

## Research Article

# Redistribution of Nodes with Two Constraints in Meshless Method of Line to Time-Dependent Partial Differential Equations

**Jafar Biazar and Mohammad Hosami**

*Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 41335-1914, Rasht 4193822697, Iran*

Correspondence should be addressed to Jafar Biazar; [biazar@guilan.ac.ir](mailto:biazar@guilan.ac.ir)

Received 6 July 2015; Accepted 19 October 2015

Academic Editor: Patricia J. Y. Wong

Copyright © 2015 J. Biazar and M. Hosami. 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.

Meshless method of line is a powerful device to solve time-dependent partial differential equations. In integrating step, choosing a suitable set of points, such as adaptive nodes in spatial domain, can be useful, although in some cases this can cause ill-conditioning. In this paper, to produce smooth adaptive points in each step of the method, two constraints are enforced in Equidistribution algorithm. These constraints lead to two different meshes known as quasi-uniform and locally bounded meshes. These avoid the ill-conditioning in applying radial basis functions. Moreover, to generate more smooth adaptive meshes another modification is investigated, such as using modified arc-length monitor function in Equidistribution algorithm. Influence of them in growing the accuracy is investigated by some numerical examples. The results of consideration of two constraints are compared with each other and also with uniform meshes.

## 1. Introduction

Meshless methods have gained much attention in the last decade. They are well known for their simplicity and ability in reconstructing multivariate functions from scattered data. Moreover, meshless methods using radial basis functions (RBFs) are powerful methods to solve partial differential equations (PDEs). The initial development in applying RBFs for numerical solution of PDEs is due to the pioneering work of Kansa [1, 2]. He used some collocation nodes to collocate the RBFs. Meshless methods using RBFs have several advantages comparing to finite difference method (FDM) finite element method (FEM) and other mesh based methods [3]. One of these advantages is that they do not require a mesh or element. These methods need only some scattered nodes. It means that the nodes can be chosen, freely. Due to this useful property, an important geometric problem arises: how to choose the nodes to improve the accuracy? This problem causes too many researches about distribution

of nodes in meshless methods. Several researchers have considered this problem [4–8]. One of the effective methods to choose an efficient set of central nodes, known as adaptive nodes, is Equidistribution method [9, 10]. In this method, the objective is to find a partition of the interval, such that a given weight function takes a given constant value over each subinterval. These adaptive central nodes can be used in meshless methods, such as meshless method of line (MMOL). The method of line (MOL) is a well-known numerical method to solve PDEs [11, 12]. In the meshless method of line, the RBFs are used to approximate the solution in MOL. This method is very reliable for using adaptive nodes [13, 14]. In each step of this method, some central nodes, in spatial domain, are required. Adaptive central nodes can be a good selection to use in this method. But, due to ill-conditioning of the problem in some cases, when the nodes are near to each other, in many practical cases, it is necessary for the chosen nodes to have certain smoothness properties. This leads to some constraints.

In this study, two reliable constraints are presented, and the influence of them in applying the MMOL with adaptive nodes is investigated. The first constraint, known as quasi-uniform mesh, has been applied in some researches as well [13]. The aim of this study is to show application of another constraint, known as locally bounded mesh, and to discuss impact of it and some modification on meshes on increasing the accuracy. This paper is organized as follows. In Section 2, radial basis functions interpolation is introduced. In Section 3, an Equidistribution algorithm is presented and two constraints are imposed to obtain central nodes. In Section 4, adaptive nodes are applied in MMOL to solve some time-dependent PDEs.

## 2. Radial Basis Functions to Approximate a Function

Radial basis functions are real valued basis functions which depend on the distance  $r$  between two points. The commonly used RBFs are multiquadric (MQ), Gaussian, Thin-Plate Spline (TPS), and compactly supported RBFs (CS-RBFs). The MQ radial basis function provides the most accurate approximation, in most applications of RBFs [15]. MQ is defined as  $\varphi(r) = \sqrt{1 + (\varepsilon r)^2}$ , where  $\varepsilon$  is called shape parameter. In this paper, MQ is used in numerical examples. To approximate a given scattered data  $\{f_i\}_{i=1}^n$  at nodes  $\{x_i\}_{i=1}^n$ , RBF interpolation is given by combination of  $n$  RBFs; that is,

$$u(x) = \sum_{j=1}^n \lambda_j \varphi(r_j), \quad \lambda_j \in \mathbb{R}, \quad (1)$$

where  $r_j = \|x - x_j\|_2$  and  $\|\cdot\|$  denote the Euclidean norm.  $\varphi(r)$  is an RBF and  $\lambda_j$ s are the coefficients that will be determined. By collocating the interpolation conditions ( $u(x_j) = f_j, j = 1, \dots, n$ ), the system of equations is obtained as the following matrix form:

$$A\lambda = f, \quad (2)$$

where  $a_{ij} = \varphi(r_{ij})$  and  $r_{ij} = \|x_i - x_j\|_2$ .

The accuracy of the approximate function depends on various factors. Some of the most important ones are as follows: how the RBFs are chosen, nodes distribution, and selecting the shape parameter. Finding the optimal shape parameter is an open problem, although concentrated researches have been made to determine some appropriate shape parameter for a given problem [16, 17]. In this paper, we focus on distributing the nodes to obtain more accurate approximations. In next section, an Equidistribution method is introduced to select adaptive central nodes.

## 3. An Adaptive Method to Central Nodes

Based on the mesh-free property of RBF meshless methods, one can select a set of nodes freely, such as uniform or random scheme. But in the case that the solution is relatively more oscillatory or even shocks appear, some adaptive schemes can be applied [4–8]. Equidistribution algorithm is a reliable

approach to construct adaptive central nodes. Equidistribution is the process of distributing the nodes in an interval such that a determined weight function is equally distributed over the chosen mesh.

### 3.1. Equidistribution

*Definition 1* (Equidistribution). Let  $M$  be a nonnegative piecewise continuous function ( $M \in C^+[a, b]$ ) and  $c$  a constant such that  $n = (1/c) \int_a^b M(x)dx$  is an integer. The mesh

$$\Delta : \{x_1, x_2, \dots, x_n\} \quad (3)$$

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

$$\int_{x_{i-1}}^{x_i} M(x) dx = c, \quad j = 2, \dots, n. \quad (4)$$

The function  $M$  is called “monitor,” which depends on the underlying function  $u$ . To find more details about the monitors refer to [18, 19]. In this paper the arc-length monitor ( $M(x) = \sqrt{1 + u_x^2}$ ) is applied. To enforce at least a few nodes in the flat part of the interval, a parameter  $\alpha$  can be inserted in the arc-length monitor, that is, the modified arc-length  $M(x) = \sqrt{\alpha + u_x^2}$ . In numerical examples the influence of this parameter is illustrated.

For a given monitor function  $M$  and the constant  $c$ , the Equidistribution to produce an e.d. mesh in  $[a, b]$  is done in three steps.

*Step 1.* Approximate  $g(t) = \int_a^t M(\tau)d\tau, t \in [a, b]$ .

*Step 2.* Determine the smallest integer  $n$  such that  $nc \geq g(b)$  and define  $d = g(b)/n$ .

*Step 3.* Find the mesh

$$\Pi_n : a = x_0 < \dots < x_n = b \quad (5)$$

by inverse interpolation. The points  $x_j$  are given by  $x_j = g^{-1}(jd)$ .

Figure 1 shows the e.d. nodes based on  $u(x) = e^{x^4}$  with  $\alpha = 1$ . It shows that central nodes are concentrated at the region with steepest gradient. The concentration makes the nodes in the region near to each other. With smaller minimum distance between centrals, the MQ shape parameter must be adjusted, so that the condition number of the associated linear system remains reasonable. This adjustment is not always applicable. Thus some constraint on the distribution of central nodes can be imposed. The two most common constraints and an algorithm to construct these constraint adaptive nodes are introduced in the following section.

*3.2. Equidistribution with Constraints.* The two most important and common constraints are quasi-uniform and locally bounded. Equidistributing a function subject to the first constraint has been introduced in Pereyra and Sewell’s work in 1974 [9]. The second constraint that is more important is introduced by Kautsky and Nichols at 1980 [10].

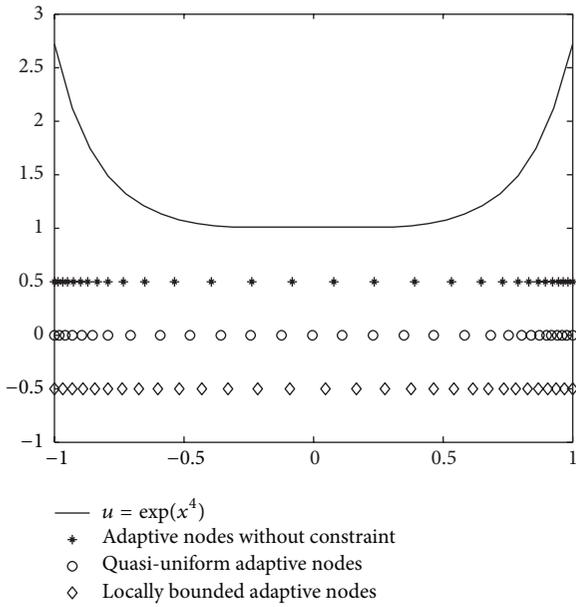


FIGURE 1: The nodes produced with quasi-uniform and locally bounded constraints.

3.2.1. *Quasi-Uniform Mesh.* The mesh  $\Pi : \{x_1, x_2, \dots, x_n\}$  is called quasi-uniform with respect to the constant  $k$  if

$$\frac{\max_j h_j}{\min_j h_j} \leq k, \tag{6}$$

where  $h_j = x_{j+1} - x_j, j = 1, 2, \dots, n - 1$ .

*Definition 2* (sub-Equidistribution). According to Definition 1, the mesh  $\Pi$  is called subequidistributing (s.e.d.) on  $[a, b]$  with respect to  $M$  and  $c$  if, for  $nc \geq \int_a^b M$ ,

$$\int_{x_{j-1}}^{x_j} M(x) dx \leq c, \quad j = 2, \dots, n. \tag{7}$$

Now the problem is as follows: given a function  $M \in C^+$  and constants  $c > 0$  and  $k \geq 1$ , find a mesh which is

- (1) s.e.d. on  $[a, b]$  with respect to  $M$  and  $c$ ,
- (2) quasi-uniform with respect to  $k$ .

The following theorem solves the problem.

**Theorem 3.** *If  $\Pi : \{a = x_1, x_2, \dots, x_n = b\}$  is an e.d. mesh on  $[a, b]$  with respect to  $M^*$  and  $d$ , where*

$$M^*(t) = \max(M(t), p), \tag{8}$$

with

$$p = \left(\frac{1}{k}\right) \max_{t \in [a,b]} M(t), \tag{9}$$

and  $d = (1/c) \int_a^b M^*(x) dx$  (and  $n$  is equal to the smallest integer such that  $nc \geq \int_a^b M^* dx$ ), then  $\Pi$  is a s.e.d. on  $[a, b]$  with respect to  $M$  and  $c$  and is satisfied in (6). For proof and more details about implementation refer to [9].

3.2.2. *Locally Bounded Mesh.* The mesh  $\Delta : \{x_1, x_2, \dots, x_n\}$  is locally bounded with respect to constant  $k \geq 1$ , if

$$\frac{1}{k} \leq \frac{h_j}{h_{j-1}} \leq k, \tag{10}$$

where  $h_j = x_{j+1} - x_j, j = 1, 2, \dots, n - 1$ .

The Equidistribution problem is as follows: given a function  $M \in C^+[a, b]$ , constants  $c > 0$  and  $k \geq 1$ , find a mesh which is

- (1) s.e.d. on  $[a, b]$  with respect to  $M$  and  $c$ ,
- (2) locally bounded with respect to  $k$ .

The following theorem solves the problem.

**Theorem 4.** *Let  $\lambda > 0, M \in C^+[a, b]$ , and*

$$M^*(t) = \max_{\tau \in [a,b]} G(M; t, \tau), \tag{11}$$

where  $G(M; t, \tau) = M(\tau)/(1 + \lambda|t - \tau|M(\tau))$ . If

$$\Pi : \{a = x_1, x_2, \dots, x_n = b\} \tag{12}$$

is an e.d. mesh on  $[a, b]$  with respect to  $M^*$  and some  $c > 0$ , then  $\Pi$  is s.e.d. on  $[a, b]$  with respect to  $M$  and  $c$ , and for  $k = e^{\lambda c}$  one has  $1/k \leq h_j/h_{j-1} \leq k, j = 2, \dots, n$ .

For proof and more details about implementation, interested reader is referred to [10].

3.2.3. *Numerical Algorithm.* Based on Theorems 3 and 4, it is now possible to build a mesh which is s.e.d. and is quasi-uniform or locally bounded. In practical application the function  $u$ , which is to be approximated, is generally given in discrete form; that is, we have the monitor function  $M$  at some points  $t_j$  on the mesh  $\Pi_N (\Pi_N : a = t_0 < \dots < t_N = b)$ . Based on Theorems 3 and 4, to find a s.e.d. mesh with respect to  $M$  and  $c$ , which is quasi-uniform or locally bounded with respect to  $k$ , we require three main steps.

*Step 1.* Pad the monitor function  $M$  to produce the padded function  $M^*$  using (8) and (11), respectively, for the quasi-uniform and locally bounded (with  $\lambda = (\ln k)/c$  for locally bounded).

*Step 2.* Determine the smallest number of nodes  $n$ , such that  $nc \geq \int_a^b M^* dx$ .

*Step 3.* Equidistribute  $M^*$  with respect to  $d = (\int_a^b M^* dx)/n$  to find the new mesh  $\Pi_n$ .

It is noted that as  $d \leq c$ , the mesh is constraint with respect to a constant  $k' \leq k$ , so that the number of points in the mesh may be greater than required to satisfy the constraints. In practice, the number of points  $N$  is constant, and the e.d. process is performed with respect to  $d = (\int_a^b M^* dx)/N$ . For deeper insight, the interested reader is referred to [9, 10].

Figure 1 shows the nodes produced with quasi-uniform and locally bounded constraints. It illustrates that the constrained meshes have more smoothness properties, than those produced without any constraint. Moreover, by modifying the value of  $k$  and  $\alpha$ , one can gain some arbitrary properties in the mesh. With smaller value of  $k$ , the mesh becomes near to uniform mesh so that with  $k = 1$  the two constraint meshes are same to uniform.

#### 4. Applying Adaptive Nodes in Meshless Method of Line

*4.1. Meshless Method of Line (MMOL).* Meshless method of line (MMOL) is a numerical method to approximate the solution of time-dependent PDEs. In MMOL, in each time step, the solution is approximated by radial basis functions.

Consider the following equation:

$$u_t + L(u) = 0, \quad (13)$$

with the boundary and initial conditions

$$\begin{aligned} u(a, t) &= f_1(t), \\ u(b, t) &= f_2(t), \\ t &> 0, \end{aligned} \quad (14)$$

$$u(x, t_0) = g(x), \quad a \leq x \leq b.$$

By time discretizing, the RBF approximation for the solution  $u$  in step  $n$  is given by

$$u(x, t_n) = u^n(x) = \sum_{j=1}^N \lambda_j \varphi(r_j) = \Phi^T(x) \lambda, \quad (15)$$

where  $\Phi^T(x) = [\varphi(r_1(x)), \varphi(r_2(x)), \dots, \varphi(r_N(x))]$  and  $r_j = \|x - x_j\|$ . Suppose that the spatial domain  $[a, b]$  is discretized by  $N$  nodes  $x_j$  ( $j = 1, 2, \dots, N$ ), such that  $x_1 = a$ ,  $x_N = b$ . By collocating (13) with the center nodes we have

$$u(x_i, t_n) = u^n(x_i) = \sum_{j=1}^N \lambda_j \varphi(r_{ij}), \quad i = 1, \dots, N. \quad (16)$$

These equations, in the matrix notation, are given by

$$u^{(n)} = A\lambda, \quad (17)$$

where  $u^{(n)} = [u^n(x_1), u^n(x_2), \dots, u^n(x_N)]^T$  and  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_N]^T$ . From (15) and (17),  $u^n(x)$  can be written as the following:

$$u^n(x) = \Phi^T(x) A^{-1} u^{(n)} = N(x) u^{(n)}, \quad (18)$$

where  $N(x) = \Phi^T(x) A^{-1} = [N_1(x), N_2(x), \dots, N_N(x)]$ .

Generally, for an arbitrary time  $t$ , we have

$$u^n(x; t) = N u^{(n)}. \quad (19)$$

Substituting (19) into (13), the following ordinary differential equation will be obtained:

$$\frac{du_i^n}{dt} + L(u_i^n) = 0, \quad i = 1, \dots, N. \quad (20)$$

To write (20) in the matrix form, let

$$\begin{aligned} U &= [u^n(x_1), u^n(x_2), \dots, u^n(x_N)]^T, \\ L(U) & \\ &= [L(u^n(x_1))^T, L(u^n(x_2))^T, \dots, L(u^n(x_N))^T]^T. \end{aligned} \quad (21)$$

Thus (20) can be written as follows:

$$\frac{dU}{dt} + L(U) = 0. \quad (22)$$

Equation (22) is a system of ordinary differential equations with the following initial and boundary conditions:

$$\begin{aligned} U(t_0) &= [u(x_1, t_0), u(x_2, t_0), \dots, u(x_N, t_0)]^T, \\ u(x_1, t) &= f_1(t), \\ u(x_N, t) &= f_2(t). \end{aligned} \quad (23)$$

This ODE can be solved by an ODE solver such as Runge-Kutta method. By applying fourth-order Runge-Kutta method (RK4) to (22), the following scheme will be obtained:

$$\begin{aligned} U^{n+1} &= U^n + \frac{k}{6} (K_1 + 2K_2 + 2K_3 + K_4), \\ K_1 &= -L(U^n), \\ K_2 &= -L\left(U^n + \frac{k}{2} K_1\right), \\ K_3 &= -L\left(U^n + \frac{k}{2} K_2\right), \\ K_4 &= -L(U^n + kK_3), \end{aligned} \quad (24)$$

where  $k$  is the time step. This scheme yields to the solution at each step of time.

*4.2. Adaptive Nodes in MMOL.* In this section adaptive meshless method of line is applied to time-dependent partial differential equation with initial conditions. In first step, adaptive nodes are produced in the spatial domain based on initial condition. Assume that  $s_j^n$  ( $j = 1, \dots, N$ ) is an approximate solution at time  $t^n$  at distinct nodes  $x_j^n$  ( $j = 1, \dots, N$ ). Then, MMOL is applied on these central nodes to obtain approximations  $\bar{s}_j^{n+1}$  ( $j = 1, \dots, N$ ) at time  $t^{n+1}$ . Next, a new set of adaptive nodes is obtained based on the properties of  $\bar{s}^{n+1}$  as a function. Finally,  $s_j^{n+1}$  ( $j = 1, \dots, N$ ) are obtained by interpolation of the values  $(x_j^n, \bar{s}_j^{n+1})$ . Application of adaptive central nodes in MMOL has been studied in some literatures [13, 14]. In producing the adaptive central nodes in

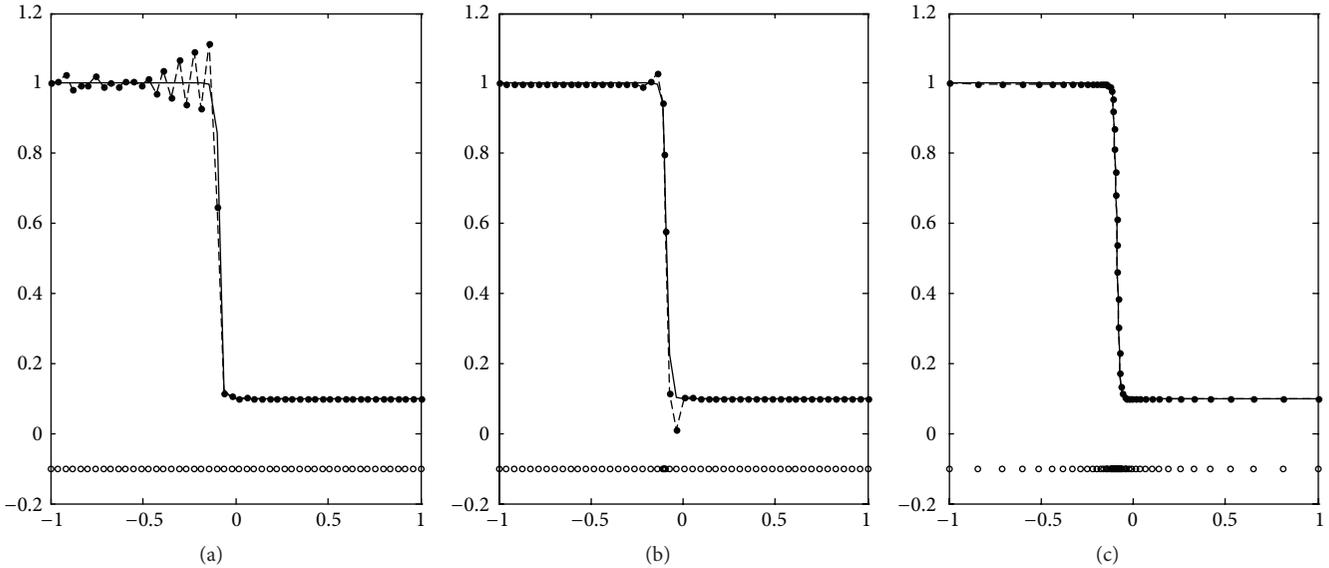


FIGURE 2: The plot of exact and approximate solutions of Example 5 with 50 uniform, quasi-uniform, and locally bounded nodes (resp., (a)–(c)).

TABLE 1: The results of Example 5 with different number of nodes and different mesh distribution.

Mesh	$t$	$n$	Shape	$k$	$\alpha$	$L_\infty$	$L_2$
Uniform	1	50	11	—	—	0.2142	0.0416
		100	11	—	—	0.0214	0.0038
		150	11	—	—	0.0020	$3.8450e - 4$
Quasi-uniform	1	50	51	6	$1e - 8$	0.1149	0.0234
		100	51	6	$1e - 8$	0.0049	0.0014
		150	61	3	$1e - 8$	0.0014	$5.2280e - 4$
Locally bounded	1	50	61	1.05	$1e - 16$	0.0060	0.0030
		100	21	1.001	$1e - 16$	0.0017	$5.6623e - 4$
		150	51	1.001	$1e - 16$	0.0018	$6.6146e - 4$

each step one can apply each of the constraints. In this study, we investigate the impact of two introduced constraints and the parameters  $\alpha$  and  $k$  to control the position of the central nodes. To compare the accuracy of the approximate solution, two well-known PDEs, Burgers equation and KdV equation, are investigated. Throughout the numerical examples, we use MQ-RBF.

*Example 5.* Consider Burger equation

$$u_t + uu_x = \nu u_{xx}, \tag{25}$$

on the interval  $[-1, 1]$ . The exact solution is

$$u(x, t) = \frac{0.1e^a + 0.5e^b + e^c}{e^a + e^b + e^c}, \tag{26}$$

where  $a = -(x+0.5+4.95t)/(20\nu)$ ,  $b = -(x+0.5+0.75t)/(4\nu)$ , and  $c = -(x + 0.625)/(2\nu)$ . The initial condition  $u(x, 0)$  and boundary conditions  $u(-1, t)$  and  $u(1, t)$  are determined using exact solution. By choosing  $\nu = 0.0035$  the equation is solved by using uniform nodes, quasi-uniform, and

locally bounded adaptive nodes. In Figure 2 the approximate solutions and the central nodes are shown. The values of parameters  $k$ ,  $\alpha$  and shape parameter  $c$ , the  $L_2$ -norms, and  $L_\infty$ -norms of the approximate solutions are presented in Table 1. Note that the impact of different values of  $k$  and  $\alpha$  make some freedom to avoid the ill-conditioning. The values of these parameters which depend on the number of nodes and their distribution form are shown in Table 1.

*Example 6.* Consider KdV equation:

$$u_t + \varepsilon uu_x + \mu u_{xxx} = 0, \tag{27}$$

with  $\varepsilon = 6$  and  $\mu = 1$ . The initial condition is

$$u(x, 0) = 2 \operatorname{sech}^2(x). \tag{28}$$

The exact solution is  $u(x, t) = 2 \operatorname{sech}^2(x - 4t)$ .

The computational domain is  $[-10, 40]$ . The boundary conditions  $u(-10, t)$  and  $u(40, t)$  are specified by exact solution. In this example, the gradient of the initial condition is not noticeable with respect to the length of the region

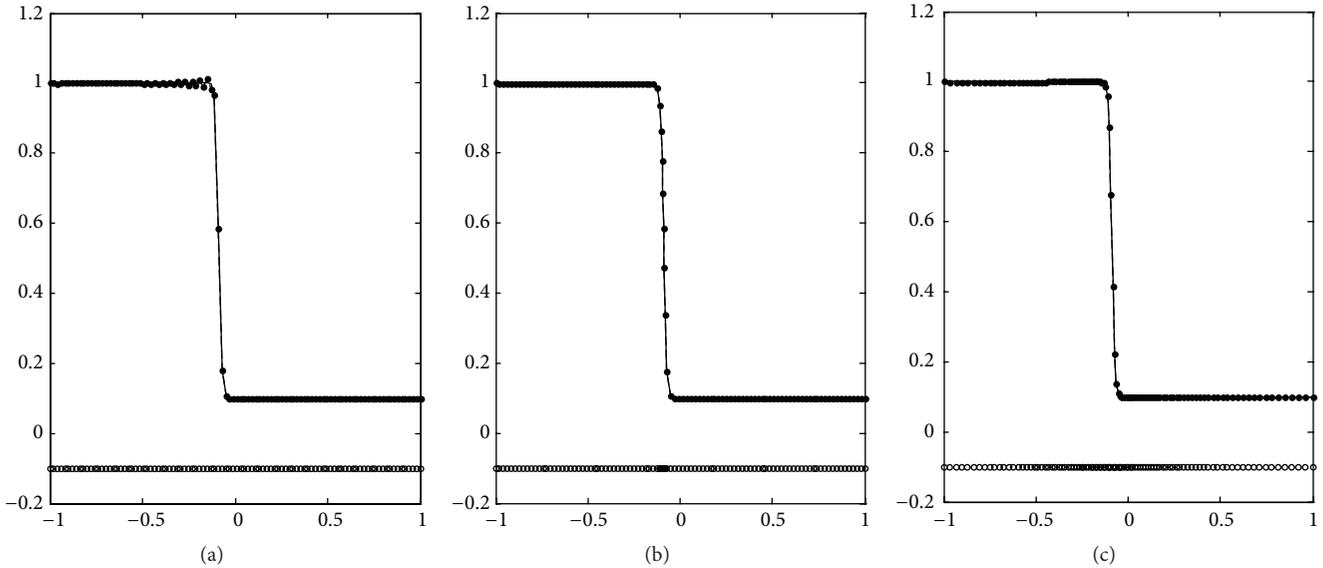


FIGURE 3: The plot of exact and approximate solutions of Example 5 with 100 uniform, quasi-uniform, and locally bounded nodes (resp., (a)–(c)).

TABLE 2: The results of Example 6 with different number of nodes and different mesh distribution.

Mesh	$t$	$n$	Shape	$k$	$\alpha$	$L_\infty$	$L_2$
Uniform	1	100	1.5	—	—	0.2679	0.0692
		150	1.5	—	—	0.0043	0.0020
		200	1.5	—	—	$1.4216e - 4$	$6.6842e - 5$
Quasi-uniform	1	100	1.1	4	$1e - 8$	0.0136	0.0046
		100	0.8	4	$1e - 8$	0.0067	0.0020
		151	0.8	2	$1e - 8$	0.0033	0.0015
Locally bounded	1	50	1.7	1.05	$1e - 16$	0.0074	0.0027
		80	1.1	1.01	$1e - 16$	$7.6419e - 4$	$3.0145e - 4$

$[-10, 40]$ . Thus the role of the parameters  $k$  and  $\alpha$  to control the concentrate of the nodes in appropriate region is very important. Solution and absolute error of the approximate solutions for different mesh nodes are plotted in Figure 3. The results of implementation are summarized in Table 2.

### 5. Results and Conclusions

In this paper, application of adaptive nodes, in MMOL, to time-dependent PDEs is investigated. An Equidistribution algorithm was introduced to produce adaptive central nodes. To avoid ill-conditioning in the RBF approximation, two constraints were imposed to adaptive nodes. To illustrate the effect of constraints, the method was applied to two well-known time-dependent PDEs.

In Example 5, the results shown in Figure 2 confirm that, for  $n = 50$ , the approximate solution by the locally bounded nodes has a good accuracy while two others cannot make a reliable approximate solution with the same number of nodes; they need more number of nodes. For  $n = 100$ , the quasi-uniform nodes work as well as locally bounded nodes but the uniform nodes do not have good accuracy yet (Figure 3).

For  $n = 150$ , all of three approaches are in a good accuracy. Thus, the results illustrate that the approximate solution based on the locally bounded nodes has more accuracy than two others, although the quasi-uniform nodes make more accurate approximate solution from uniform nodes. Based on the results of Table 1, the  $L_2$ -norm of the approximate solution for  $n = 100$  with uniform nodes seems acceptable but the figures show that this approximate solution is not accurate, whereas two other meshes make more accurate approximations. Additively, the results reveal that the locally bounded mesh has more accurate approximation solution with smaller number of nodes. For  $n = 150$ , it is illustrated that the results of three meshes are same, because with a large number of nodes, to avoid the ill-conditioning, the value of  $k$  must be small, and this leads the constraint meshes to a uniform mesh. Consequently, the results become same or close together.

In Example 6, Figure 4 shows that for  $n = 100$  uniform nodes the approximate solution is not accurate, and to obtain more accurate solution, at least 150 uniform nodes are needed, whereas for  $n = 100$  quasi-uniform nodes and  $n = 100$  locally bounded nodes, one can obtain a good accuracy.

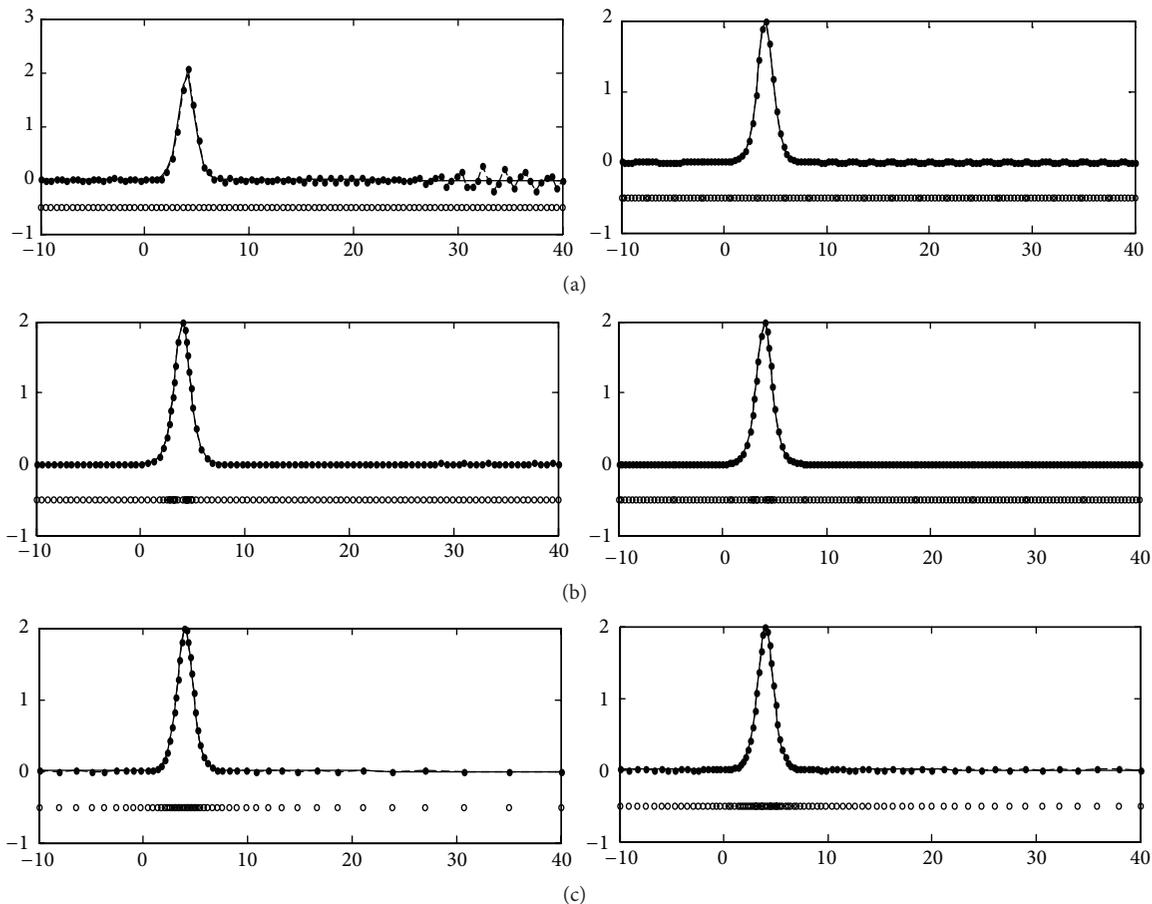


FIGURE 4: The plot of exact and approximate solutions of Example 6 with 100 and 150 uniform nodes (a), 100 and 150 quasi-uniform nodes (b), and 51 and 81 locally bounded nodes (c).

The results of this example illustrate that the quasi-uniform mesh makes more accurate results from uniform mesh. Also by using the locally bounded mesh, the results have the best accuracy than the others. It comes from the smoothness properties of the produced mesh in locally bounded based on initial condition and the solution function. Analysis of the results in Table 2 illustrates that, in this example by applying locally bounded mesh with  $n = 81$ , the error of approximate solution is more accurate than the error of approximate solution obtained by uniform mesh and quasi-uniform mesh with  $n = 150$ . The obtained results by uniform mesh with  $n = 200$  are comparable with the one obtained by locally bounded with  $n = 81$ . Thus the results reveal that locally bounded mesh produces more accurate solution in comparison with quasi-uniform mesh, and both of them are better than uniform mesh. It is noted that two parameters  $k$  and  $\alpha$  have very important role in the smoothness of the meshes and so in the accuracy of the results.

### Conflict of Interests

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

### References

- [1] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates," *Computers & Mathematics with Applications*, vol. 19, no. 8-9, pp. 127–145, 1990.
- [2] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations," *Computers & Mathematics with Applications*, vol. 19, no. 8-9, pp. 147–161, 1990.
- [3] S. R. Idelsohn and E. Oñate, "To mesh or not to mesh. That is the question...", *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 37–40, pp. 4681–4696, 2006.
- [4] M. Bozzini, L. Lenarduzzi, and R. Schaback, "Adaptive interpolation by scaled multiquadrics," *Advances in Computational Mathematics*, vol. 16, no. 4, pp. 375–387, 2002.
- [5] R. Schaback and H. Wendland, "Adaptive greedy techniques for approximate solution of large RBF systems," *Numerical Algorithms*, vol. 24, no. 3, pp. 239–254, 2000.
- [6] Y. C. Hon, R. Schaback, and X. Zhou, "An adaptive greedy algorithm for solving large RBF collocation problems," *Numerical Algorithms*, vol. 32, no. 1, pp. 13–25, 2003.

- [7] Y. C. Hon, "Multiquadric collocation method with adaptive technique for problems with boundary layer," *International Journal of Applied Science and Computations*, vol. 6, no. 3, pp. 173–184, 1999.
- [8] J. Behrens and A. Iske, "Grid-free adaptive semi-Lagrangian advection using radial basis functions," *Computers & Mathematics with Applications*, vol. 43, no. 3–5, pp. 319–327, 2002.
- [9] V. Pereyra and E. G. Sewell, "Mesh selection for discrete solution of boundary problems in ordinary differential equations," *Numerische Mathematik*, vol. 23, no. 3, pp. 261–268, 1974.
- [10] J. Kautsky and N. K. Nichols, "Equidistributing meshes with constraints," *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 4, pp. 499–511, 1980.
- [11] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: an overview and recent developments," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 3–47, 1996.
- [12] W. E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press, San Diego, Calif, USA, 1991.
- [13] S. A. Sarra, "Adaptive radial basis function methods for time dependent partial differential equations," *Applied Numerical Mathematics*, vol. 54, no. 1, pp. 79–94, 2005.
- [14] T. A. Driscoll and A. R. Heryudono, "Adaptive residual subsampling methods for radial basis function interpolation and collocation problems," *Computers & Mathematics with Applications*, vol. 53, no. 6, pp. 927–939, 2007.
- [15] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, World Scientific, Singapore, 2007.
- [16] E. J. Kansa and R. E. Carlson, "Improved accuracy of multiquadric interpolation using variable shape parameters," *Computers & Mathematics with Applications*, vol. 24, no. 12, pp. 99–120, 1992.
- [17] S. Rippa, "An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation," *Advances in Computational Mathematics*, vol. 11, no. 2–3, pp. 193–210, 1999.
- [18] W. Cao, W. Huang, and R. D. Russell, "A study of monitor functions for two dimensional adaptive mesh generation," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1978–1994, 1999.
- [19] G. F. Carey and H. T. Dinh, "Grading functions and mesh redistribution," *SIAM Journal on Numerical Analysis*, vol. 22, no. 5, pp. 1028–1040, 1985.



# Hindawi

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

