

Research Article

Solving Elliptical Equations in 3D by Means of an Adaptive Refinement in Generalized Finite Differences

Luis Gavete ¹, Francisco Ureña,² Juan Jose Benito,²
Miguel Ureña ², and Maria Lucia Gavete¹

¹ETSIME, Universidad Politécnica Madrid (UPM), Madrid, Spain

²ETSII, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

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

Received 28 November 2017; Revised 6 February 2018; Accepted 19 February 2018; Published 27 March 2018

Academic Editor: Luis J. Yebra

Copyright © 2018 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.

We apply a 3D adaptive refinement procedure using meshless generalized finite difference method for solving elliptic partial differential equations. This adaptive refinement, based on an octree structure, allows adding nodes in a regular way in order to obtain smooth transitions with different nodal densities in the model. For this purpose, we define an error indicator as stop condition of the refinement, a criterion for choosing nodes with the highest errors, and a limit for the number of nodes to be added in each adaptive stage. This kind of equations often appears in engineering problems such as simulation of heat conduction, electrical potential, seepage through porous media, or irrotational flow of fluids. The numerical results show the high accuracy obtained.

1. Introduction

The generalized finite difference method, hereafter GFDM, is a meshless method that allows the linearization of the partial derivatives in a partial differential equation. Among the first works of the method, the works of Jensen [1], Perrone and Kao [2], Liszka and Orkisz [3], and Orkisz [4] are highlighted. Later, Benito et al. [5] provided explicit formulae of the GFDM and Ureña et al. [6] provided a compact matrix formulation of the method. Gavete et al. [7] reported improvements of the GFDM and comparison with other meshless methods. Various applications and implementations of the GFDM have been developed for solving different inverse problems by Fan et al. [8, 9]. Hosseini [10, 11] has applied the GFDM for natural frequency analysis of nanocomposite cylinders and for shock-induced two-dimensional coupled non-Fickian diffusion-elasticity analysis.

Clark et al. [12] have investigated the problem of optimizing and adapting a surface mesh with high-order accurate nodal positions using the GFDM. The GFDM has been also used for solving 3D partial differential equations by Izadian et al. [13], Gavete et al. [14], and Hua et al. [15]. The GFDM was also applied for different practical tasks as

in Chan et al. [16] for solving two-dimensional nonlinear obstacle problems, in Zhang et al. [17, 18] for simulating two-dimensional sloshing phenomena and for propagation of nonlinear water waves, in Mochnacki and Majchrzak [19] for modelling of casting solidification, and in Li and Fan [20] for solving two-dimensional shallow water equations.

In 2D, two adaptive methods have been described in Benito et al. [21–23] using a posteriori error indicator. In 3D, an adaptive method has been described in Benito et al. [24]. These adaptive methods add nodes as midpoints between the central node and the furthest nodes in the star or in the barycentre of triangles formed in the star.

Both in 2D and in 3D, Ureña et al. [25] have developed an improved adaptive method based on previous ones and with the ability of moving and removing nodes. This improved adaptive method allows greater reductions of the error using even fewer nodes.

An advantage of increasing few nodes in a domain is to avoid the increasing in the calculation time. However, if the calculation time is not a problem and it is possible to add much more nodes, then it is better to use an adaptive method that places nodes in a more regular way than the improved adaptive method given in [25]. With this in mind,

Gavete et al. [26] developed an adaptive method using a quadtree algorithm in 2D obtaining satisfactory results. The improved adaptive method and the proposed adaptive method using quadtree structure are not incompatible; in fact, they complement each other.

It is more difficult to deal with adaptivity in 3D cases because the appearing of degenerated solutions is possible and it is also necessary to keep the number of nodes that are added in the refinement stages limited.

In this paper, an octree data structure for avoiding the possibility of obtaining degenerated solutions in the adaptive process is shown. Because of the local behaviour of the refinement procedure and after successive refinement iterations, a cloud of nodes with different nodal densities is obtained. The octree split is controlled by an error indicator and a constraint that limit the possibility of adding too many nodes in each refinement stage, in order to obtain smooth transitions between the different nodal densities in the model.

In the developed adaptive algorithm, the following characteristics are included: the use of a background octree structure, a criterion of selection of cells based on the variations of the gradient, an error indicator that works as a stop condition, and a limit for the addition of new nodes in each refinement stage. Moreover, different 3D elliptic partial differential equations are solved for cases with sharp gradients. A practical application to the electric potential problem with unknown solution is also included.

The remainder of the paper is organized as follows. In Section 2, the GFDM for solving 3D elliptic equations is reviewed. Section 3 describes the use of the octree structure, the error indicator, and the implemented 3D adaptive procedure. Several examples are given in Section 4 where the application of the method to reduce the error is analyzed for different elliptic PDEs including cases with singularities in the solution. Finally, conclusions are given in Section 5.

2. Generalized Finite Differences

Let $D \subset \mathbb{R}^3$ and $\mathbf{x} \in D$ be a domain and a point of such domain and let us consider the following 3D elliptical partial differential equation:

$$L(U(\mathbf{x})) = f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega = \text{int}(D) \quad (1)$$

with Dirichlet boundary condition

$$U(\mathbf{x}) = g(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma = \text{fr}(D), \quad (2)$$

where U is the unknown function and f and g are known functions.

Since generalized finite differences method is a meshless method, a discretization of the domain, $M \subset D$, is performed. Given a star $E(\mathbf{x}_0) = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$, the Taylor series is developed for each node in the star and the N resulting Taylor series are summed. Then the function given by the sum of the weighted quadratics errors in Taylor series for terms up to second order is defined:

$$B(\mathbf{D}_u) = \sum_{i=1}^N (u_0 - u_i + \boldsymbol{\varepsilon}_i^T \mathbf{D}_u)^2 w_i^2, \quad (3)$$

where $w_i = w(\mathbf{x}_0, \mathbf{x}_i)$ is the weighting function, $u_0 = u(\mathbf{x}_0)$ and $u_i = u(\mathbf{x}_i)$ are the approximations of $U_0 = U(\mathbf{x}_0)$ and $U_i = U(\mathbf{x}_i)$, respectively, and N is the number of nodes of the star without the central node. In 3D, the vectors \mathbf{D}_u and $\boldsymbol{\varepsilon}_i$ are

$$\mathbf{D}_u = \left(\frac{\partial u_0}{\partial x}, \frac{\partial u_0}{\partial y}, \frac{\partial u_0}{\partial z}, \frac{\partial^2 u_0}{\partial x^2}, \frac{\partial^2 u_0}{\partial y^2}, \frac{\partial^2 u_0}{\partial z^2}, \frac{\partial^2 u_0}{\partial x \partial y}, \frac{\partial^2 u_0}{\partial x \partial z}, \frac{\partial^2 u_0}{\partial y \partial z} \right)^T \quad (4)$$

$$\boldsymbol{\varepsilon}_i = \left(h_i, k_i, l_i, \frac{h_i^2}{2}, \frac{k_i^2}{2}, \frac{l_i^2}{2}, h_i k_i, h_i l_i, k_i l_i \right)^T.$$

In order to minimize the error, the function defined in (3) is minimized and the following system is obtained:

$$\mathbf{A} \cdot \mathbf{D}_u = \mathbf{b}, \quad (5)$$

where

$$\mathbf{A} = \sum_{i=1}^N w_i^2 \boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_i^T \quad (6)$$

$$\mathbf{b} = \sum_{i=1}^N w_i^2 (u_i - u_0) \boldsymbol{\varepsilon}_i.$$

Solving (5) for \mathbf{D}_u , the approximated values of the partial derivatives are obtained and, substituting these values in the original equation (1), the *equation of the star* is achieved:

$$-\lambda_0 u_0 + \sum_{i=1}^N \lambda_i u_i = f(\mathbf{x}_0). \quad (7)$$

The numerical values in each inner node are obtained by solving the linear system of equations formed by each equation of the star. For more details about this matrix formulation see [6].

Theorem 1. *\mathbf{A} is a positive definite matrix if and only if the set $\{\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_N\}$ contains a base of \mathbb{R}^9 .*

Proof. As $\mathbf{v}^T \mathbf{A} \mathbf{v} = \mathbf{v}^T \sum_{i=1}^N w_i^2 \boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_i^T \mathbf{v} = \sum_{i=1}^N w_i^2 \mathbf{v}^T \boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_i^T \mathbf{v} = \sum_{i=1}^N w_i^2 (\mathbf{v}^T \boldsymbol{\varepsilon}_i)^2$, it is followed that $\mathbf{A} \in \mathcal{M}_9(\mathbb{R})$ is a positive definite matrix if and only if, $\forall \mathbf{v} \in \mathbb{R}^9 - \{\mathbf{0}\}$, $\mathbf{v}^T \mathbf{A} \mathbf{v} > 0 \Leftrightarrow \nexists \mathbf{v} \in \mathbb{R}^9 - \{\mathbf{0}\}$ satisfying $\mathbf{v}^T \boldsymbol{\varepsilon}_i = 0$, $\forall i = 1, 2, \dots, N \Leftrightarrow \{\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_N\}$, contains a base of \mathbb{R}^9 . \square

Consistency, Order, and Convergence. In order to obtain the truncation error, Taylor series expansion including higher order derivatives is used and therefore a higher order function $B^*(\mathbf{D}_u)$ is obtained. The expression of $B^*(\mathbf{D}_u)$ is similar to the one given in (3). If $B^*(\mathbf{D}_u)$ is minimized with respect to the partial derivatives down to the second order, the following system is obtained:

$$\mathbf{A} \cdot \mathbf{D}_u = \mathbf{b}^* = \mathbf{b} + \mathbf{b}^{**}. \quad (8)$$

The new term \mathbf{b}^{**} corresponds to the new higher order derivatives incorporated in the Taylor series expansion to extend the function $B(\mathbf{D}_u)$ to $B^*(\mathbf{D}_u)$. Then

$$\mathbf{D}_u = \mathbf{A}^{-1} \cdot \mathbf{b}^* = \mathbf{A}^{-1} \cdot \mathbf{b} + \mathbf{A}^{-1} \cdot \mathbf{b}^{**} \quad (9)$$

and the truncation error of derivatives is

$$\begin{aligned} \text{TE}_{\mathbf{D}_u} &= \mathbf{A}^{-1} \cdot \mathbf{b}^{**} \\ \text{TE}_{\mathbf{D}_u} &= \Theta(h_i^2, k_i^2, l_i^2) \implies \lim_{(h_i, k_i, l_i) \rightarrow (0,0,0)} \text{TE}_{\mathbf{D}_u} \rightarrow \mathbf{0}. \end{aligned} \quad (10)$$

By considering bounded derivatives, the truncation error shows the consistency of the approximation for the equation given in (1).

Let $e_0 = u_0 - U_0$ be the error between the approximate and exact solution at the node \mathbf{x}_0 and let us consider (1) for U_0 and u_0 :

$$\begin{aligned} L_a[U] &= -\lambda_0 U_0 + \sum_{i=1}^N \lambda_i U_i = f(x_0, y_0, z_0) \\ L_a[u] &= -\lambda_0 u_0 + \sum_{i=1}^N \lambda_i u_i = f(x_0, y_0, z_0) \end{aligned} \quad (11)$$

Taking into account the fact that

$$|L_a[u] - L_a[U]| \leq \text{TE}_{\mathbf{D}_u} \quad (12)$$

it is followed

$$|L_a[e]| \leq \text{TE}_{\mathbf{D}_u} \quad (13)$$

and, appealing to the maximum principle [27], we find that

$$|e_0| \leq \text{TE}_{\mathbf{D}_u} = \Theta(h_i^2, k_i^2, l_i^2). \quad (14)$$

Therefore both the local truncation error $\text{TE}_{\mathbf{D}_u}$ and the error e are $\Theta(h_i^2, k_i^2, l_i^2)$ and the numerical solution converges to the exact solution as $(h_i, k_i, l_i) \rightarrow (0, 0, 0)$.

Further details about consistency, order, and convergence are given in [28].

Numerical Convergence. In order to study the rate of convergence numerically, a sequence of uniform discretizations is considered. As in [29], the parameter $\delta_k = E_{k+1}/E_k$ is defined and it was proven that $\delta_k \approx (h_k/h_{k+1})^\gamma$, where k represents the different discretizations, γ is the rate of convergence, and

$$E_k = \frac{\sqrt{(\sum_{i=1}^{\text{NI}} (U_i - u_i^k)^2) / \text{NI}}}{\max |U_i|} \times 100\%. \quad (15)$$

The considered domain is the unit cube and the considered partial differential equation is the following

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 0 \quad (16)$$

TABLE 1: Numerical convergence.

	δ_k	h_k/h_{k+1}	γ
$k = 1$	1.46	1.25	1.70
$k = 2$	2.32	1.60	1.79
$k = 3$	1.51	1.25	1.85
$k = 4$	3.74	2.00	1.90

with exact solution

$$U(x, y, z) = e^x \sin(y) + e^y \sin(z) + e^z \sin(x). \quad (17)$$

The considered sequence of discretizations with the same step in the three axes is

$$\begin{aligned} h_1 &= 0.25, \\ h_2 &= 0.2, \\ h_3 &= 0.125, \\ h_4 &= 0.1, \\ h_5 &= 0.05. \end{aligned} \quad (18)$$

Table 1 shows the values of δ_k , h_k/h_{k+1} , and γ . Figure 1 displays the numerical convergence.

As it can be appreciated, the rate of convergence is close to quadratic convergence.

3. An Adaptive Method Using Octree Algorithm

With the aim of improving the accuracy of the obtained approximation in a particular problem, it is possible to increase the number of nodes in the considered cloud of nodes. Since the GFDM allows using irregular clouds of nodes, the position of the added nodes can be selective and so it is possible to add more nodes in regions with higher errors. An important case of regions with higher errors gives rise when there are sharp variations of the gradient. In those cases, it is important to account for an adaptive algorithm that allows adding nodes as regular as possible.

An octree algorithm is a tree data structure based on a cell with eight children. Each cell of an octree represents a cube in the physical space and each child represents an octant of its parent. In the first step, the octree algorithm uses the initial nodes and, hereafter, the process is realized recursively; that is, each cell in turn can be split into eight children that have a higher level than their parent. Figures 2(a) and 2(b) display both the initial structure and the first stage of the refinement.

In spite of the fact that the GFDM is a meshless method and the application of the octree structure requires the computation of a mesh, it is important to note that there is not a loss of applicability in the method as the structure is just needed for the discretization of the domain, but not for the selection of the nodes of the star. Moreover, the use of the octree structure in the adaptive refinement places nodes in a favourable way in order to avoid ill-conditioned stars.

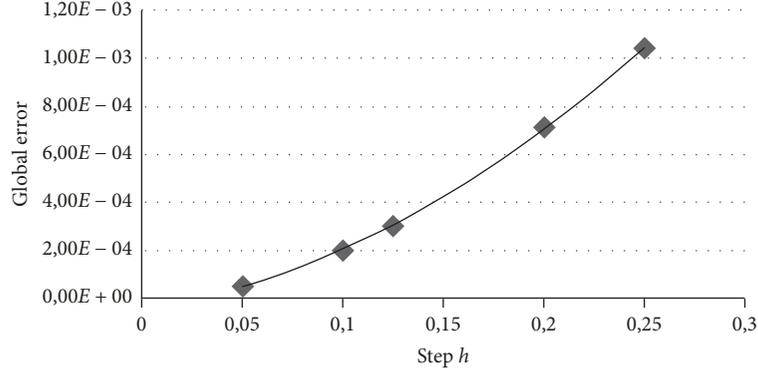


FIGURE 1: Numerical convergence. Abscissas represent the step of each discretization and ordinates represent the corresponding global error.

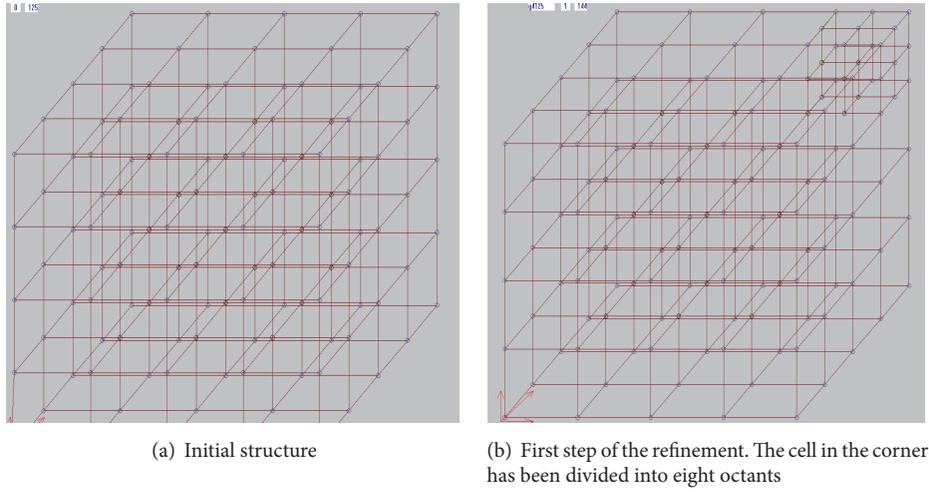


FIGURE 2

As error indicator, we use an algorithm based on the computing of the variations of the gradient between the nodes of an octree cell.

Firstly, the first-order derivatives are calculated by solving (5) and the gradient in each node is calculated as follows:

$$\nabla u_i = \sqrt{\left(\frac{\partial u_i}{\partial x}\right)^2 + \left(\frac{\partial u_i}{\partial y}\right)^2 + \left(\frac{\partial u_i}{\partial z}\right)^2}. \quad (19)$$

Secondly, for each cell, we calculate the mean value of the absolute variations of the gradient in the nodes of the cell divided by the distances and then this mean value is multiplied by the volume of the cell.

$$(\overline{\nabla u})_k = \text{Vol}_k \cdot \frac{(\sum_{i=1}^7 \sum_{j=i+1}^8 |\nabla u_i - \nabla u_j|) / \text{dist}(\mathbf{x}_i, \mathbf{x}_j)}{28}. \quad (20)$$

Finally, the criterion of selection of cells is given by

$$(\overline{\nabla u})_k > \beta_s \cdot \left(\frac{3}{4} \max_k (\overline{\nabla u})_k + \frac{1}{4} \min_k (\overline{\nabla u})_k \right), \quad (21)$$

where $k \in K$, K is the set of cells formed in the domain, and

$$\begin{aligned} \beta_0 &= 0.4 \\ \beta_{s+1} &= \beta_s + \frac{1}{20}s \end{aligned} \quad (22)$$

with $s = 1, 2, \dots$, nas and nas is the number of adaptive stages.

As stop condition, we calculate the following error between two successive adaptive stages:

$$e^{j+1} = \frac{\sqrt{(\sum_{i=1}^{\text{NI}} (u_i^{j+1} - u_i^j)^2) / \text{NI}}}{\max_i |u_i^{j+1}|} \cdot 100\%, \quad (23)$$

where u_i^j is the solution at the node i at the adaptive stage j and NI is the number of inner nodes at the adaptive stage j .

In order to achieve better results, the following consideration should be taken into account. The refined cells should be less than 15% of the number of cells in the domain in each stage due to the amount of nodes generated by the octree method, 8 new cells and 19 new nodes by cell. This situation can provoke an unnecessary increase of nodes in the same stage and because of that it is better to increase the number of nodes in several stages.

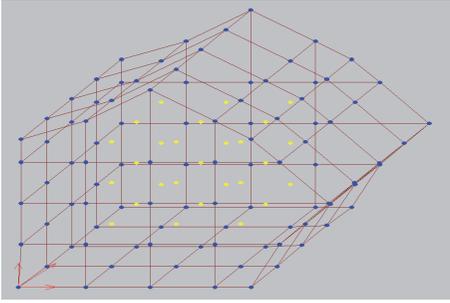


FIGURE 3: Initial cloud of nodes with 125 nodes.

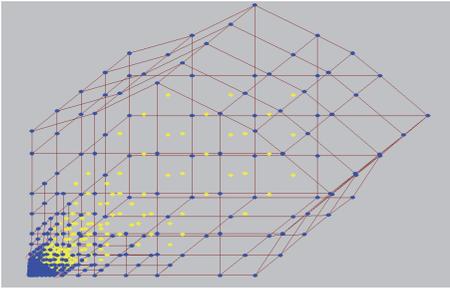


FIGURE 4: Final cloud of nodes with 1776 nodes.

4. Numerical Results

We solve five elliptic problems corresponding to partial differential equations with singularities or sharp variations of the gradients. In addition, we solve a practical case of potential electric where the exact solution is unknown.

In the following examples, we use stars with 16 nodes formed by the octant criterion and we use the potential weighting function

$$w_i = \|x_i - x_0\|_2^{-3}, \quad (24)$$

where $\|\cdot\|_2$ is the Euclidean norm.

The global error is calculated in each stage as follows:

$$E^j = \frac{\sqrt{\sum_{i=1}^{NI} (U_i - u_i^j)^2 / NI}}{\max |U_i|} \cdot 100\%, \quad (25)$$

where u_i^j is the solution at the node i at the adaptive stage j , U_i is the exact solution at the node i , and NI is the number of inner nodes at the adaptive stage j .

In order to clarify the visualization of the figures, we only show the background octree structure in the boundaries.

4.1. *Example 1.* Let us consider the following heat transmission problem:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 0 \quad (26)$$

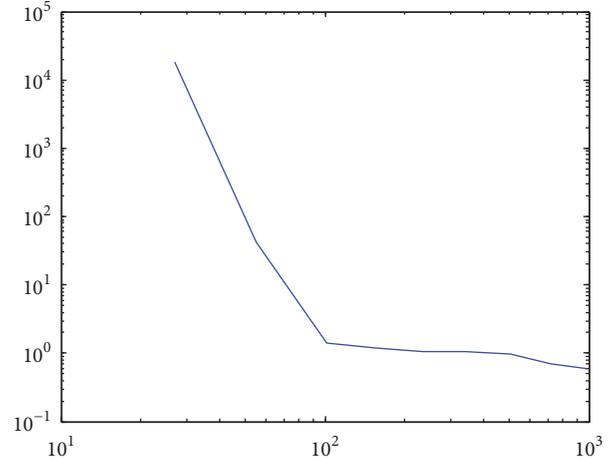


FIGURE 5: Reduction of the global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

with exact solution

$$U(x, y, z) = \frac{1}{\sqrt{x^2 + y^2 + z^2}}. \quad (27)$$

Note that in (27) there is a singular point in $(0,0,0)$, outside the domain. Figures 3 and 4 show the initial and final discretization of the domain. The initial discretization has 125 nodes; see Table 2. The final discretization has 1776 nodes (8 stages). The initial global error is 18596% and the final global error is 0.3% after applying the proposed adaptive refinement and Figure 5 shows the successive reductions of the global error in logarithmic scale.

4.2. *Example 2.* Let us consider the following Poisson problem:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 2 \cdot \frac{y^2 z^2 + x^2 z^2 + x^2 y^2}{x^3 y^3 z^3} \quad (28)$$

with exact solution

$$U(x, y, z) = \frac{1}{xyz}. \quad (29)$$

Note that in (29) there is a singular point in $(0, 0, 0)$, outside the domain. Figures 6 and 7 show the initial and final discretization of the domain. The initial discretization has 125 nodes and is generated as follows:

$$x(i) = \frac{1}{\sqrt{0.75}} (xc(i) - 0.51) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2} + 0.51$$

$$\begin{aligned}
y(i) &= \frac{1}{\sqrt{0.75}} (yc(i) - 0.51) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2} + 0.51 \\
z(i) &= \frac{1}{\sqrt{0.75}} (zc(i) - 0.51) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2} + 0.51
\end{aligned}$$

$0.01 \leq xc, yc, zc \leq 1.01$ with $xc, yc, zc = \{0.01, 0.026, 0.051, 0.076, 1.01\}$.

(30)

The final discretization has 608 nodes (8 stages). The initial global error is 69% and the final global error is 0.7% after applying the proposed adaptive refinement and Figure 8 shows the successive reductions of the global error in logarithmic scale.

4.3. *Example 3.* Let us consider the problem governed by the following partial differential equation:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = \frac{6}{(x + y + z)^3} \quad (31)$$

$$\begin{aligned}
x(i) &= \frac{1}{\sqrt{0.75}} xc(i) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2} \\
y(i) &= \frac{1}{\sqrt{0.75}} yc(i) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2} \\
z(i) &= \frac{1}{\sqrt{0.75}} zc(i) \sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2}
\end{aligned}$$

$0.01 \leq xc, yc, zc \leq 1.01$ with $xc, yc, zc = \{0.01, 0.026, 0.051, 0.076, 1.01\}$.

(33)

The final discretization has 863 nodes (7 stages). The initial global error is 78% and the final global error is 0.7% after applying the proposed adaptive refinement and Figure 11 shows the successive reductions of the global error in logarithmic scale.

4.4. *Example 4.* Let us consider the problem governed by the following partial differential equation:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = g(x, y, z) \quad (34)$$

with exact solution

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

$$x(i) = \frac{1.3 (xc(i) - 0.51)}{\sqrt{(xc(i) - 0.51)^2 + (yc(i) - 0.51)^2 + (zc(i) - 0.51)^2}} \max(|xc(i) - 0.51|, |yc(i) - 0.51|, |zc(i) - 0.51|) + 0.51$$

with exact solution

$$U(x, y, z) = \frac{1}{x + y + z}. \quad (32)$$

Note that in (32) there is a singular point in (0, 0, 0), outside the domain. Figures 9 and 10 show the initial and final discretization of the domain. The initial discretization has 125 nodes and is generated as follows:

$$\cdot (e^{10y^2} - 1) z^2 (1 - z)^2 (e^{10z^2} - 1) \quad (35)$$

and $g(x, y, z)$ is the function obtained by applying left hand side in (34) to the exact function.

Figures 12 and 13 show the initial and final discretization of the domain. The initial discretization has 125 nodes; see Table 3. The final discretization has 1449 nodes (7 stages). The initial global error is 188418% and the final global error is 0.15% after applying the proposed adaptive refinement and Figure 14 shows the successive reductions of the global error in logarithmic scale.

4.5. *Example 5.* Let us consider the problem governed by the partial differential equation (34) and with exact solution (35) in a different domain. Figures 15 and 16 show the initial and final discretization of the domain. The initial discretization has 125 nodes and is generated as follows:

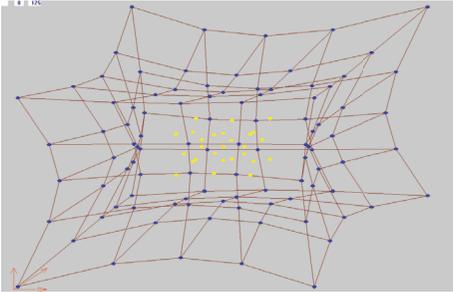


FIGURE 6: Initial cloud of nodes with 125 nodes.

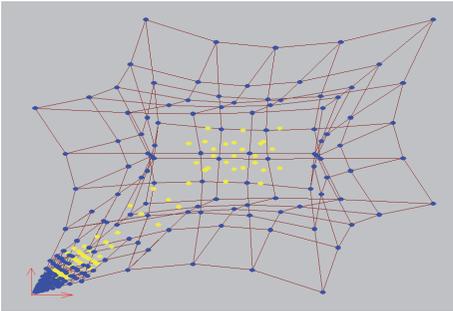


FIGURE 7: Final cloud of nodes with 608 nodes.

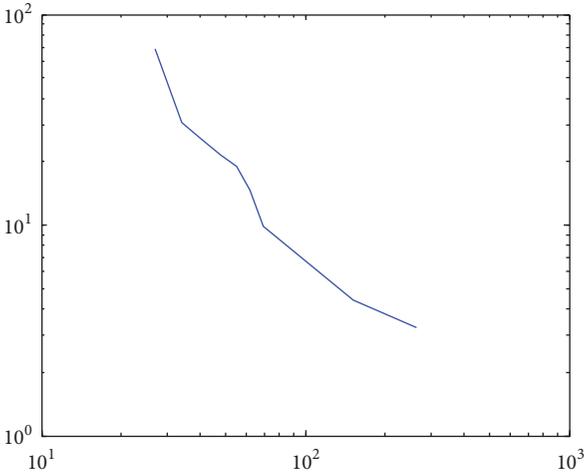


FIGURE 8: Reduction of the global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

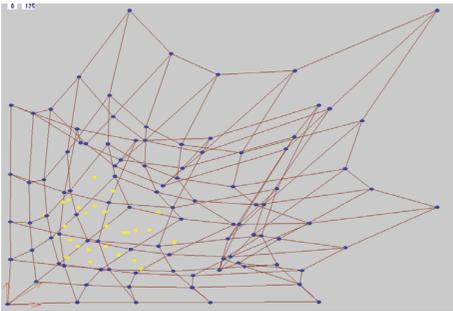


FIGURE 9: Initial cloud of nodes with 125 nodes.

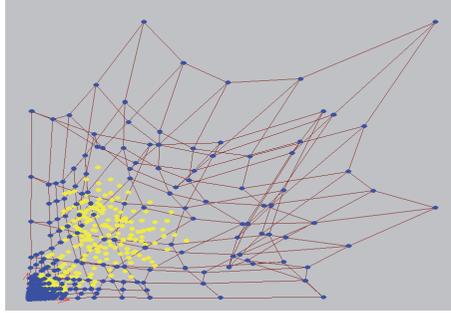


FIGURE 10: Final cloud of nodes with 863 nodes.

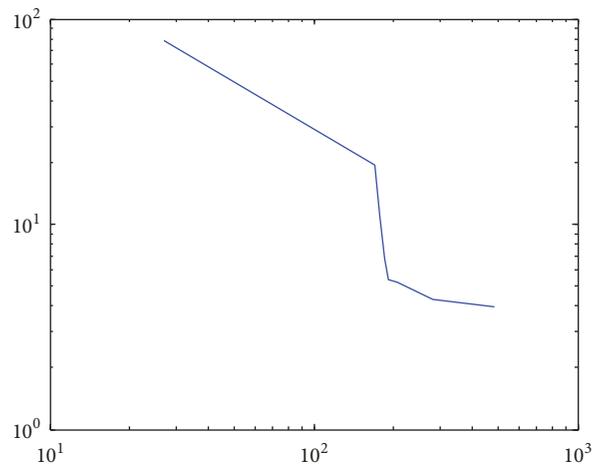


FIGURE 11: Reduction of the global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

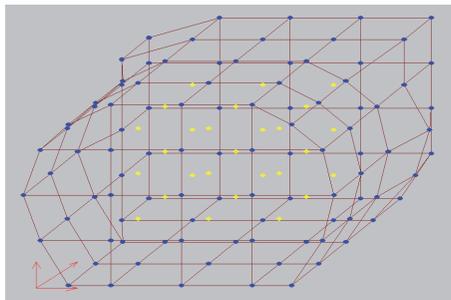


FIGURE 12: Initial cloud of nodes with 125 nodes.

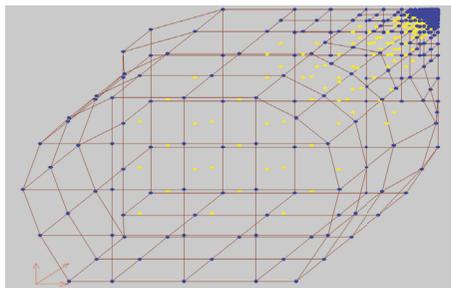


FIGURE 13: Final cloud of nodes with 1449 nodes.

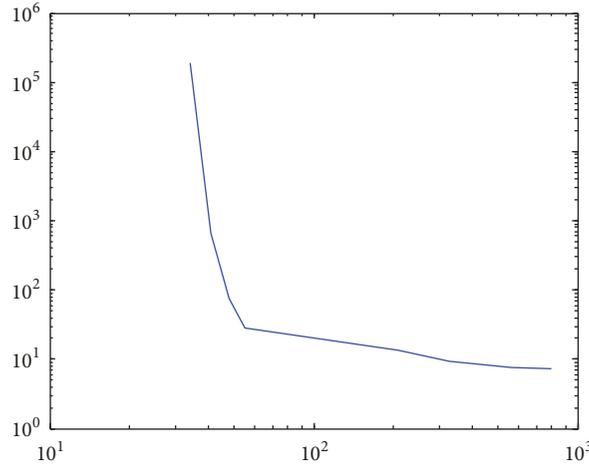


FIGURE 14: Reduction of the global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

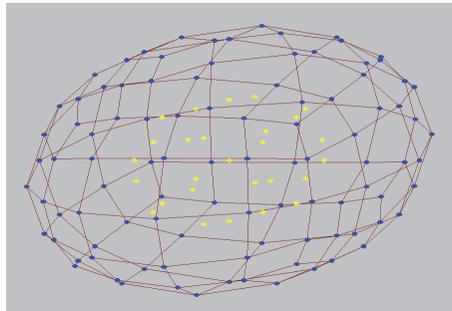


FIGURE 15: Initial cloud of nodes with 125 nodes.

$$y(i) = \frac{1.3(y_c(i) - 0.51)}{\sqrt{(x_c(i) - 0.51)^2 + (y_c(i) - 0.51)^2 + (z_c(i) - 0.51)^2}} \max(|x_c(i) - 0.51|, |y_c(i) - 0.51|, |z_c(i) - 0.51|) + 0.51$$

$$z(i) = \frac{1.3(z_c(i) - 0.51)}{\sqrt{(x_c(i) - 0.51)^2 + (y_c(i) - 0.51)^2 + (z_c(i) - 0.51)^2}} \max(|x_c(i) - 0.51|, |y_c(i) - 0.51|, |z_c(i) - 0.51|) + 0.51$$

$$0.01 \leq x_c, y_c, z_c \leq 1.01 \text{ with } x_c, y_c, z_c = \{0.01, 0.026, 0.051, 0.076, 1.01\}.$$

(36)

The final discretization has 3629 nodes (8 stages). The initial global error is 188935% and the final global error is 3% after applying the proposed adaptive refinement and Figure 17 shows the successive reductions of the global error in logarithmic scale.

The initial global errors in the examples are too high because we have generated starting clouds of few nodes to show the extreme conditions of the solutions involved. As it can be seen, the refinement gets large reductions of the global error in a few stages.

The efficiency of the algorithm adding progressively new nodes shows the reduction in the solution error. The refinement procedure creates new nodes in the neighborhood of areas with higher gradients; furthermore these nodes have

a more balanced distribution and it is not necessary further constraints to control ill-conditioned stars. Moreover, as the number of new nodes is limited in each adaptive stage, the transition between areas with different density of nodes is smooth.

The method can be also applied to the case of Neumann boundary conditions. In this case the procedure described in [14] for the treatment of Neumann boundary conditions can be used.

4.6. Example 6. Electric fields are used for designing high voltage equipment. In these cases, it is necessary to know the distribution of the electrical potential. In electrical power distribution, a busbar is a metallic strip for local high current

TABLE 2: Example 1.

Node	X	Y	Z
1	.00001	.00001	.00001
2	.2600	.0001	.0001
3	.5100	.0001	.0001
4	.7600	.0001	.0001
5	0.900	.0001	.0001
6	.0100	.2600	.0001
7	.2600	.2600	.0001
8	.5100	.2600	.0001
9	.7600	.2600	.0001
10	1.0100	.2600	.0001
11	.0100	.5100	.0001
12	.2600	.5100	.0001
13	.5100	.5100	.0001
14	.7600	.5100	.0001
15	1.2000	.5100	.0001
16	.0100	.7600	.0001
17	.2600	.7600	.0001
18	.5100	.7600	.0001
19	.7600	.7600	.0001
20	1.0100	.7600	.0001
21	.0100	0.9000	.0001
22	.2600	1.0100	.0001
23	.5100	1.2000	.0001
24	.7600	1.0100	.0001
25	.9000	.9000	.0001
26	.0100	.0001	.2600
27	.2600	.0001	.2600
28	.5100	.0001	.2600
29	.7600	.0001	.2600
30	0.9000	.0001	.2600
31	.0100	.2600	.2600
32	.2600	.2600	.2600
33	.5100	.2600	.2600
34	.7600	.2600	.2600
35	1.0100	.2600	.2600
36	.0100	.5100	.2600
37	.2600	.5100	.2600
38	.5100	.5100	.2600
39	.7600	.5100	.2600
40	1.2000	.5100	.2600
41	.0100	.7600	.2600
42	.2600	.7600	.2600
43	.5100	.7600	.2600
44	.7600	.7600	.2600
45	1.0100	.7600	.2600
46	.0100	0.9000	.2600
47	.2600	1.0100	.2600
48	.5100	1.2000	.2600
49	.7600	1.0100	.2600
50	0.9000	0.9000	.2600
51	.0100	.0001	.5100
52	.2600	.0001	.5100

TABLE 2: Continued.

Node	X	Y	Z
53	.5100	.0001	.5100
54	.7600	.0001	.5100
55	0.9000	.0100	.5100
56	.0100	.2600	.5100
57	.2600	.2600	.5100
58	.5100	.2600	.5100
59	.7600	.2600	.5100
60	1.0100	.2600	.5100
61	.0100	.5100	.5100
62	.2600	.5100	.5100
63	.5100	.5100	.5100
64	.7600	.5100	.5100
65	1.2000	.5100	.5100
66	.0100	.7600	.5100
67	.2600	.7600	.5100
68	.5100	.7600	.5100
69	.7600	.7600	.5100
70	1.0100	.7600	.5100
71	.0100	0.9000	.5100
72	.2600	1.0100	.5100
73	.5100	1.2000	.5100
74	.7600	1.0100	.5100
75	0.9000	0.9000	.5100
76	.0100	.0001	.7600
77	.2600	.0001	.7600
78	.5100	.0001	.7600
79	.7600	.0001	.7600
80	0.9000	.0100	.7600
81	.0100	.2600	.7600
82	.2600	.2600	.7600
83	.5100	.2600	.7600
84	.7600	.2600	.7600
85	1.0100	.2600	.7600
86	.0100	.5100	.7600
87	.2600	.5100	.7600
88	.5100	.5100	.7600
89	.7600	.5100	.7600
90	1.2000	.5100	.7600
91	.0100	.7600	.7600
92	.2600	.7600	.7600
93	.5100	.7600	.7600
94	.7600	.7600	.7600
95	1.0100	.7600	.7600
96	.0100	0.9000	.7600
97	.2600	1.0100	.7600
98	.5100	1.2000	.7600
99	.7600	1.0100	.7600
100	0.9000	0.9000	.7600
101	.0100	.0001	1.0100
102	.2600	.0001	1.0100
103	.5100	.0001	1.0100
104	.7600	.0001	1.0100
105	0.9000	.0001	1.0100

TABLE 2: Continued.

Node	X	Y	Z
106	.0100	.2600	1.0100
107	.2600	.2600	1.0100
108	.5100	.2600	1.0100
109	.7600	.2600	1.0100
110	1.0100	.2600	1.0100
111	.0100	.5100	1.0100
112	.2600	.5100	1.0100
113	.5100	.5100	1.0100
114	.7600	.5100	1.0100
115	1.2000	.5100	1.0100
116	.0100	.7600	1.0100
117	.2600	.7600	1.0100
118	.5100	.7600	1.0100
119	.7600	.7600	1.0100
120	1.0100	.7600	1.0100
121	.0100	0.9000	1.0100
122	.2600	1.0100	1.0100
123	.5100	1.2000	1.0100
124	.7600	1.0100	1.0100
125	0.9000	0.9000	1.0100

TABLE 3: Example 4.

Node	X	Y	Z
1	.1100	.0100	.0100
2	.2600	.0100	.0100
3	.5100	.0100	.0100
4	.7600	.0100	.0100
5	0.9000	.0100	.0100
6	.0100	.2600	.0100
7	.2600	.2600	.0100
8	.5100	.2600	.0100
9	.7600	.2600	.0100
10	1.0100	.2600	.0100
11	-.0500	.5100	.0100
12	.2600	.5100	.0100
13	.5100	.5100	.0100
14	.7600	.5100	.0100
15	1.0500	.5100	.0100
16	.0100	.7600	.0100
17	.2600	.7600	.0100
18	.5100	.7600	.0100
19	.7600	.7600	.0100
20	1.0100	.7600	.0100
21	.1100	0.9000	.0100
22	.2600	1.0100	.0100
23	.5100	1.0100	.0100
24	.7600	1.0100	.0100
25	0.9000	0.9000	.0100
26	.1100	.0100	.2600
27	.2600	.0100	.2600
28	.5100	.0100	.2600
29	.7600	.0100	.2600
30	0.9000	.0100	.2600
31	.0100	.2600	.2600
32	.2600	.2600	.2600
33	.5100	.2600	.2600
34	.7600	.2600	.2600
35	1.0100	.2600	.2600
36	-.0500	.5100	.2600
37	.2600	.5100	.2600
38	.5100	.5100	.2600
39	.7600	.5100	.2600
40	1.0500	.5100	.2600
41	.0100	.7600	.2600
42	.2600	.7600	.2600
43	.5100	.7600	.2600
44	.7600	.7600	.2600
45	1.0100	.7600	.2600
46	.1100	0.9000	.2600
47	.2600	1.0100	.2600
48	.5100	1.0100	.2600
49	.7600	1.0100	.2600
50	0.9000	0.9000	.2600
51	.1100	.0100	.5100
52	.2600	.0100	.5100

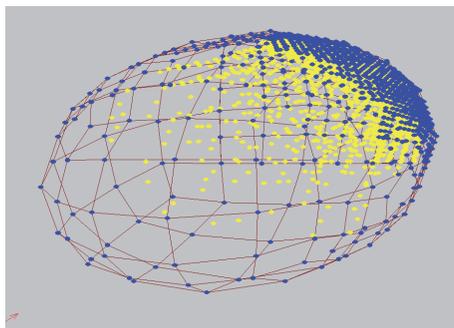


FIGURE 16: Final cloud of nodes with 3629 nodes.

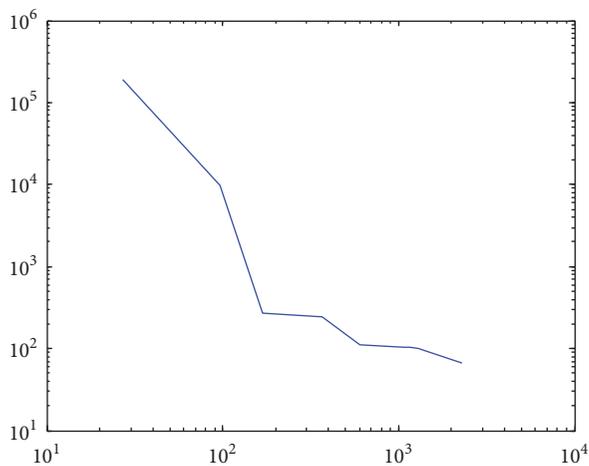


FIGURE 17: Reduction of the global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

TABLE 3: Continued.

Node	X	Y	Z
53	.5100	.0100	.5100
54	.7600	.0100	.5100
55	0.9000	.0100	.5100
56	.0100	.2600	.5100
57	.2600	.2600	.5100
58	.5100	.2600	.5100
59	.7600	.2600	.5100
60	1.0100	.2600	.5100
61	-.0500	.5100	.5100
62	.2600	.5100	.5100
63	.5100	.5100	.5100
64	.7600	.5100	.5100
65	1.0500	.5100	.5100
66	.0100	.7600	.5100
67	.2600	.7600	.5100
68	.5100	.7600	.5100
69	.7600	.7600	.5100
70	1.0100	.7600	.5100
71	.1100	0.9000	.5100
72	.2600	1.0100	.5100
73	.5100	1.0100	.5100
74	.7600	1.0100	.5100
75	0.9000	0.9000	.5100
76	.0100	.0100	.7600
77	.2600	.0100	.7600
78	.5100	.0100	.7600
79	.7600	.0100	.7600
80	0.9000	.0100	.7600
81	.0100	.2600	.7600
82	.2600	.2600	.7600
83	.5100	.2600	.7600
84	.7600	.2600	.7600
85	1.0500	.2600	.7600
86	.0100	.5100	.7600
87	.2600	.5100	.7600
88	.5100	.5100	.7600
89	.7600	.5100	.7600
90	1.1000	.5100	.7600
91	.0100	.7600	.7600
92	.2600	.7600	.7600
93	.5100	.7600	.7600
94	.7600	.7600	.7600
95	1.0100	.7600	.7600
96	.0100	0.9000	.7600
97	.2600	1.0100	.7600
98	.5100	1.0100	.7600
99	.7600	1.0100	.7600
100	1.0100	1.0100	.7600
101	.0100	.0100	1.0100
102	.2600	.0100	1.0100
103	.5100	.0100	1.0100
104	.7600	.0100	1.0100
105	0.9000	.0100	1.0100
106	.0100	.2600	1.0100

TABLE 3: Continued.

Node	X	Y	Z
107	.2600	.2600	1.0100
108	.5100	.2600	1.0100
109	.7600	.2600	1.0100
110	1.0100	.2600	1.0100
111	.0100	.5100	1.0100
112	.2600	.5100	1.0100
113	.5100	.5100	1.0100
114	.7600	.5100	1.0100
115	1.0100	.5100	1.0100
116	.0100	.7600	1.0100
117	.2600	.7600	1.0100
118	.5100	.7600	1.0100
119	.7600	.7600	1.0100
120	1.0100	.7600	1.0100
121	.0100	0.9000	1.0100
122	.2600	1.0100	1.0100
123	.5100	1.0100	1.0100
124	.7600	1.0100	1.0100
125	1.0100	1.0100	1.0100

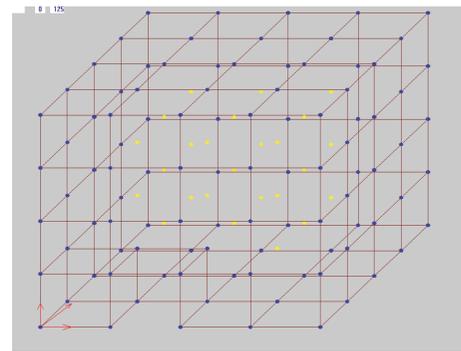


FIGURE 18: Initial discretization of the domain.

power distribution. We consider the case of a busbar box at potential 220 volts. The considered domain is $[0, 1] \times [0, 1] \times [0, 1]$ and the busbar is located at $[0.25, 0.5] \times [0, 0.25] \times [0, 0.25]$. The boundary of the domain has potential 0 volts except on the busbar where potential is 220 volts. The problem is governed by the Laplace equation and the exact solution is unknown. Note that this case is very different regarding the previous ones. The solution is smooth and the variations of the gradient are more widespread in the domain.

Figures 18 and 19 show the initial and final discretization of the domain. The initial discretization has 125 nodes and the final discretization has 399 nodes (8 refinement stages).

In the octree refinement, the new interior nodes of each stage are incorporated in the next adaptive stages as nodes with unknown potential. As the exact error is unknown, the results obtained in the last adaptive step have been considered as the best approximation to the solution; then formula (25) has been used to compute the global error approximation in the initial and successive adaptive stages by comparing the obtained values with that best approximation. The initial

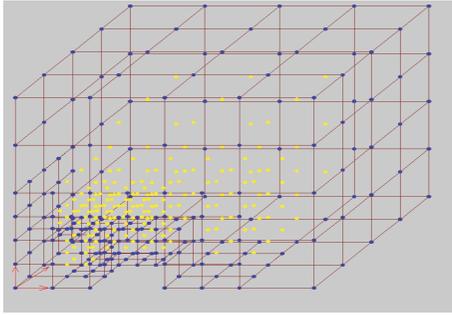


FIGURE 19: Final discretization of the domain.

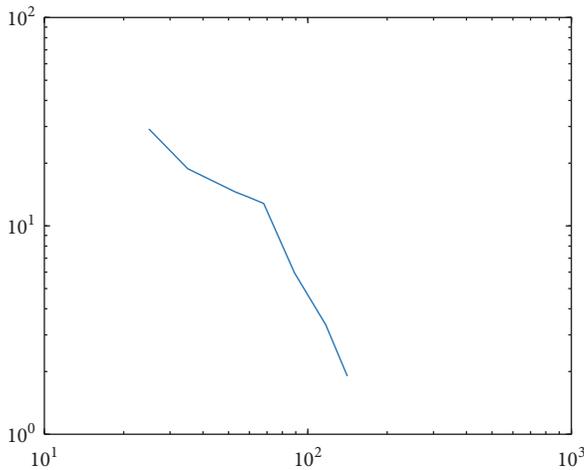


FIGURE 20: Reduction of the approximated global error in logarithmic scale. Abscissa axis represents the number of inner nodes.

global error approximation obtained by using (25) is 29.09% and the final global error approximation is 1.91%. Note that the global error (23) between two successive adaptive stages has been also used as stop condition of the refinement procedure with limit 2% in this case.

Figure 20 shows the successive reductions of the global error approximation in logarithmic scale.

5. Conclusions

A fully automatic 3D adaptive refinement procedure using GFDM for solving elliptic equations has been developed and implemented. An octree background structure has been used to guide the refinement of the clouds of nodes.

An error indicator has been used as a stop condition. This indicator is based on the differences between the gradients of the nodes in the same cell. In each adaptive stage, the number of new nodes is limited in order to obtain smooth transition areas between regions with different gradients. The efficiency of the algorithm is shown by means of the progressive decrease of the error. It is important to note that this adaptive method allows considering a coarse starting cloud of nodes and making it finer in several stages. This allows having a more suitable cloud of nodes adapted to the considered problem.

Simulation of heat conduction, electric potential, seepage through porous media, and irrotational flow of fluids are some problems where it is necessary to solve elliptical partial differential equations. The adaptive procedure refines the regions with high gradients and obtains very accurate results in different types of domain. All has been shown in the considered examples which is why the proposed refinement procedure will help to improve future applications of the GFDM for solving three-dimensional problems in complex domains.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the support of the Technical University of Madrid (Research Group) and the UNED of Spain, Project 2017-IFC02.

References

- [1] P. S. Jensen, "Finite difference technique for variable grids," *Computer & Structures*, vol. 2, pp. 17–29, 1972.
- [2] N. Perrone and R. Kao, "A general finite difference method for arbitrary meshes," *Computer & Structures*, vol. 5, pp. 45–58, 1975.
- [3] T. Liszka and J. Orkisz, "The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics," *Computer & Structures*, vol. 11, pp. 83–95, 1980.
- [4] J. Orkisz, "Finite Difference Method (Part, III)," in *handbook of Computational Solid Mechanics*, M. Kleiber, Ed., Springer-Verlag, Berlin, Germany, 1998.
- [5] 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.
- [6] M. Ureña, J. J. Benito, F. Ureña, E. Salet, and L. Gavete, "Application of generalised finite differences method to reflection and transmission problems in seismic SH waves propagation," *Mathematical Methods in the Applied Sciences*, vol. 41, no. 6, pp. 2328–2339, 2018.
- [7] 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.
- [8] C.-M. Fan, Y.-K. Huang, P.-W. Li, and C.-L. Chiu, "Application of the generalized finite-difference method to inverse biharmonic boundary-value problems," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 65, no. 2, pp. 129–154, 2014.
- [9] C.-M. Fan, P.-W. Li, and W. Yeih, "Generalized finite difference method for solving two-dimensional inverse Cauchy problems," *Inverse Problems in Science and Engineering*, vol. 23, no. 5, pp. 737–759, 2015.
- [10] S. M. Hosseini, "Application of a hybrid mesh-free method based on generalized finite difference (GFD) method for natural frequency analysis of functionally graded nanocomposite cylinders reinforced by carbon nanotubes," *CMES-Computer Modelling Engineering Science*, vol. 95, pp. 1–29, 2013.
- [11] S. M. Hosseini, "Shock-induced two dimensional coupled non-Fickian diffusion-elasticity analysis using meshless generalized finite difference (GFD) method," *Engineering Analysis with Boundary Elements*, vol. 61, pp. 232–240, 2015.

- [12] B. Clark, N. Ray, and X. Jiao, "Surface mesh optimization, adaption, and untangling with high-order accuracy," in *Proceedings of the 21st International Meshing Roundtable*, Weill Jiao, Ed., pp. 385–402, Springer, Berlin, Germany, 2013.
- [13] J. Izadian, N. Ranjbar, and M. Jalili, "The generalized finite difference method for solving elliptic equation on irregular mesh," *World Appl. Sci. J. (ISSN, vol. 1818-4952)* 21, pp. 95–100, 2013.
- [14] L. Gavete, J. J. Benito, and F. Ureña, "Generalized finite differences for solving 3D elliptic and parabolic equations," *Applied Mathematical Modelling*, vol. 40, no. 2, pp. 955–965, 2016.
- [15] Q. Hua, Y. Gu, W. Qu, W. Chen, and C. Zhang, "A meshless generalized finite difference method for inverse Cauchy problems associated with three-dimensional inhomogeneous Helmholtz-type equations," *Engineering Analysis with Boundary Elements*, vol. 82, pp. 162–171, 2017.
- [16] H.-F. Chan, C.-M. Fan, and C.-W. Kuo, "Generalized finite difference method for solving two-dimensional non-linear obstacle problems," *Engineering Analysis with Boundary Elements*, vol. 37, no. 9, pp. 1189–1196, 2013.
- [17] T. Zhang, Y.-F. Ren, C.-M. Fan, and P.-W. Li, "Simulation of two-dimensional sloshing phenomenon by generalized finite difference method," *Engineering Analysis with Boundary Elements*, vol. 63, pp. 82–91, 2016.
- [18] T. Zhang, Y.-F. Ren, Z.-Q. Yang, C.-M. Fan, and P.-W. Li, "Application of generalized finite difference method to propagation of nonlinear water waves in numerical wave flume," *Ocean Engineering*, vol. 123, pp. 278–290, 2016.
- [19] B. Mochnacki and E. Majchrzak, "Numerical modeling of casting solidification using generalized finite difference method," *Materials Science Forum*, vol. 638–642, pp. 2676–2681, 2010.
- [20] P.-W. Li and C.-M. Fan, "Generalized finite difference method for two-dimensional shallow water equations," *Engineering Analysis with Boundary Elements*, vol. 80, pp. 58–71, 2017.
- [21] J. J. Benito, F. Ureña, L. Gavete, and R. Alvarez, "An h-adaptive method in the generalized finite differences," *Computer Methods Applied Mechanics and Engineering*, vol. 192, no. 5–6, pp. 735–759, 2003.
- [22] 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.
- [23] J. J. Benito, F. Ureña, L. Gavete, and B. Alonso, "Application of the Generalized Finite Difference Method to improve the approximated solution of pdes," *Computer Modelling in Engineering & Sciences*, vol. 38, no. 1, pp. 39–58, 2008.
- [24] J. J. Benito, F. Ureña, R. Alvarez, and L. Gavete, "Computational error approximation and H-Adaptive Algorithm for the 3-D Generalized Finite Differences Method," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 6, pp. 31–39, 2005.
- [25] M. Ureña, J. J. Benito, F. Ureña, A. García, L. Gavete, and L. Benito, "Adaptive strategies to improve the application of the generalized finite differences method in 2D and 3D," *Mathematical Methods in the Applied Sciences*, 2017.
- [26] L. Gavete, M. L. Gavete, F. Ureña, and J. J. Benito, "An approach to refinement of irregular clouds of points using generalized finite differences," *Mathematical Problems in Engineering*, vol. 2015, Article ID 283757, 2015.
- [27] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, John Wiley & Sons, New York, NY, USA, 1980.
- [28] J. J. Benito, F. Ureña, and L. Gavete, "Solving parabolic and hyperbolic equations by the generalized finite difference method," *Journal of Computational and Applied Mathematics*, vol. 209, pp. 208–233, 2007.
- [29] D. Vishnevsky, V. Lisitsa, V. Tcheverda, and G. Reshetova, "Numerical study of the interface errors of finite-difference simulations of seismic waves," *Geophysics*, vol. 79, pp. T219–T232, 2014.



Hindawi

Submit your manuscripts at
www.hindawi.com

