Numer Algor (2012) 61:83–103
DOI 10.1007/s11075-011-9531-0

ORIGINAL PAPER

Adapting nodes in three dimensional irregular regions for meshless-type methods

Kamal Shanazari · Mohammad Hosami

Received: 13 July 2011 / Accepted: 21 December 2011 /
Published online: 12 January 2012

Abstract In this paper, a new adaptive nodes technique based on equi-distribution principles and dimension reduction is presented for irregular regions in three dimensional cases. The mesh generation is performed by first producing some adaptive nodes in a cube based on equi-distribution along the coordinate axes and then transforming the generated nodes to the physical domain followed by a refinement process. The mesh points produced are appropriate for meshless-type methods which need only some scattered points rather than a mesh with some smoothness properties. The effectiveness of the generated mesh points is examined by a collocation meshless method using a well known radial basis function, namely $\phi(r) = r^5$ which is sufficiently smooth for our purpose. Some experimental results will be presented to illustrate the effectiveness of the proposed method.

Keywords Adaptive mesh · Equi-distribution · Collocation meshless method · Dimension reduction method, Three dimension · Irregular regions

1 Introduction

In the recent years the meshless methods have attracted more attention as a powerful tool to deal with the partial differential equations (PDEs). The main advantage of this technique over the mesh dependent methods such as finite element, finite difference and boundary element method is that they

K. Shanazari (✉) · M. Hosami
Department of Mathematics, University of Kurdistan, Sanandaj, Iran
e-mail: k.shanazari@uok.ac.ir, kamal_shanazari@yahoo.co.uk

M. Hosami
e-mail: mohammad_hosami@yahoo.com

do not require a mesh or element to discretize the domain or boundary which is a difficult task, especially in the three dimensional (3D) problems. In stead this method needs only some scattered points whose connectivity are not important. The meshfree methods are often divided into two major categories: boundary-type (see for example [22, 27]) and domain-type methods (see for example [12]). This study concerns with a collocation meshless method which falls into the domain-type category and was first presented by Kansa [14, 15]. The Kansa's work was based on interpolating the solution in terms of radial basis functions (RBFs). This method can be easily applied to the case of higher dimensional spaces due to the nature of the RBFs (see [28]). The condition number of RBF interpolation matrix dramatically increases for the large scale problems. In particular, the size of the 3D problems are essentially large and, consequently, the interpolation matrix in this case becomes ill-conditioned [16].

Another difficulty concerns their computational complexity, due to the dense matrices arising from interpolation. To tackle the above difficulties some sort of localization, such as domain decomposition methods (DDM) [20] and compactly supported RBFs (CS-RBFs) [6], have been recommended.

To treat ill-conditioning, this paper suggests a new approach based on reducing the size of the interpolation matrices. As was noted before, in using the classical RBFs, increasing the size of the problem itself affects the conditioning. Consequently, any technique which can reduce the number of nodes would improve the conditioning. In order to achieve this goal a set of adaptive nodes rather than equally spaced points can be applied. As is well known, the main idea in adaptive meshes is to use a minimum number of nodes while keeping a desired accuracy. This is achieved by allocating more mesh points to the areas where the solution is rapidly varied. The adaptive mesh strategies often fall into two categories: the equi-distribution principle [9] and the variational principle [30]. The most popular technique, which has been widely considered in the literature, is based on the equi-distribution strategy, which is also employed in this work. In this method the mesh distribution is carried out in such a way that some measure of error, called a monitor function, is equalized over each subinterval.

Much effort has been devoted to generating adaptive meshes in two and three dimensional spaces, based on both equi-distribution and variational principles (see for example [2, 24, 25]). In the literature two major methods, namely transformation [7] and dimension reduction [26], have been employed to produce 2D meshes. The first category is based on mapping the physical domain into a simple domain with a uniform mesh and it leads to solving a differential equation in order to obtain an adaptive mesh. In the latter category, which is also used in this study, the equi-distribution is reduced to the case of 1D. We previously presented a method in [21] to produce a set of adaptive nodes in rectangular regions. Recently, a generalization of this approach to the case of non-rectangular domains was presented in [23]. This was based on transforming the physical domain to a rectangle using the polar coordinate system and applying the method described in [21]. A more

generalization of the above methods to the case of 3D has been presented in [24] to generate adaptive nodes in a cube. The method was based on equi-distributing along the three coordinate axes. The mesh points produced by the above method was successfully applied to a collocation meshless method. However, due to the use of the grid lines in the coordinate directions, this method was limited to the case of cubic domains.

The purpose of the present work is to extend the above method to more general cases of 3D domains with irregular boundaries. This is carried out by employing a suitable transformation which maps the physical domain into a cube using the spherical coordinates. A set of adaptive points, first, is generated in the cube using the method presented in [24] and, then, it is mapped into the physical domain. The mesh points produced in the main domain are not smoothly distributed due to the nature of the spherical transformation and need some refining process.

We remark that constructing monitor functions for different applications is a line of research which is widely seen in the literature [3, 4, 8, 18]. For example, some optimal monitors for piecewise polynomial interpolation have been derived in [4, 8]. In addition, some monitors suitable for linear reaction diffusion two-point boundary value problem have been presented in [18]. While the former monitors depend on the derivative of the solution, the latter ones are constructed based on the finite differences of the solution. The present work is not concerned with giving a special monitor for the underlying method. We employ some well known monitors including the arc-length and those presented in [4, 8] to examine the effectiveness of the adaptation method. Our used monitors depend on the derivatives of the solution function, however, they are approximated by a finite difference formula.

This paper is organized as follows. The concept of adaptive mesh in 1D and 3D is reviewed in Section 2. In Section 3 the new mesh generation technique is presented. A collocation meshless method will be reviewed in Section 4. Finally, some numerical results are presented in Section 5.

2 Adaptive mesh

We first introduce the concept of equi-distribution in the 1D case.

Definition 1 (Equi-distributing) Let M be a non-negative piecewise continuous function on $[a, b]$ and c be a constant such that $n = \frac{1}{c}\int_a^b M(x)dx$ is an integer. The mesh

$$\Pi : \quad a = x_0 < x_1 < \cdots < x_n = b,$$

is called equi-distributing (e.d.) on $[a, b]$ with respect to M and c if

$$\int_{x_j}^{x_{j+1}} M(x)dx = c, \qquad j = 0, 1, \cdots, n-1.$$

A suitable algorithm to produce an e.d. mesh has been given in [17]. In Definition 1 the function M, often called a monitor, depends on the solution of the PDE and its derivatives. The arc-length monitor

$$M = \sqrt{1 + u_x^2}\,, \tag{1}$$

which is also employed in this work, has been widely used, as a general monitor, in the literature (see for example [1, 29]). The function u in (1) is the solution function of the underlying PDE, x is the coordinate in the direction of which the adaptivity is performed, and u_x is the partial derivative with respect to x.

Much effort has been devoted to finding monitor functions suitable for different applications. Good examples can be found in [4, 8]. Carey and Dinh considered the interpolation by a kth degree piecewise polynomial and obtained the following monitor based on minimization of the H^m-seminorm of the local truncation error [4]

$$M = [u^{(k+1)}]^{2/[2(k+1-m)+1]}.$$

For instance, the choice of $m = 1$ in the case of a quadratic polynomial ($k = 2$) gives the monitor $M = (u^{''})^{2/5}$.

Chen generalized the above work by employing a Sobolev (l_1, l_2)-seminorm and obtained the Monitor (see [8]).

$$M = \left\{ \sum_{r=l_1}^{l_2} \left[u^{k+r+1-l_1)} \right]^2 \right\}^{1/[2(k+1-l_1)+1]}.$$

For example, this monitor, in the case of a quadratic polynomial with the choice of $l_1 = 0$ and $l_2 = 2$, leads to the monitor

$$M = \left[u^{'''2} + u^{''''2} + u^{'''''2} \right]^{1/7}.$$

In Section 5, we will employ these monitors to test our proposed adaptation method. To find more details about the monitors see for example [3, 5, 18].

2.1 3D equi-distribution

A natural extension of Definition 1 to the case of 3D is as follows,

Definition 2 (3D Equi-distributing) Given a 3D domain Ω, an adaptive mesh based on equi-distributing will be a mesh obtained by dividing the domain Ω into n subdomains Ω_i such that

$$\int\int\int_{\Omega_i} M(x, y, z)dxdydz = constant,$$

where M is a suitable monitor function.

Obviously an infinite number of adaptive meshes based on Definition 2 exist, though obtaining even one of these meshes is a very difficult task. Below we review a method previously presented to produce adaptive nodes in a cubic domain [24]. This approach was base on dimension reduction, i.e. reducing the adaptive process to the 1D case, as performed for the 2D case in [21].

We start with a uniform mesh in a cube in the form

$$\{(x, y, z) |\, a_1 \le x \le b_1,\ a_2 \le y \le b_2,\ a_3 \le z \le b_3\}\,.$$

We perform the equi-distribution process in four stages. The first three stages are the same as those in the 2D case, as suggested in [21]. In the first stage, equi-distribution is performed in the direction of the x-axis (Fig. 1a). In the second stage the equi-distribution is performed in the vertical direction along the grid lines produced in the first stage. Since the grid lines are curved, the equi-distribution is performed along the arc rather than the vertical coordinate (Fig. 1b). A similar procedure is performed in the third stage along the horizontal grid curves, with the monitor in the x coordinate direction again (Fig. 1c).

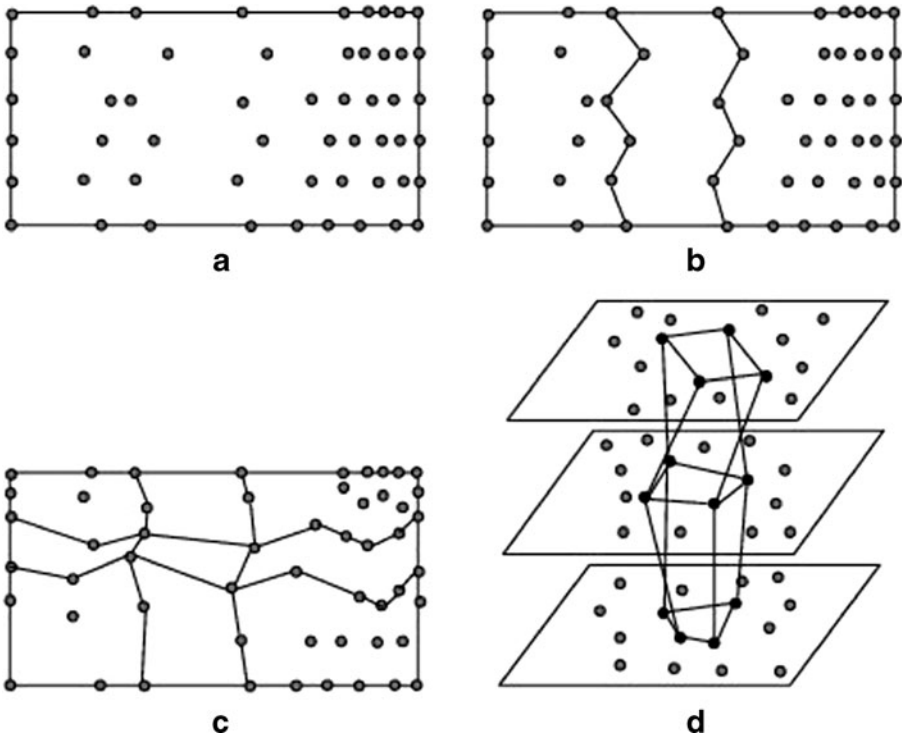

Fig. 1 The four stages of adapting mesh are displayed in (**a**), (**b**), (**c**) and (**d**), respectively

The mesh resulting from the above procedure in each plane $z = z_k$ forms quadrilaterals whose sides equi-distribute the grid lines in the two coordinate directions (see Fig. 1c).

In the final stage, the equi-distribution is performed with respect to the third coordinate, i.e. equi-distributing along the grid curves, roughly parallel to the z-axis using the monitor function $M_z(x, y, z)$. This stage of equi-distribution is illustrated by a simple case in Fig. 1d showing three planes and four grid curves.

The above process results in a 3D mesh containing hexahedrons whose edges form 1D e.d. meshes along the grid curves in the three coordinate directions (see Fig. 1d). To find more details see [24]. In the next Section we apply this method to the case of non-cubic domains.

3 Adapting mesh in an irregular region

Let Ω_p be a physical domain in $\Re^3$ bounded by Γ whose parametric equations are given by

$$\begin{aligned} \Gamma: \quad & x = g_1(\theta, \phi),\ y = g_2(\theta, \phi),\ z = g_3(\theta, \phi), \\ & 0 \le \theta \le 2\pi,\ 0 \le \phi \le \pi, \end{aligned} \tag{2}$$

where θ and ϕ are the angular components in the spherical coordinate system. We introduce a suitable mapping to transform Ω_p to a cubic domain. In order to do this, we use the cube

$$\Omega_c = \{(\rho, \theta, \phi): \quad 0 \le \rho \le 1, \quad 0 \le \theta \le 2\pi, \quad 0 \le \phi \le \pi\}, \tag{3}$$

in the cartesian coordinate system $\rho\theta\phi$ and employ a transformation

$$\psi : \Omega_c \to \Omega_p \tag{4}$$

defined by

$$x = \rho g_1(\theta, \phi), \quad y = \rho g_2(\theta, \phi), \quad z = \rho g_3(\theta, \phi). \tag{5}$$

The above mapping transforms each point $(\rho, \theta, \phi) \in \Omega_c$ to a unique point $(x, y, z) \in \Omega_p$ under some restrictions on the mesh points. In this transformation $\rho = 1$ corresponds to the boundary Γ and for any $0 < \bar{\rho} < 1$, $\rho = \bar{\rho}$ corresponds to a closed surface in xyz coordinate system. Therefore, any subregion

$$\Omega_{c_i} = \{(\rho, \theta, \phi): \quad 0 < \rho_1 \le \rho \le \rho_2 \le 1, \quad 0 \le \theta \le 2\pi, \quad 0 \le \phi \le \pi\},$$

in Ω_c is mapped into a subdomain Ω_{p_i} which is a region enclosed by two closed surfaces $\rho = \rho_1$ and $\rho = \rho_2$ in Ω_p.

The main idea in this study is, first, prepare a uniform mesh

$$\begin{aligned} 0 = \theta_0 < \theta_1 < \theta_2 < \ldots < \theta_{n-1} = 2\pi - h_\theta, \\ h_\phi = \phi_0 < \phi_1 < \ldots < \phi_{m-1} = \pi - h_\phi \,, \\ h_\rho = \rho_1 < \rho_2 < \ldots < \rho_{k-1} < \rho_k = 1 \,, \end{aligned} \tag{6}$$

where

$$\begin{aligned} h_\theta &= \frac{2\pi}{n} \quad , \qquad \theta_j = j h_\theta \,, \qquad\qquad j = 0, 1, \ldots, n-1 \,, \\ h_\phi &= \frac{\pi}{m} \quad , \qquad \phi_i = (i+1) h_\phi \,, \quad i = 0, 1, \ldots, m-1 \,, \\ h_\rho &= \frac{1}{k} \quad , \qquad \rho_l = l h_\rho \,, \qquad\qquad l = 1, 2, \ldots, k \,, \end{aligned} \tag{7}$$

in the cube defined by (3) and then make the grid mesh adaptive in the cube using the method described in Section 2. It should be noted that the points $\theta = 2\pi$, $\phi = 0, \pi$, and $\rho = 0$ are excluded from the uniform mesh points (6) in order to have a one to one corresponding between Ω_c and Ω_p. Finally, the adaptive nodes produced in Ω_c are mapped into the physical region Ω_p. However, the mesh points transformed into Ω_p will be clustered in some parts of the domain due to the nature of the spherical coordinate system (see Fig. 2). This issue is treated by a refinement process similar to that presented in [23] for two dimensional cases. For this purpose we first present a method to produce some roughly uniform mesh points in Ω_p using a refinement process. Of course there are easier ways to produce a uniform mesh, but, the proposed method will be combined with the adapting process later.

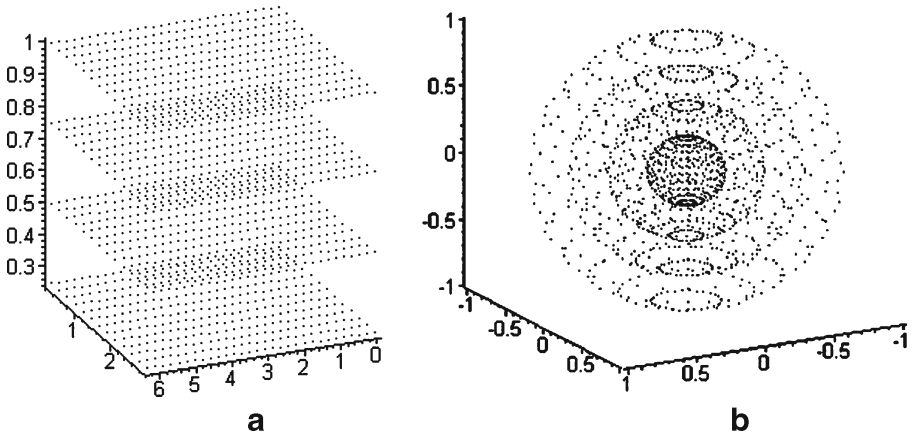

Fig. 2 The uniform mesh in Ω_c and the corresponding mesh points in Ω_p are displayed in (**a**) and (**b**), respectively, in the case of a spherical domain

3.1 A uniform mesh in the physical domain

We propose a process to generate some uniform mesh points in the physical domain Ω_p. We start with a uniform mesh in the cartesian coordinate system $\rho\theta\phi$ (Fig. 2a). Acting the transformation (4) on the uniform mesh, some new points (x, y, z) in Ω_p are produced as displayed in Fig 2b. Similar to the two dimensional case, the new mesh points are not uniformly distributed due to the fact that the same number of nodes are distributed on the concentric surfaces with different area and this make the nodes cluster in some parts, especially, when the coordinate ρ becomes too small.

Constructing a uniform mesh in Ω_p requires a refining process which is more complicated than that in the 2D case (see [23]). Here, in addition to the refinement applied in the 2D case, two more refinements are proposed to modify the number of nodes on each surface. Below we present a three-stage refinement in order to obtain a uniform mesh in Ω_p.

Suppose that the uniform mesh in Ω_c is given by (6) and the coordinates of the points are denoted by

$$\left\{\left(\rho_{ijk}, \theta_{ijk}, \phi_{ijk}\right) : \ i = 0, 1, \ldots, n_\theta, \quad j = 0, 1, \ldots, n_\phi, \quad k = 1, \ldots, n_\rho\right\}, \tag{8}$$

and the corresponding points in the xyz system are indicated by $(x_{ijk}, y_{ijk}, z_{ijk})$.

Stage 1 The first stage of refinement is to find a suitable number of nodes on each closed surface $\rho = \rho_k$. This can be accomplished by determining an approximate value of the area for each surface $\rho = \rho_k$ and choosing the number of nodes based on the area evaluated. In order to do so, we approximate the area of the kth surface by the area of a sphere whose radius is equal to the average values of the position vector, i.e.

$$R_k = \frac{\sum_{i=0}^{n_\theta} \sum_{j=0}^{n_\phi} R_{ij}}{(n_\theta + 1)\left(n_\phi + 1\right)}, \qquad k = 1, \ldots, n_\rho,$$

where

$$R_{ij} = \sqrt{x_{ijk}^2 + y_{ijk}^2 + z_{ijk}^2}.$$

A suitable number of nodes on $\rho = \rho_k$, therefore, can be given by

$$n_\rho(k) = \frac{s(k)}{s\left(n_\rho\right)} \times M, \tag{9}$$

where $s(k)$ is the area of the sphere associated with the kth surface and M is the number of nodes on the boundary and will be determined based on the next stages of the refinement.

Stage 2 This stage involves a refinement of the nodes associated with the coordinate ϕ. As can be seen in Fig. 3, by decreasing the coordinate ρ, the curves associated with ϕ_j's become closer. This causes the grid points cluster for small values of ρ. Thus, an appropriate partition of the coordinate ϕ for each surface $\rho = \rho_k, \ k = 1, \cdots, n_\rho$, is required.

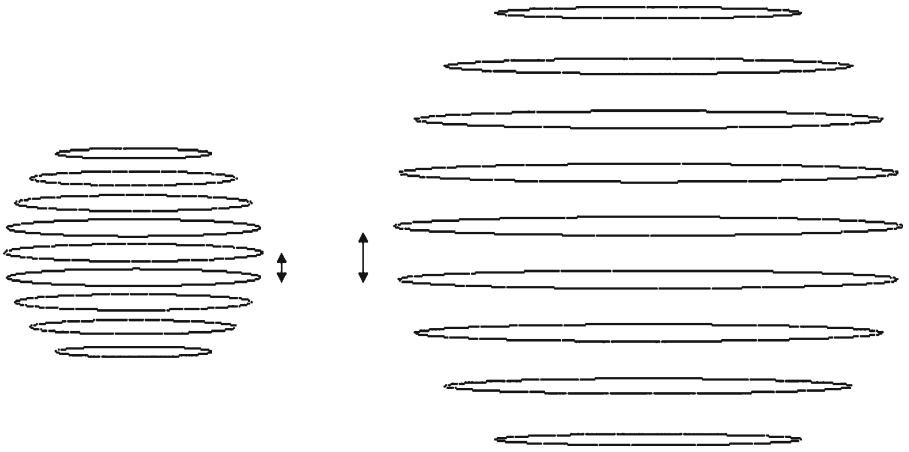

Fig. 3 The distance between the curves corresponding to ϕ_j's are closer for smaller values of ρ

In order to do so for a surface $\rho = \rho_k$, we first evaluate an average length of the curves formed by the ϕ_j's over the variation of θ. More precisely, for $\theta = \theta_i$ the length of such a curve can be evaluated by (see Fig. 4)

$$l'_i = \sum_{j=0}^{n_\phi - 1} l''_{ij}, \qquad i = 0, 1, \ldots, n_\theta,$$

where

$$l''_{ij} = \sqrt{\left(x_{i,j+1,k} - x_{i,j,k}\right)^2 + \left(y_{i,j+1,k} - y_{i,j,k}\right)^2 + \left(z_{i,j+1,k} - z_{i,j,k}\right)^2}.$$

An average measure of these curves are given by

$$l(k) = \left(\sum_{i=0}^{n_\theta} l'_i\right) / (n_\theta + 1). \tag{10}$$

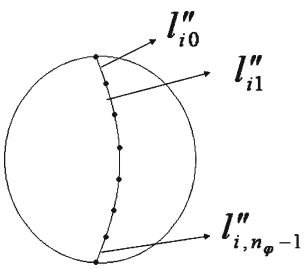

Fig. 4 The arc-length of the curves associated with θ_i's

It is quite reasonable that $n_\phi(k)$, the number of partitions for ϕ on $\rho = \rho_k$ to be proportional to the length derived in (10), that is

$$n_\phi(k) = \frac{l(k)}{l\left(n_\rho\right)} \times \left(n_\phi + 1\right). \tag{11}$$

So far, for each surface $\rho = \rho_k$, we have obtained a total number of nodes suitably selected and an appropriate mesh size for coordinate ϕ.

Stage 3 The last stage aims to select a suitable partition for the coordinate θ. More precisely, $n_\theta(j,k)$, the number of steps along the θ axes for $k = 1, \cdots, n_\rho$, $j = 0, \cdots, n_\phi(k)$ is determined. As seen in Fig. 5, for a surface $\rho = \rho_k$ the same number of nodes are distributed on the curves associated with ϕ_j's. This again causes the mesh points cluster when the perimeter of the curves becomes small. In order to find a suitable number $n_\theta(j,k)$, we consider the uniform mesh in Ω_c with the revised number of $n_\phi(k)$ in stage 2. The perimeter of the curves ϕ_j's can be used for this purpose (see Fig. 5). Let $p(j,k)$ represents the perimeter of the jth curve on the kth surface, then we have

$$p(j,k) = \sum_{i=0}^{n_\theta - 1} l_{ijk}, \qquad j = 0, 1, \ldots, n_\phi(k),$$

where

$$l_{ijk} = \sqrt{\left(x_{i+1\,jk} - x_{ijk}\right)^2 + \left(y_{i+1\,jk} - y_{ijk}\right)^2 + \left(z_{i+1\,jk} - z_{ijk}\right)^2},$$
$$i = 0, 1, \ldots, n_\theta - 1.$$

Now let p_0 denotes the maximum values of the perimeters and N is the number of the steps along the curve associated with p_0, then a reasonable value for $n_\theta(j,k)$ is a number which satisfies

$$\frac{n_\theta(j,k)}{N} = \frac{p(j,k)}{p_0} \quad \text{or} \quad n_\theta(j,k) = N.\frac{p(j,k)}{p_0}\,, \quad j = 0, 1, \ldots, n_\phi(k). \tag{12}$$

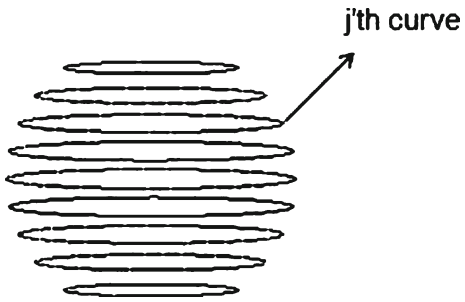

Fig. 5 The curves associated with ϕ_j's

On the other hand, the values of $n_\theta(j,k)$ must satisfy

$$\sum_{j=0}^{n_\phi(k)} n_\theta(j,k) = n_\rho(k), \tag{13}$$

where $n_\rho(k)$ previously evaluated in (9) is the total number of nodes on $\rho = \rho_k$. Substituting (12) in (13) yields

$$\frac{N}{p_0}\sum_{j=0}^{n_\phi(k)} p(j,k) = n_\rho(k) \quad \text{or} \quad N = n_\rho(k) \times \frac{p_0}{\sum_{j=0}^{n_\phi(k)} p(j,k)}. \tag{14}$$

Therefore, having obtained the value of N, $n_\theta(j,k)$ can be provided by applying (12). However, N itself depends on $n_\rho(k)$, the total number of nodes on the kth surface, and (9) states that this number is dependent on M, the total number of nodes on the boundary, which is still unknown. In practice, M can be constructed through the refinement process. We propose the following algorithm in order to produce the uniform mesh points:

Algorithm 1 Constructing a uniform mesh

For given values of n_θ, n_ϕ and n_ρ

1. $n_\theta(j,n_\rho)$, $j = 1,\cdots,n_\phi$, the appropriate number of divisions of θ for each curve associated with ϕ_j, on the boundary are determined using (12) for $k = n_\rho$ in stage 3. Note that $N = n_\theta$. Having obtained the $n_\theta(j,n_\rho)$, M, the total number of points on the boundary, can be derived by
$$M = \sum_{j=0}^{n_\phi} n_\theta(j,n_\rho).$$
2. Having obtained M, $n_\rho(k)$, $k = 1,\cdots,n_\rho - 1$, can be calculated using (9).
3. Having obtained $n_\rho(k)$, for each surface k, $k = 1,\cdots,n_\rho - 1$, the suitable partition of ϕ, $n_\phi(k)$, can be evaluated using (11).
4. Having prepared $n_\rho(k)$ and $n_\phi(k)$, $k = 1,\cdots,n_\rho - 1$, $n_\theta(k,j)$ for $j = 1,\cdots,n_\phi(k)$ can be derived using (12).

A set of roughly uniform mesh points obtained from Fig. 2 after refinement is displayed in Fig. 6.

3.2 Adapting the nodes

We now describe the adapting method. We start with the uniform mesh (8) in Ω_c where Ω_c is the cube introduced in (3). First, an adaptive mesh is made in the cube employing the technique presented in Section 2.1 and then the mesh points produced are transformed into the physical domain Ω_p using the mapping (4). A set of adaptive nodes in an ellipsoid for the solution function $u = x^2 + y^2 + z^2$ is shown in Fig. 7. As depicted in the figure, the

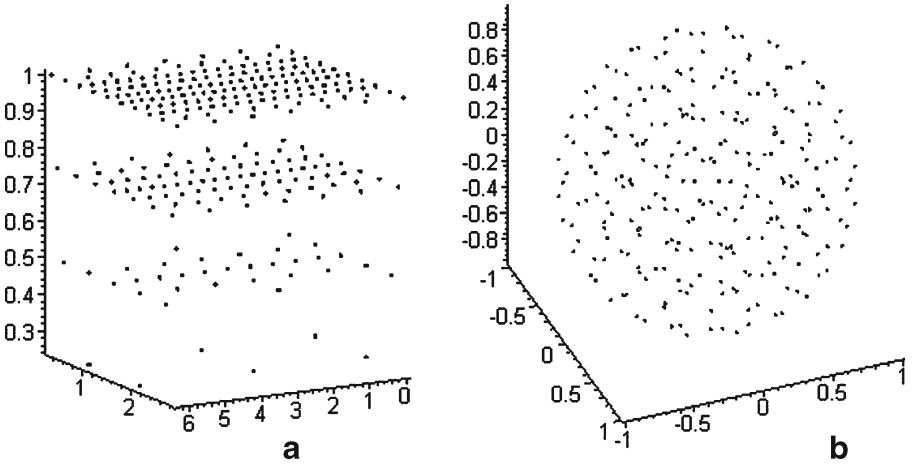

Fig. 6 A set of roughly uniform mesh points obtained from the nodes in Fig. 2 after refinement

mesh points are clustered in some parts of the domain. This is due to the issue highlighted in Section 3. In order to have some adaptive nodes smoothly distributed, the adapting method needs to be combined with the refinement process described in Section 3.1 for constructing a uniform mesh. In fact the same number of divisions, $n_\rho(k)$, $n_\phi(k)$ and $n_\theta(j,k)$ obtained in Section 3.1 are suitable for this purpose. The key point to our proposed method is to distribute an appropriate number of adaptive nodes a long the grid curves produced in the adapting method (see Fig. 1). More precisely, for each surface $\rho = \rho_k,\ k = 1, \cdots, n_\rho$, which has been resulted from the initial plane $\rho = \rho_k$

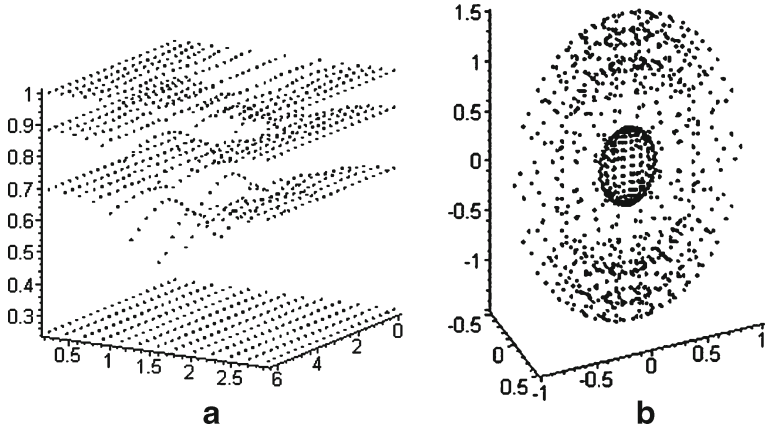

Fig. 7 A set of adaptive nodes produced by the new method without refinement in Ω_c and Ω_p are displayed, respectively in (**a**) and (**b**)

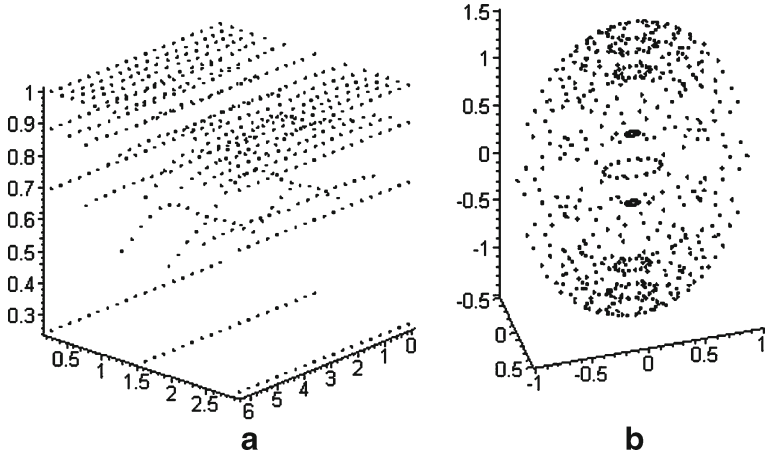

Fig. 8 The adaptive nodes after refining the ϕ partition

based on the adapting method, $n_\phi(k)$ adaptive nodes are distributed along the associated grid curves formed by the four stages of the adapting methods (Fig. 8). The final refinement is related to the partition of the coordinate θ based on the appropriate number $n_\theta(j,k)$. To this end, for $k = 1, \cdots, n_\rho$ and $j = 0, \cdots, n_\phi(k)$, $n_\theta(j,k)$ adaptive nodes are distributed along the respective grid curves constructed by the four stages of the method and the above refinement of the ϕ partition (Fig. 9).

Another example of the adaptive points produced by the new method is given in Fig. 10 for two solution functions $u = e^{4-x^2-y^2-z^2}$ and $u = e^{x^2+y^2+z^2}$ in an ellipsoid. As expected, for the former case, the mesh points are more dense

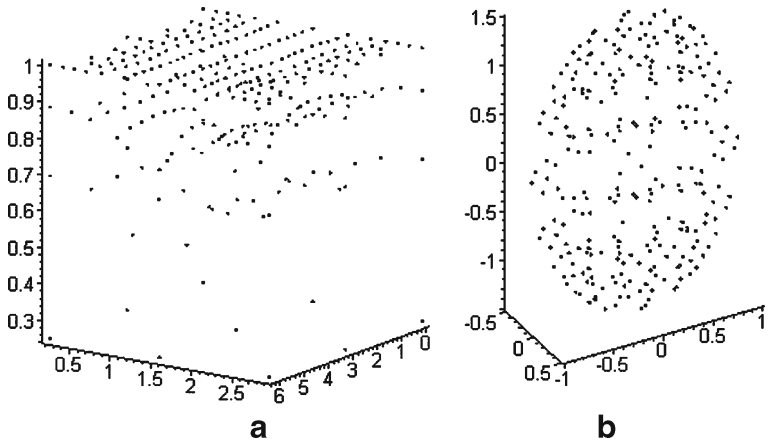

Fig. 9 The adaptive nodes after the final refinement

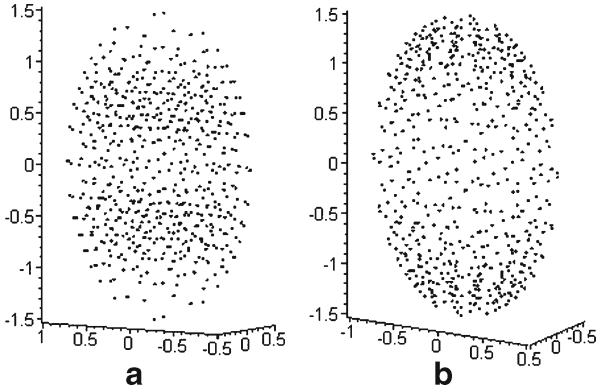

Fig. 10 Adaptive nodes produced for the solution functions $u = e^{4-x^2-y^2-z^2}$ and $u = e^{x^2+y^2+z^2}$ in an ellipsoid are displayed, respectively, in (**a**) and (**b**)

around the center and, for the latter case, the nodes are more dense close to the boundary.

4 Collocation meshless method

In this Section we first introduce the RBFs and then describe their application to the numerical solution of PDEs based on the collocation method.

4.1 Radial basis functions

RBFs are known as the natural extensions of splines to multi-variate interpolation. Suppose the set of points

$$\{x_i \in \Omega | i = 1, 2, \cdots, N\},$$

is given, where Ω is a bounded domain in R^n. The radial function $\varphi : \Omega \longrightarrow R$ is used to construct the approximate function

$$s(x) = \sum_{k=1}^{N} \alpha_k \varphi(\|x - x_k\|),$$

which interpolates an unknown function f whose values at $\{x_i\}_{i=1}^N$ are known. $\|.\|$ represents the Euclidean norm. The unknown coefficients α_k are determined such that the following N interpolation conditions are satisfied,

$$f(x_i) = s(x_i) = \sum_{k=1}^{N} \alpha_k \varphi(\|x_i - x_k\|), \quad i = 1, \cdots, N.$$

There is a large class of interpolating RBFs [19] that can be used in meshless methods. These include the linear $1 + r$, the polynomial $P_k(r)$, the thin plate spline (TPS) $r^2 \log r$, the Gaussian $\exp(-r^2/\beta^2)$, and the multi-quadrics $\sqrt{\beta^2 + r^2}$ (with β a constant parameter) [13]. In this paper we shall mainly use

the RBF r^5 which is a particular case of r^{2k+1} (k=2), known as the generalized linear RBF. The theoretical discussion of this RBF has been given in [10].

4.2 Collocation method

We describe the collocation method for a general case of PDEs in the form

$$Lu = F, \tag{15}$$

where $L = [L_1, \cdots, L_N]^T$ represents a vector of linear operations and $F = [f_1, \cdots, f_N]^T$ denotes a vector containing the right hand sides of the equations. For instance, Poisson's equation with a Dirichlet boundary condition

$$\begin{aligned} \Delta u &= f, \quad \text{in } \Omega, \\ u &= g, \text{ on } \partial\Omega, \end{aligned}$$

is a very simple case of (15) where $L = [\Delta, I]^T$, $F = [f, g]^T$ and the operators Δ and I act on the domain Ω and the boundary $\partial\Omega$ respectively.
The collocation method is simply to express the unknown function u in terms of the $RBFs$ as

$$u(x) = \sum_{k=1}^{N} \alpha_k \varphi(\|x - x_k\|), \tag{16}$$

and determine the unknowns α_k in such a way that (16) satisfies (15) for all interpolation points. Substituting (16) in (15) and imposing the N essential conditions of the collocation method lead to a linear system of equations whose coefficient matrix consists of N row blocks, the entries of which are of the form

$$A_{ij}^{\mu} = L_\mu \varphi(\|x - x_j\|)|_{x=x_i}, \quad i = 1, \cdots, N_\mu, \ j = 1, \cdots, N,$$

where N_μ indicates the number of nodes associated with the operator L_μ (see [11]).

4.3 Implementation

We now discuss the practical aspects of the adaptive nodes method in connection with the meshless method. The first point is how the monitor function is evaluated? We explain the process for the case of the arc-length monitor. In order to equi-distribute the mesh points in the three coordinate directions, θ, ϕ and ρ, the monitors

$$M_\theta = \sqrt{1 + u_\theta^2}, \quad M_\phi = \sqrt{1 + u_\phi^2}, \quad M_\rho = \sqrt{1 + u_\rho^2},$$

are required where

$$u_\theta = \frac{\partial u}{\partial x}\frac{dx}{d\theta} + \frac{\partial u}{\partial y}\frac{dy}{d\theta} + \frac{\partial u}{\partial z}\frac{dz}{d\theta}, \quad u_\phi = \frac{\partial u}{\partial x}\frac{dx}{d\phi} + \frac{\partial u}{\partial y}\frac{dy}{d\phi} + \frac{\partial u}{\partial z}\frac{dz}{d\phi},$$

$$u_\rho = \frac{\partial u}{\partial x}\frac{dx}{d\rho} + \frac{\partial u}{\partial y}\frac{dy}{d\rho} + \frac{\partial u}{\partial z}\frac{dz}{d\rho}.$$

In the above expressions, we approximate the partial derivatives of u by the symmetric three-point finite difference formula using a primary solution obtained by a set of uniformly distributed points. The other derivative terms can be exactly obtained using (5).

The application of the adaptation method to the meshless method is summarized in the following algorithm.

Algorithm 2 Adaptive meshless method

For a given PDE and a domain Ω_p with boundary Γ determined by (2),

1. Get the values n_θ, n_ϕ and n_ρ.
2. Produce a uniform mesh in Ω_p and apply the collocation meshless method to obtain a primary approximate solution.
3. Produce a uniform mesh in the cube Ω_c in the cartesian system $\rho\theta\phi$ using n_θ, n_ϕ and n_ρ as the numbers of points in each coordinate direction.
4. For $k = 1, \cdots, n_\rho - 1$, evaluate $n_\rho(k)$ and $n_\phi(k)$ by Algorithm 1.
5. For $k = 1, \cdots, n_\rho - 1$, evaluate $n_\theta(k, j)$, $j = 1, \cdots, n_\phi(k)$ by Algorithm 1.
6. Use the solution of step 2 to approximate the derivatives at the nodal points and evaluate the monitor function at these points.
7. Start with the uniform mesh of step 3 and use the four-stage algorithm of Section 2.1 to obtain the adaptive mesh and the grid curves generated in the adaptation process.
8. For each surface $\rho = \rho_k$, $k = 1, \cdots, n_\rho$, distribute $n_\phi(k)$ adaptive nodes along the associated grid curves derived in step 7.
9. For $k = 1, \cdots, n_\rho$ and $j = 0, \cdots, n_\phi(k)$, distribute $n_\theta(j, k)$ adaptive nodes along the respective grid curves associated with θ.
10. Transform the mesh points produced in steps 8 and 9 into Ω_p using mapping (4) and obtain the final adaptive nodes.
11. Apply the collocation meshless method to the underlying PDE using the nodes generated in step 10.

5 Numerical results

We now solve some PDEs by the collocation meshless method to show the effectiveness of the new mesh generation approach. In each case, a PDE is considered with (i) some equally spaced nodes and (ii) adaptive nodes and the numerical errors are compared. As previously noted, $\phi(r) = r^5$ is used as a basis function. In each example, M test points, which do not coincide with the interpolation nodes, are randomly selected and a root mean square (RMS) error at these points is evaluated by

$$\text{RMS error} = \sqrt{\sum_{i=1}^{M} \left(u_{apr,i} - u_{ex,i}\right)^2 / M},$$

where $u_{apr,i}$ and $u_{ex,i}$ denote the approximate and exact values of u, respectively, at a test point i.

We consider the Poisson equation

$$\begin{aligned} \Delta u &= f(x, y, z) \quad \text{in } \Omega, \\ u(x, y, z) &= g(x, y, z) \quad \text{on } \partial\Omega, \end{aligned} \tag{17}$$

for three different cases with various solutions and regions.

Example 1

$$f(x, y, z) = 6, \quad g(x, y, z) = x^2 + y^2 + z^2,$$

Ω is a region bounded by the ellipsoid $16x^2 + 4y^2 + 16/9z^2 = 1$. The exact solution for the above equation is given by $u(x, y, z) = x^2 + y^2 + z^2$.

Example 2

$$f(x, y, z) = 6e^{x^2+y^2+z^2} + 4\left(x^2 + y^2 + z^2\right) e^{x^2+y^2+z^2}$$

$$g(x, y, z) = e^{x^2+y^2+z^2}$$

and the boundary is given by the parametric equations:

$$\begin{cases} x = \frac{1}{4}\sin(\phi)\cos(\theta) \\ y = \frac{1}{2}\sin(\phi)\sin(\theta) \\ z = \frac{3}{4}\cos(\phi) \end{cases}$$

The exact solution in this case is given by $u(x, y, z) = e^{x^2+y^2+z^2}$.

Example 3 As the last example, we consider (17) in a sphere in the case of

$$g(x, y, z) = \operatorname{sech}\left(50\left((x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 - 0.3^2\right)\right),$$

and the function $f(x, y, z)$ is determined such that the function

$$u(x, y, z) = \operatorname{sech}\left(50\left((x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 - 0.3^2\right)\right),$$

is the exact solution.

The mesh points generated by the new method for the above solutions of Examples 1, 2 and 3 are, respectively, displayed in Figs. 9b, 11 and 12. As observed in Fig. 9b, more mesh points have been allocated to the area where the solution function has more variation. A similar situation is observed in Fig. 11 for solution of Example 2. The solution of Example 3 has a large variation across the sphere $(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 = 0.3^2$. This can be observed in the produced adaptive nodes in which the mesh points are dense around the above sphere.

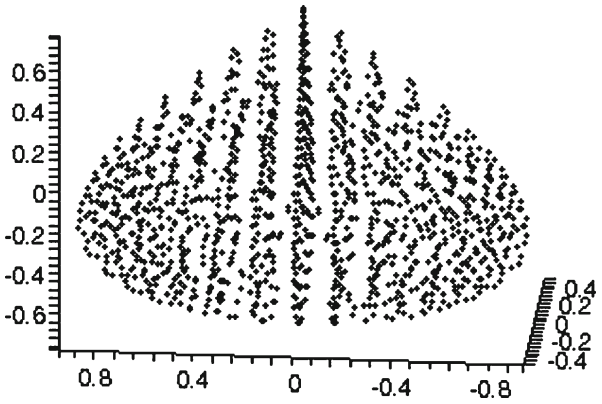

Fig. 11 Adaptive nodes produced for the solution of Example 2

The collocation meshless method was applied to the above equation using the adaptive nodes generated in three cases: (I) Arc-length monitor, (II) $M = (u^{''})^{2/5}$ and (III) $M = [u^{'''2} + u^{''''2} + u^{'''''2}]^{1/7}$ referred to as Monitor 1, Monitor 2 and Monitor 3, respectively, in the Tables. The error values for Examples 1 and 2 are listed in Tables 1 and 2, respectively for various number of nodes, n and different monitors. In most cases, the results demonstrate a considerable improvement in the accuracy. In particular, for Example 1, the error in the case of using 436 adaptive nodes is less than that in the case of using 1360 uniform mesh points. We also observe that the most accurate solution is obtained in the case of using an arc-length monitor. Moreover, Monitor 3 generally acts better than Monitor 2. In particular, in the case of

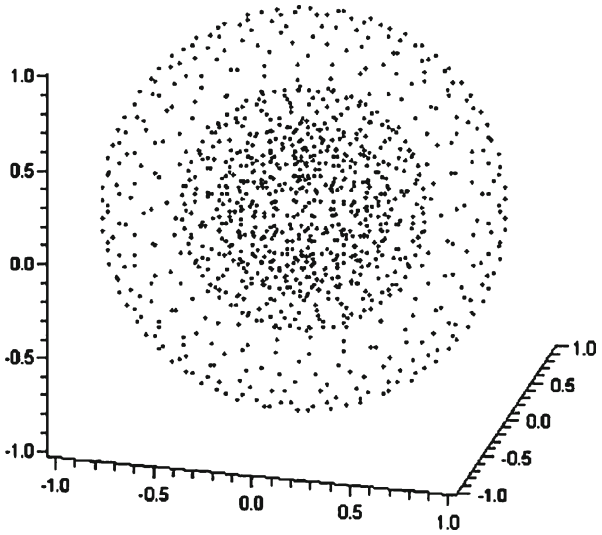

Fig. 12 Adaptive nodes produced for the solution of Example 3

Table 1 Error values for Example 1

n_θ, n_ϕ, n_ρ n	21,14,4 288	23,15,5 384	24,16,5 436	26,17,6 590	30,20,7 934	32,22,7 1,126	33,23,7 1,202	34,23,8 1,360
Uniform	9.34E-5	7.19E-5	5.47E-5	1.95E-5	7.81E-6	5.80E-6	4.07E-6	2.381E-6
Monitor 1	2.05E-6	5.21E-6	1.77E-6	2.53E-7	4.71E-7	2.24E-7	1.90E-7	5.05E-8
Monitor 2	8.43E-6	8.25E-6	4.47E-6	2.00E-6	1.47E-6	1.07E-6	1.15E-6	3.54E-7
Monitor 3	9.36E-6	1.27E-6	2.25E-6	1.05E-6	1.32E-6	3.87E-7	3.13E-7	2.26E-7

Example 2, whose solution has a much larger variation than that of Example 1, the accuracy, in the case of Monitor 2 is even less than that in the case of equally spaced nodes.

The error values for Example 3 in the case of using the arc-length monitor is given in Table 3 for various number of nodes, n. The solution of this example has a significantly larger variation than those of the previous examples. This seriously affects the accuracy in the cases of both uniform and the adaptive nodes. However, the adaptation method demonstrates some improvement in the results.

5.1 A general irregular region

In Section 3, we assumed analytical equations for the boundary, in order to explain the transformation used in the adaptation process. Here we propose a linear piecewise interpolation to approximate the functions $g_i(\theta, \phi),\ i = 1, 2, 3$ in (2) for a general case of an irregular domain. Let $\{(x_j, y_j, z_j)|\ j = 1, 2, \cdots, N\}$, be a set of points distributed on the boundary. Using these points, we construct a set of triangular elements covering whole the boundary. Let (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3) be the vertexes of an element. Using the relations between the spherical and cartesian coordinates, we find the spherical coordinates of these points. Suppose that $(\theta_1, \phi_1, \rho_1)$, $(\theta_2, \phi_2, \rho_2)$ and $(\theta_3, \phi_3, \rho_3)$ are the spherical coordinates corresponding to the above three points. Now the function $x = g_1(\theta, \phi)$ is a surface in the cartesian coordinate (θ, ϕ, x) which can be approximated in the triangle by a linear piecewise interpolation $x = \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_3$, where $\phi_i,\ i = 1, 2, 3$ are the appropriate basis functions used in the classical finite element. Other functions, $y = g_2(\theta, \phi)$ and $z = g_3(\theta, \phi)$ can be approximated in a similar way.

Table 2 Error values for Example 2

n_θ, n_ϕ, n_ρ n	31,14,4 420	32,14,5 500	35,16,5 628	45,21,7 1,372
Uniform	8.65E-4	7.96E-4	4.73E-4	1.47E-4
Monitor 1	3.24E-4	6.37E-5	3.38E-5	9.77E-6
Monitor 2	4.07E-2	4.13E-2	4.12E-2	4.13E-2
Monitor 3	3.42E-4	9.67E-5	4.01E-5	9.78E-6

Table 3 Error values for Example 3

n_θ, n_ϕ, n_ρ	16-10-3	17-11-4	18-11-4
n	126	172	180
Uniform	7.41E-1	7.59E-1	6.89E-1
Adaptive	8.23E-1	4.97E-2	4.36E-2

6 Conclusion

An adaptive nodes method based on an equi-distribution property was developed for irregular domains in the case of 3D. The method was based on transforming the physical domain to a cube and generating some adaptive nodes in the cube using dimension reduction and equi-distributing principle. The adapting process in the cube was accomplished by equi-distributing along the grid curves in the three coordinate directions using a previous work. The adaptive nodes produced was transformed into the physical domain. The proposed method offers a set of 3D scattered points suitable for the meshless-type methods in which the connectivity of the points is not used and the smoothness of the mesh is not required.

The new method was examined by considering some Poisson equations solved by a collocation meshless method and the results demonstrated considerable reduction in the error values. Since the current work was not involved with constructing a monitor for the underlying method, some well known monitors were employed to produce the adaptive nodes and the best accuracy was achieved in the case of using the arc-length monitor. Of course, using a suitable monitor for the underlying method could have resulted in more improvement in the numerical results.

In addition to improving the computational efficiency, the proposed method can be a treatment for ill-conditioning of the RBFs due to the fact that it reduces the total number of the nodes. Moreover, applying the new adaptive nodes technique to the methods such as domain decomposition and locally supported RBFs, previously suggested to treat ill-conditioning, is recommended and left for a future work.

The new method was implemented for regions whose boundary was given by parametric equations in terms of spherical coordinates. This method can be also applied to more general cases. We suggested a linear piecewise interpolation, in terms of some scattered points on the boundary, to make a set of parametric equations.

References

1. Beckett, G., Mackenzie, J.A., Ramage, A., Sloan, D.M.: On the numerical solution of one-dimensional PDEs using adaptive methods based on equi-distribution. J. Comput. Phys. **167**(2), 372–392 (2001)
2. Choi, Y., Know, K.Y., Chae, S.W., Kim, D.M.: A modified advancing layers mesh generation for thin three-dimensional objects with variable thickness. Comput. Graph. **33**(3), 195–203 (2009)

3. Cao, W., Huang, W., Russell, R.D.: A study of monitor functions for two-dimensional adaptive mesh generation. SIAM J. Sci. Comput. **20**(6), 1978–1994 (1999)
4. Carey, G.F., Dinh, H.T.: Grading functions and mesh redistribution. SIAM J. Numer. Anal. **22**(5), 1028–1040 (1985)
5. Carey, G.F., Humphrey, D.L.: Mesh refinement and iterative solution methods for finite element computations. Int. J. Numer. Methods Eng. **17**(11), 1717–1734 (2005)
6. Chen, C.S., Ganesh, M., Golberg, M.A., Cheng, A.H.D.: Multi-level compact radial functions based computational schemes for some elliptic problems. Comput. Math. Appl. **43**(3), 359–378 (2002)
7. Chen, K.: Two-dimensional adaptive quadrilateral mesh generation. Commun. Numer. Methods Eng. **10**(10), 815–825 (1994)
8. Chen, K.: Error equidistribution and mesh adaptation. SIAM J. Sci. Comput. **15**(4), 798–818 (1994)
9. De Boor, C.: In good approximation by splines with variable knots II. Lecture Notes Series, vol. 363. Springer, Berlin (1973)
10. Duchon, J.: Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In: Schempp, W., Zeller, K. (eds.) Constructive Theory of Function of Several Variables. Lecture Notes in Mathematics, vol. 571, pp. 85–100. Springer, Berlin (1977)
11. Fasshauer, G.E.: Solving partial differential equations with radial basis function: multi-level methods and smoothing. Adv. Comput. Math. **11**, 139–159 (1999)
12. Fasshauer, G.E.: Newton iteration with multiquadrics for the solution of PDEs. Comput. Math. Appl. **43**, 423–438 (2002)
13. Hardy, R.L.: Theory and applications of the multiquadrics-biharmonic method: 20 years of discovery. Comput. Math. Appl. **19**(8/9), 163–208 (1990)
14. Kansa, E.J.: Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics- part I. Comput. Math. Appl. **19**(8/9), 147–161 (1990)
15. Kansa, E.J.: Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics–part II. Comput. Math. Appl. **19**(8/9), 147–161 (1990)
16. Kansa, E.J., Hon, Y.C.: Circunventing the ill-conditioning problem with multiquadric radial basis functions. Comput. Math. Appl. **39**(7/8), 123–137 (2000)
17. Kautsky, J., Nichols, N.K.: Equi-distributing meshes with constraints. SIAM J. Sci. Statist. Comput. **1**(4), 449–511 (1980)
18. Kopteva, N., Madden, N., Stynes, M.: Grid equidistribution for reactiondiffusion problems in one dimension. Numer. Algorithms **40**, 305–322 (2005)
19. Powell, M.J.D.: The theory of radial basis function approximation in 1990. In: Light, W.A. (ed.) Advances in Numerical Analysis, vol. 2: Wavelets, Subdivision-algorithms and Radial Basis Functions, pp. 105–210. Oxford Univ. Press (1992)
20. Shanazari, K., Chen, K.: An overlapping domain decomposition for the dual reciprocity method. Eng. Anal. Bound. Elem. **27**, 945–953 (2003)
21. Shanazari, K., Chen, K.: A minimal distance constrained adaptive mesh algorithm with application to the dual reciprocity method. Numer. Algorithms **32**, 275–286 (2003)
22. Shanazari, K., Fallahi, M.: A quasi-linear method applied to the method of fundamental solution. Eng. Anal. Bound. Elem. **34**, 388–392 (2010)
23. Shanazari, K., Hosami, M.: A two dimensional adaptive nodes technique in irregular regions applied to meshless-type methods. Eng. Anal. Bound. Elem. **36**, 161–168 (2012)
24. Shanazari, K., Rabie, N.: A three dimensional adaptive nodes technique applied to meshless-type methods. Appl. Numer. Math. **59**, 1187–1197 (2009)
25. Shishkina, O., Wagner, C.: Adaptive meshes for simulations of turbulent Rayleigh-Benard convection. J. Numer. Anal. Indust. Appl. Math. **1**(2), 219–228 (2006)
26. Sweby, P.K.: Data-dependent grids. Numerical Analysis Report 7/87, University of Reading, UK (1987)
27. Wong, H., Qin, Q.H.: A meshless method for generalized linear or nonlinear Poisson-type problems. Eng. Anal. Bound. Elem. **30**(6), 515–521 (2006)
28. Wong, S.M., Hon, Y.C., Li, T.S.: A meshless multilayer model for a coastal system by radial basis functions. Comput. Math. Appl. **43**, 585–605 (2002)
29. White, A.B.: On selection of equi-distribution meshes for two-point boundary value problems. SIAM J. Numer. Anal. **16**(3), 472–502 (1979)
30. Winslow, A.M.: Numerical solution of the quasi-linear Poisson's equation in a non-uniform triangle mesh. J. Comput. Phys. **1**(2), 149–172 (1967)