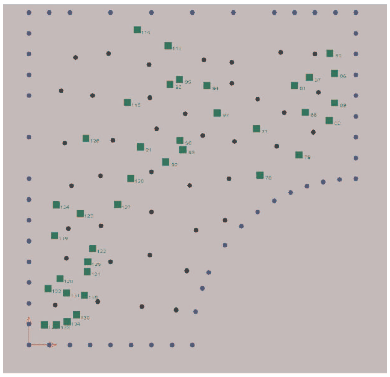

Figure 14: Last cloud of 134 nodes

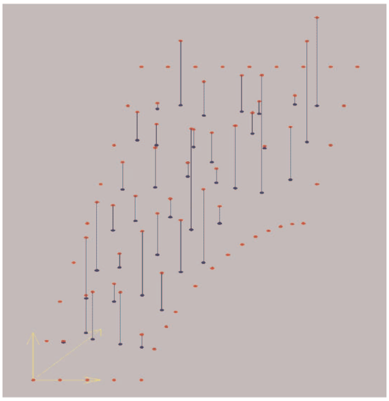

Figure 15: First cloud. Nodal errors (76 nodes)

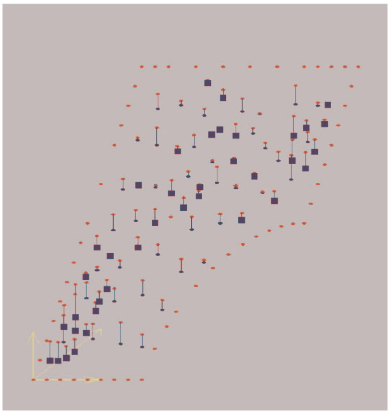

Figure 16: Last cloud. Nodal errors (134 nodes)

Table 6: Adaptive Method (case 2). Global error.

Number of nodes	Error %
76	0.2706
83	0.2197
116	0.08236
130	0.07868
134	0.07069

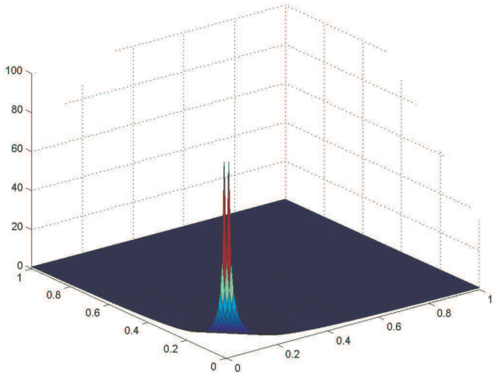

Figure 17: $U(x,y) = \frac{1}{x+y}$

6.3 Case parabolic

Application to solve the equation

$$\frac{\partial^2 U}{\partial x^2} - 2\frac{\partial^2 U}{\partial x \partial y} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial U}{\partial x} - \frac{\partial U}{\partial y} = 0 \tag{25}$$

with Dirichlet boundary condition and the exact solution is (Fig. 17)

$$U(x,y) = \frac{1}{x+y} \tag{26}$$

The domain is defined by Fig. 18 (mesh with 102 nodes)(with $0.02 \le x \le 1$).

The first cloud, Fig. 18, has 102 nodes (46 nodes in the boundary) and after three steps ($EL = 0.0086$, $EL = 0.0005$ and $EL = 0.00025$) using the adaptive algorithm we obtain the cloud showed in Fig. 19, which has 134 nodes (52 nodes in the boundary, six nodes more that the first clouds, included in the side with a greater error).

In Tab. 7 we can see a summary of the global errors ((Eq. 18)) obtained using the adaptive method.

Fig. 20 and Fig. 21 show the nodal errors ((Eq. 19)) for step 1 (cloud of 102 nodes) and for the last step (cloud of 134 nodes) respectively.

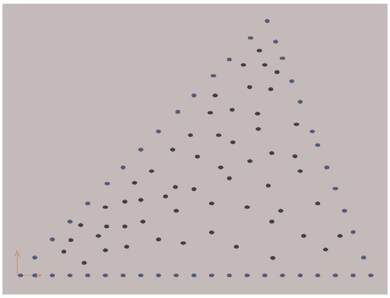

Figure 18: First cloud of 102 nodes

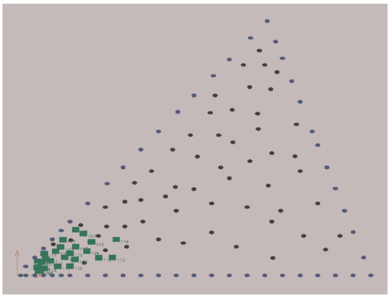

Figure 19: Last cloud of 134 nodes

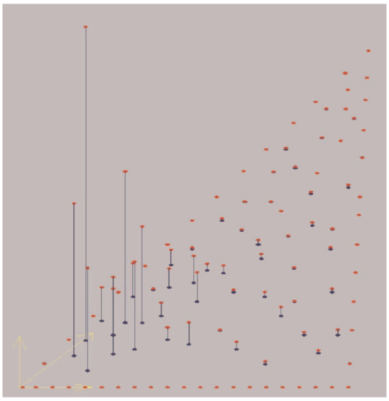

Figure 20: First cloud. Nodal errors (102 nodes)

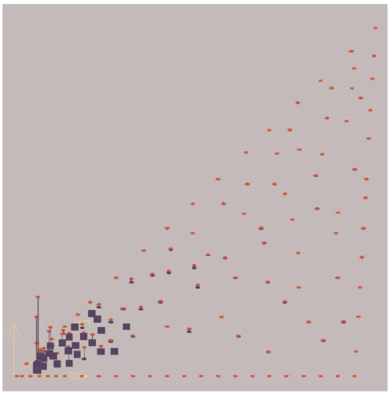

Figure 21: Last cloud. Nodal errors (134 nodes)

Table 7: Adaptive Method (case 3). Global error.

Number of nodes	Error %
102	0.01588
106	0.009414
123	0.00726
134	0.006231

7 Conclusions

The use of the generalized finite difference method using irregular clouds of points is an interesting way of solving partial differential equations. The results obtained for different equations show that the generalized finite difference method provides excellent results for the value of the function and its derivatives.

The best results are obtained using functions with a steeper slope, then weighting functions like the potential are advisable. It is more difficult to appreciate this when the solutions are smoother.

The h-adaptive algorithm efficiency has been checked, analyzing the reduction in the solution error value as well as the situation of the added nodes for different partial differential equation, defined in several domains. The main control parameter of the h-adaptive method is the error limit of each step.The error limit value for each step of the h-adaptive method, must be smaller than the average of the estimated error of the nodes and to include a number of nodes less than the 40% of total nodes.

The application of the h-adaptive method to several cases, shows that better results are obtained when the error limit is progressively reduced in every step in order to reach the benchmark error fixed for the problem.

The h-adaptive method proposed add nodes so that the possibility of ill-conditioned clouds of points is avoided.

Acknowledgement: The authors acknowledge the support from Ministerio de Ciencia e Innovación of Spain, project CGL2008 – 01757/CLI, and the support from Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha of Spain, project PA106 – 0123 – 7821.

References

Atluri, S. N.; Shen, S. P. (2002): *The meshless local Petrov-Galerkin (MLPG) method.* Tech. Science Press.

Atluri, S. N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in modeling and simulation. *Comput. Modeling Simulation in Engrg.*, vol. 3, pp. 187–196.

Atluri, S. N.; Zhu, T. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.*, vol. 22, pp. 117–127.

Atluri, S. N.; Liu, H. T.; Han, Z. D. (2006a): Meshless local Petrov-Galerkin (MLPG) mixed finite difference method for solid mechanics. *CMES: Computer Modelling in Engineering & Sciences.*, 15(1):1–16.

Atluri, S. N.; Liu, H. T.; Han, Z. D. (2006b): Meshless local Petrov-Galerkin (MLPG) mixed collocation method for elasticityproblems. *CMES: Computer Modelling in Engineering & Sciences.*, 14(3):141–152.

Babuska, I.; Melenk, J. M. (1997): The partition of unity method. *International Journal for Numerical Methods in Engineering*, vol 40, pp. 727–758.

Belytschko, T.;Lu, Y. Y.; Gu, L. (1994): Element free-Galerkin method. *Comput. Methods Appl. Mech. Eng.*, vol.113, pp. 397–414.

Benito, J. J.; Urena, F.; Gavete, L. (2001): Influence several factors in the generalized finite difference method. *AMM: Applied Mathematical Modeling*, vol 25, pp. 1039–1053.

Benito, J. J.; Urena, F.; Gavete, L.; Alvarez, R. (2003): An h-adaptive method in the generalized finite difference. *Comput. Methods Appl. Mech. Eng.*, vol 192, pp. 735–759.

Duarte, A.; Oden, J. T. (1996): Hp-cloud a meshless method to solve boundary-value problems. *Comput. Methods Appl. Mech. Eng.*, vol 139, pp. 237–262.

Gavete, L.; Gavete, M. L.; Benito, J. J. (2003): Improvements of generalized finite difference method and comparison with other meshless method. *AMM: Applied Mathematical Modeling*, vol 27, pp. 831–847.

Jensen, P. S. (1972): Finite difference technique for variable grids. *Computer& Structures*, vol 2, pp. 17–29.

Lancaster, P.; Salkauskas, K. (1981): Surfaces generated by moving least squares methods. *Math. Comput.*, vol 137, pp. 141–158.

Li, S.; Liu, W. K. (2004): *Mesh free particle methods.* Ed. Springer-Verlag.

Liszka, T.; Orkisz, J. (1980): The Finite Difference Method at Arbbitrary Irregular Grids and its Application in Applied Mechanics. *Computer& Structures*, vol 11, pp. 83–95.

Liu, W. K.; Jun, S.; Li, S.; Adee, J.; Belytschko, T. (1995): Reproducing Kernel Particle Methods for Structural Dynamics. *International Journal for Numerical Methods in Engineering*, vol 38, pp. 1655–1679.

Liu, G. R. (2003): *Mesh free methods.* Ed. CRC Press.

Monaghan, J. J. (1982): Why particle methods work. *SIAM Journal Sci. Stat. Comput.*, vol 3, pp. 422–433.

Monaghan, J. J. (1988): An introduction to SPH. *Computer Physics Communications*, vol 48, pp. 89–96.

Nayroles, B.; Touzot, G.; Villon, P. (1992): Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, vol 10, pp. 307–318.

Orkisz, J. (1998): *Finite Difference Method. (Part. III) in Handbook of Computational Solid Mechanics.* M. Kleiber(Ed.), Springer-Verlag, Berlin.

Perrone, N.; Kao, R. (1975): A general finite difference method for arbitrary meshes. *Computer& Structures*, vol 5, pp. 45–58.

Sukumar, N.; Moran, B.; Belytschko, T. (1998): The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering*, vol 43, pp. 839–887.

Tang, D.; Chen, X. K.; Yang, C.; Kobayashi, K. and Ku, D. N. (2002): A viscoelestic model and meshless GFD method for Blood Flow in colapsible Stenotic Arteries. *Advances in Computational Engineering & Sciences.* Chap. 11, Tech. Science Press, International Conference on Computational Engineering and Sciences.

Tang, D.; Yang, C.; Kobayashi, K. and Ku, D. N. (2001): A Generalized Finite Difference Method for 3-D Viscous Flow in Stenotic Tubes with Large Wall Deformation and Collapse. *Applied Num. Math.*, 38: pp. 49–68.

Urena, F.; Benito, J. J.; Gavete, L.; Alvarez, R.(2005): Computational Error Approximation and h-Adaptive Algorithm for the 3-D Generalized Finite Difference Method. *International Journal for Computational Methods in Engineering Science and Mechanics.*, vol 6, pp. 31–39.

Yang, C.; Tang, D.; Yuan, C.; Kerwin, W.; Liu, F.; Canton, G.; Hatsukami, T. S. and Atluri,S. N. (2008): Meshless Generalized Finite Difference Method and Human Carotid Atherosclerotic Plaque Progrssion Simulation using Multi-Year MRI Patiene-Tracking Data. *CMES: Computer Modelling in Engineering & Sciences.*, 28(2):95–107.