

Radial Basis Functions Based Differential Quadrature Method for One Dimensional Heat Equation

Kedir Aliy^{1,*}, Alemayehu Shiferaw², Hailu Muleta²

¹Department of Mathematics, Ambo University College of Natural and Computational Sciences, Ambo, Ethiopia

²Department of Mathematics, Jimma University College of Natural Sciences, Jimma, Ethiopia

Email address:

kediraliyi39@gmail.com (K. Aliy)

*Corresponding author

To cite this article:

Kedir Aliyi, Alemayehu Shiferaw, Hailu Muleta. Radial Basis Functions Based Differential Quadrature Method for One Dimensional Heat Equation. *American Journal of Mathematical and Computer Modelling*. Vol. 6, No. 2, 2021, pp. 35-42. doi: 10.11648/j.ajmcm.20210602.12

Received: March 28, 2021; Accepted: May 11, 2021; Published: May 26, 2021

Abstract: In this paper, Radial basis functions based differential quadrature method has been presented for solving one-dimensional heat equation. First, the given solution domain is discretized using uniform discretization grid point in both direction and the derivative involving the spatial variable, x is replaced by the sum of the weighting coefficients times functional values at each grid points. Next by using properties of linear independence of vector space and Multiquadratic radial basis function we can find all waiting coefficient at each grid points of solution domain and we obtain first order linear system of ordinary differential equation with N by N square coefficient Matrices. Then, the resulting first order linear ordinary differential equation is solved by fifth-order Runge-Kutta method. To validate the applicability of the proposed method, one model example is considered and solved for different values of the shape parameter 'c' and mesh sizes in the direction of the temporal variable; t and fixed value of mesh size in the direction of spatial variable, x . Numerical results are presented in tables in terms of root mean square (E^2), maximum absolute error (E^{∞}) and condition number $K(A)$ of the system matrix. The numerical results presented in tables and graphs confirm that the approximate solution is in a good agreement with the exact solution.

Keywords: Heat Equation, Shape Parameter, Multiquadric-Radial Basis Functions, Weighting Coefficients, Mesh-free Method

1. Introduction

Partial Differential Equations (PDEs) are mathematical equations that are significant in modeling physical phenomena that occur in nature. Applications of PDEs can be found in physics, engineering, mathematics, and finance. Examples include: modeling mechanical vibration, heat, sound vibration, elasticity, and fluid dynamics [7]. Although PDEs have a wide range of applications to real world problems in science and engineering, the majority of PDEs do not have analytical solutions.

Traditionally, mesh-dependent methods such as the finite difference method (FDM), finite element method (FEM), and boundary element method (BEM) have been used to solve PDEs [16]. Despite their great success in the past decades in many branches of science and engineering, these mesh-based methods

require meshes or grids as the solution domain. The costs and difficulties in creating quality meshes, however, constitute one of the major bottlenecks in these methods [17]. However, may be the complications of these methods include a slow rate of convergence, spatial dependence, instability, low accuracy, and difficulty of implementation in complex geometries [16, 18].

But scientists in the field of computational mathematics have been trying to develop numerical methods by using computers for further application [18]. One of those numerical methods is differential quadrature method. The differential quadrature method (DQM) was introduced by Richard Bellman and his associates in the early of 1970s following the idea of integral quadrature [19, 14]. The DQM are able to produce the accurate result using only a small number of grid points. They are mesh-based methods. Furthermore, the distribution of the nodes has limitations,

and they must be clustered near the boundary and hence limits its usefulness [16].

However, mesh-less approximation techniques using radial basis functions (RBFs) have been developed over the last several decades. These methods are easy to implement, highly accurate, and truly mesh less, which avoids troublesome mesh generation for high-dimensional problems [20]. These methods are called radial basis function based differential quadrature methods [20]. Shu *et. al.* Solved partial differential equations by a global radial basis function-based differential quadrature method.

In recent decades, various RBFs-based methods have gained fast growing attention from abroad range of scientific computing and engineering applications such as multivariate scattered data processing, numerical solutions of PDEs, and machine learning [3]. Franke in 1982 [9] extensively tested 29 different algorithms on the typical benchmark function interpolation problems, and ranked the Multi-quadric Radial Basis Function [21] (MQ-RBF) and Thin Plate Spline Radial Basis Function (TPS-RBF) as two of the best candidates based on the following criteria: timing, storage, accuracy, and ease of implementation. Kansa (1990) first developed an RBF collocation scheme for solving PDEs of elliptic, parabolic, and hyperbolic types, in particular, using the MQ-RBFs. The methodology is now often called the Kansa method. The Kansa method is mesh less and has distinct advantages compared with the traditional methods: superior convergence, integration-free, and easy implementation. Mesh-less method are a class of numerical methods used for solving partial differential equations. In these methods, mesh generation on the spatial domain of the problem is not needed. This property is the main advantage of these techniques over the mesh dependent methods. Due to the wide range of the application of the one dimensional linear parabolic equation, several numerical methods have been developed.

Benyam Mebrate [1] presented Numerical solutions of a one-dimensional heat Equation together with initial condition and Dirichlet boundary conditions of the form: $u_t = \alpha^2 u_{xx}$. He presented computation of the numerical solutions by using two methods: Finite difference, and Finite element methods. The finite element methods are implemented by Crank-Nicolson method. This method does not always converge to the exact solutions for coarser step lengths. He got the accurate approximate solution with the length of time-step $k=0.001$ on $0 < t < 1$. The method applied is not efficient as it requires longer CPU time and large memory size. Hooshmandasl *et al.*, [10] used Chebyshev Wavelets Method to obtain a numerical solution of the one-dimensional heat equation. They applied different wavelet families and the wavelet coefficients were calculated by the Galerkin collocation method. Generally, they developed Chebyshev wavelet method with operational matrices of integration for solution of one dimensional heat equation with Dirichlet boundary conditions which is fast, mathematically simple and guarantees the necessary accuracy for a relatively small number of grid points. This confirms that the method is not accurate for a relative large number of

grid points and is difficult to apply for high dimension geometric spaces. In their work, Greengard and Li [12] also developed an explicit method for solving inhomogeneous heat equation in free space, following the time evolution of the solution in the Fourier domain. The error in the solution is simply the quadrature error in evaluating the solution and the solution is dependent on the full time-space history of the diffusion process with the time-step $k=0.0025$ and they got accurate solution for finer spatial step-length and time step. So, the method has required high cost regarding to storage capacity in the computational domain.

Kalyanil and Rao in [11] solved one-dimensional heat equation by using double interpolation. They used finite difference method for double interpolation method to solve one-dimensional heat equation, but the method gives better accuracy only for small step length and is difficult to compute the solution in complex computational domain. Even though the accuracy of the aforementioned methods is promising, they require large memory and long computational time. Besides, the methods are not suitable for higher dimensional and problems involving complex geometries. So, the treatment of the mesh-size presents severe difficulties that have to be addressed to ensure accuracy of the solution, and efficiency of the method applied.

Tatari and Dehghan [15] solved heat equation using radial basis functions. They applied the Gaussian radial basis functions for obtaining solution of heat equation. The solution diverges when shape parameters are: 0.5, 1, and 2. As compared to the exact solution, the approximate solution needs further improvement. To this end, the aim of this paper is to construct efficient and accurate numerical method for solving one-dimensional heat equation.

2. Preliminaries

2.1. Differential Quadrature Method

Differential quadrature method is one of the most efficient numerical methods to solve partial differential equations [18]. Different differential quadrature methods were developed by Bellman R. and Casti in 1971 as cited in [19]. A variety of methods have been developed based on the DQ method, including the polynomial-based differential quadrature (PDQ) and the Fourier-expansion-based differential quadrature methods (FDQ) [19]. The basic idea behind the DQ method is to determine the weighting coefficients for any order derivatives by using a weighted sum of functional values at a set of selected grid points [5]. PDQM and FDQM are highly efficient method by using a small number of grid points, they are not efficient when the number of grid-points is large and they are also sensitive to grid point distribution. While the PDQ and FDQ methods are able to obtain accurate results using only a small number of grid points, they are mesh-based methods [8, 16].

To overcome the limitations for the applications of DQ method, a new class of methods called mesh-free methods has surfaced. Each class of methods offers numerous and, in many ways, complementary benefits. In the ideal case we

seek a method defined on arbitrary geometries that behaves regularly in any dimension, and avoids the cost of mesh generation. The ability to locally refine areas of interest in a practical fashion is also desirable. Fortunately, mesh-free methods provide all of these properties: based wholly on a set of independent points in n-dimensional space, there is minimal cost for mesh generation, and refinement is as simple as adding new points where they are needed. A subset of mesh-free methods of particular interest to the numerical modeling community today revolves around Radial Basis Functions (RBFs). This method is called radial basis function differential quadrature method (RDQM). RBF methods are based on a superposition of translates of these radially symmetric functions, providing a linearly independent but non-orthogonal basis to interpolate between nodes in n-dimensional space [2].

2.2. Radial Basis Function

Definition 2.2.1 A function $\phi: \mathfrak{R}^d \rightarrow \mathfrak{R}$ is called radial if there exists a univariate function $\varphi: [0, \infty) \rightarrow \mathfrak{R}$ such that $\phi(x) = \varphi(r)$ where $r = \|x\|$ and $\|\bullet\|$ is a norm in \mathfrak{R}^d

($\|\bullet\|$ is typically the Euclidean norm).

A radial basis function $\varphi(r)$ is a one-variable, continuous function define for $r \geq 0$ that has been radialized by composition with the Euclidean norm on \mathfrak{R}^d . RBFs may have a free parameter, called the shape parameter, denoted by c . RBFs are a class of radially symmetric functions symmetric about a point x_j called the centres. It is given in the form of

$$\phi(x) = \phi(r(x))$$

Here, the value of the univariate function φ is a function of the Euclidean distance from the canter of point x_j and $r(x)$ is given by: $r_j(x) = \|x - x_j\| = \sqrt{(x - x_j)^2}$ in one dimensional space (Chenoweth, 2012). RBFs are a powerful tool in interpolating multivariable functions or approximation solution of partial or ordinary differential equations.

The most popular radial basis functions are:

- i. Multiquadric RBF: $\phi(r_j) = \sqrt{r_j^2 + c^2}$
- ii. Inverse Multiquadric RBF: $\phi(r_j) = \frac{1}{\sqrt{r_j^2 + c^2}}$
- iii. Gaussian RBF: $\phi(r_j) = e^{-(cr_j)^2}$
- iv. Thin plate spline RBF: $\phi(r_j) = r_j^2 \ln(r_j)$

These radial basis functions (except TPS) are parametric functions with shape (control) parameter c . They are smooth and continuously differentiable functions. Except the Multiquadric RBF, interpolation matrices of these radial basis functions are positive definite, but the interpolation matrices for Multiquadric is conditional positive definite

[4]. The RBFs are usually divided into global and local support RBFs. It is said to be global support if the $\lim_{r \rightarrow \infty} \phi(r_j) = \infty$ and local supporting if the $\lim_{r \rightarrow \infty} \phi(r_j) = 0$ [7].

The global approach uses information from every center in the domain to approximate a function value or derivative at a single point. In contrast, the local method only uses a small subset of the available centers [4].

For given data (x_j, f_j) the smooth function p defined by enforcing that $p(x_j) = f_j$. Therefore, for any scattered set of point N with center x_j the radial basis interpolation function is given as:

$$p(x_j) = \sum_{j=1}^N \alpha_j \phi(\|x - x_j\|)$$

By enforcing the interpolation condition $p(x_j) = f_j$, we determine the coefficients α_j from $N \times N$ matrix for $j = 1, 2, 3, \dots, N$. The choice of the basic functions will determine which methods are available for solving system of interpolation and whether such a solution even exists. If the interpolation matrix is symmetric positive definite, then the linear system has a unique solution [14].

3. Description of the Method and Results

3.1. Description of the Method

Consider the one dimensional heat equation of the form:

$$u_t = \alpha u_{xx}, (x, t) \in (a, b) \times (0, T] \tag{1}$$

Subject to initial condition:

$$u(x, 0) = f_1(x), a \leq x \leq b \tag{2}$$

and boundary conditions

$$u(a, t) = g_1(t), u(b, t) = g_2(t), 0 \leq t \leq T \tag{3}$$

Here, $f_1(x)$, $g_1(t)$ and $g_2(t)$ are continuous and differentiable functions. The computational domain $[a, b] \times [0, T]$ is partitioned as:

$$x_i = x_0 + ih, i = 1, 2, \dots, M$$

$$t_j = t_0 + jk, j = 1, 2, \dots, N$$

where h and k are mesh-size of $[0, 1]$ and $[0, T]$, respectively.

3.2. Multiquadric Radial Basis Functions (MQ-RBFs)

Multiquadric RBF was proposed by Hardy as cited by Ding et al., in [6] for the interpolation of topographical surfaces. It has very simple mathematical form as

$$\phi(r_j) = \sqrt{(x-x_j)^2 + c^2}$$

where c is a shape parameter and $c > 0$. For time dependent problems, the multiquadric radial basis function is given by

$$\phi((x,t),(x_j,t)) = \phi(\|(x,t)-(x_j,t)\|) = \sqrt{(x-x_j)^2 + c^2}, \quad j = 1, 2, \dots, N \tag{4}$$

where N is the number of grid points in the direction of the spatial variable x , t is the temporal variable.

Definition -1: [4] A function φ is completely monotonic on $[0, \infty)$ if $\varphi \in C[0, \infty)$, $\varphi \in C^\infty(0, \infty)$, and $(-1)^n \varphi^{(n)}(r) \geq 0$ where $r > 0$ and $n = 0, 1, 2, \dots$

Theorem -1: [4] Let $\varphi(r) = \varphi(\sqrt{r}) \in C[0, \infty)$ and $\varphi(r) > 0$ for $r > 0$. Let $\varphi'(r)$ be completely monotonic on $[0, \infty)$, then for any set of N distinct center x_j , $j = 1, 2, 3, \dots, N$ the $N \times N$ matrix B with entry $b_{kj} = \varphi(\|x_k - x_j\|)$ is invertible. Such function is said to be a conditional positive definite.

MQ-RBF is an example of global infinitely differentiable and conditional positive definite function.

Theorem -2: [13]. For a set of N distinct data points to be approximated by the set of N approximation function ϕ and this function must be nonsingular.

Therefore, by the above definition and theorems, the interpolation matrix resulting from MQ-RBFs is invertible, smooth, and continuously differentiable on its domain. Thus, the first and second order derivatives of multiquadric radial basis functions are given by:

$$\phi_x(\|x-x_j\|) = \frac{x-x_j}{\sqrt{(x-x_j)^2 + c^2}} \tag{5}$$

$$\phi_{xx}(\|x-x_j\|) = \frac{c}{\sqrt{[(x-x_j)^2 + c^2]^3}} \tag{6}$$

3.3. Approximating the Partial Derivatives and Determining the Weighting Coefficients

From the primary idea of differential quadrature method, $u_{xx}(x,t)$ in Eq. (1) can be expressed as a linear combination of functional values at each grid points. It is given as:

$$u_{xx}(x_i,t) = \sum_{j=1}^N w_{ij} u(x_j,t), \quad i = 1, 2, \dots, N. \tag{7}$$

Here, w_{ij} are the weighting coefficients. Since $\phi(x,t)$ is a basis function for one dimensional space, every function defined on one dimensional space can be expressed as a linear combination of these basis functions using the properties of linear independence of vector in vector space. Now for any constant λ and at least one of $\lambda \neq 0$ such that

$$u(x,t) = \sum_{j=1}^N \lambda_j \phi(x,t), \quad i = 1(N)N \tag{8}$$

Since $\phi(x,t) = \phi((x,t),(x_j,t)) = \phi(\|(x,t)-(x_j,t)\|) = \sqrt{(x-x_j)^2 + c^2}$ the first and second order derivatives of Eq. (8) become

$$u_x(x_i,t) = \sum_{j=1}^N \lambda_j \phi_x(\|x-x_j\|), \quad i = 1, 2, 3, \dots, N \tag{9}$$

$$u_{xx}(x_i,t) = \sum_{j=1}^N \lambda_j \phi_{xx}(\|x-x_j\|), \quad i = 1, 2, 3, \dots, N \tag{10}$$

Substituting Eqs.(8) and (10) into Eq. (7), we get:

$$\sum_{j=1}^N \lambda_j \phi_{xx}(\|x-x_j\|) = \sum_{j=1}^N w_{ij} \sum_{j=1}^N \lambda_j \phi(x,t) \tag{11}$$

$$\phi_{xx}(\|x-x_j\|) = \sum_{j=1}^N w_{ij} \phi(\|x-x_j\|) \tag{12}$$

From Eq. (12) we have the following system of equations:

$$Aw = b, \tag{13}$$

$$A = \begin{bmatrix} \phi(\|x_1-x_1\|) & \phi(\|x_1-x_2\|) & \dots & \phi(\|x_1-x_N\|) \\ \phi(\|x_2-x_1\|) & \phi(\|x_2-x_2\|) & \dots & \phi(\|x_2-x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_N-x_1\|) & \phi(\|x_N-x_2\|) & \dots & \phi(\|x_N-x_N\|) \end{bmatrix}$$

$$b = \begin{bmatrix} \phi_{xx}(\|x_1-x_1\|) \\ \phi_{xx}(\|x_1-x_2\|) \\ \vdots \\ \phi_{xx}(\|x_i-x_N\|) \end{bmatrix}, \quad \text{and } w = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{iN} \end{bmatrix}$$

Since ϕ is global support RBFs it produces a dense matrix A . By theorems 1 and 2 the matrix A is conditional positive definite and non-singular. Therefore, the system in Eq. (13) has a unique solution. The condition number of A is calculated as

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 \tag{14}$$

For large number of $\kappa(A)$, the system in Eq. (13) is ill-conditioned. This is because of the presence of shape parameter " c " that affects both the condition number of the interpolation matrices A and the accuracy of the method. For a fixed number of interpolation points N , the condition number of A depends on the shape parameter c and support of the RBFs [22]. Also, the condition number grows with N for fixed value of the shape parameter c [15].

Now the weighting coefficients (w) can be obtained from linear system of equation given in Eq. (13) by using Gaussian-Elimination method. In this case, the augmented matrix is written as

$$[A : b] = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1N} & \phi_{xx}^1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2N} & \phi_{xx}^2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \phi_{N1} & \phi_{N2} & \dots & \phi_{NN} & \phi_{xx}^N \end{bmatrix} \quad (15)$$

where $\phi_{ij} = \phi(\|x_i - x_j\|)$ and $\phi_{xx}^i = \phi_{xx}(x_i)$ for $i, j = 1(1)N$

By row operation, the augmented matrix is changed to the upper triangular matrix.

$$[A^c : b^c] = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1N} & \phi_{xx}^{1c} \\ 0 & \phi_{22}^c & \dots & \phi_{2N}^c & \phi_{xx}^{2c} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & \phi_{NN}^c & \phi_{xx}^{Nc} \end{bmatrix} \quad (16)$$

Now, we can calculate the weighting coefficients using backward substitution.

The N^{th} weighting coefficient w_N can be obtained by using the formula:

$$w_N = \frac{\phi_{xx}^{Nc}}{\phi_{NN}^c} \quad (17)$$

The $(N-1)^{th}$ weighting coefficient w_{N-1} can also be obtained by using the formula:

$$w_{N-1} = \frac{\phi_{xx}^{N-1c} - \phi_{xx}^{Nc} w_N}{\phi_{N-1, N-1}^c} \quad (18)$$

Generally, the i^{th} weighting coefficient can be obtained by using the formula:

$$w_i = \frac{\phi_{xx}^i - \sum_{j=i+1}^N \phi_{ij}^c w_j}{\phi_{ii}^c} \quad (19)$$

for $i = N-2, N-3, \dots, 2, 1$

Then, the coefficient vector (w_i) can be used to approximate the second-order derivative in the x-direction for unknown smooth function at i^{th} nodes. From the procedure of DQ approximation of derivatives, it can be observed that the weighting coefficients are only dependent on the selected RBFs and the distribution of the supporting points in the local support [6].

3.4. Results and Discussion

From Eqs. (1-3), (8), (10) and (12), the resulting system of initial-boundary value problem is given as:

$$\frac{du(x_i, t)}{dt} = \alpha \sum_{j=2}^{N-1} w_{ij} u(x_j, t), \quad i = 1(1)N \quad (20)$$

Subject to initial condition

$$u(x_i, 0) = f_1(x_i), \quad a \leq x_i \leq b, \quad (21)$$

and boundary conditions

$$u_1(t) = g_1(t), u_N(t) = g_2(t), \quad 0 \leq t \leq T \quad (22)$$

3.5. Criteria for Investigating the Accuracy of the Method

The root Mean Square (RMS) error (E^2), maximum absolute error (E^∞) are used to measure the accuracy of the method. The RMS error and maximum absolute error are calculated as follows in [15],

$$E^2 = \sqrt{\frac{1}{N} \sum_{k=1}^N |U(x_i, t_j) - u(x_i, t_j)|^2}, \quad i = 1(1)N, \quad (23)$$

$$E^\infty = \max_{1 \leq i \leq N} |U(x_i, t_M) - u(x_i, t_M)|, \quad (24)$$

Here, $U(x_i, t_M)$ and $u(x_i, t_M)$ are the exact and approximation solutions of Eqs.(1), (2), and (3), respectively.

3.6. Numerical Experiments

Example1: Consider the classical heat equation considered by Dehghan & Tatari, in [15],

$$u_t = u_{xx}, \quad 0 < x < 1, \quad 0 < t \leq T,$$

with initial condition

$$u(x_i, 0) = \sin(\pi x_i), \quad 0 \leq x_i \leq 1$$

and boundary conditions

$$u(0, t) = u(1, t) = 0, \quad 0 \leq t \leq T,$$

The exact solution is given by:

$$U(x_i) = \sin(\pi x_i) \exp(-\pi^2 t),$$

The numerical results are presented in tables in terms of E^2 , E^∞ and $K(A)$ as the means for measuring the accuracy the present method.

Table 1. Point wise absolute error, root mean square error, and Condition Number of example1 when $h=0.1, T = 1$ and $\Delta t = 0.01$.

Our Method			
Shape parameter c	E^2	E^∞	$\kappa(A)s$
0.5	3.2888E-01	3.3052E-0	4.0762E+07
1	6.2787E-02	6.3100E-01	1.0457E+12
2	1.5974E-02	1.6054E-01	8.0464E+16
3	7.2063E-03	7.2422E-02	2.6283E+17
4	4.0799E-03	4.1003E-02	6.4077E+16
5	2.6197E-03	2.6327E-02	2.9761E+17
6	1.8226E-03	2.6327E-02	3.5931E+17
7	1.3405E-03	1.3472E-03	7.3104E+17

Tatari, M. and Dehghan, M., 2010			
Shape parameter c	E^2	E^∞	$\kappa(A)s$
0.5	9.0651E+39	7.0000E+73	1.9676E+19
1	3.3136E+17	2.0000E+29	2.0434E+17
2	5.0227E+1	5.0001E+0	9.3751E+17
3	1.0196E-01	7.1266E-03	6.6616E+17
4	1.0384E-01	7.4346E-03	4.0811E+17
5	1.0400E-03	7.6029E-03	1.1529E+16
6	1.0150E-03	7.5571E-03	1.4832E+16
7	9.5715E-03	7.3241E-03	3.0267E+15

Table 2. Point wise absolute error, root mean square error and Condition number of example1 when $h=0.1, T = 1$ and $\Delta t = 0.025$.

Our Method			
Shape parameter c	E^2	E^∞	$\kappa(A)s$
0.5	1.7625E+01	1.1286E+02	4.0762E+07
1	3.3389E-01	3.3556E-00	1.0457E+12
2	7.5336E-02	3.3389E-01	8.0464E+16
3	3.0695E-02	1.9655E-01	2.6283E+17
4	1.6767E-02	1.0736E-01	6.4077E+16
5	1.0588E-02	6.7799E-02	2.9761E+17
6	7.3003E-03	4.6745E-02	3.5931E+17
7	5.3403E-03	3.4195E-02	7.3104E+17

Table 3. Point wise absolute error, root mean square error and Condition number of example1 when $h=0.1, T = 1$ and $\Delta t = 0.001$.

Our Method			
Shape parameter c	E^2	E^∞	$\kappa(A)s$
0.5	2.9601E-03	9.3653E-02	4.0762E+07
1	1.3346E-03	4.2226E-02	1.0457E+12
2	4.4049E-04	1.3937E-02	8.0464E+16
3	2.0921E-04	6.6190E-03	2.6283E+17
4	1.2064E-04	3.8168E-03	6.4077E+16
5	8.1757E-05	2.5867E-03	2.9761E+17
6	5.7186E-05	1.8093E-03	3.5931E+17
7	4.2198E-05	1.3351E-03	7.3104E+17

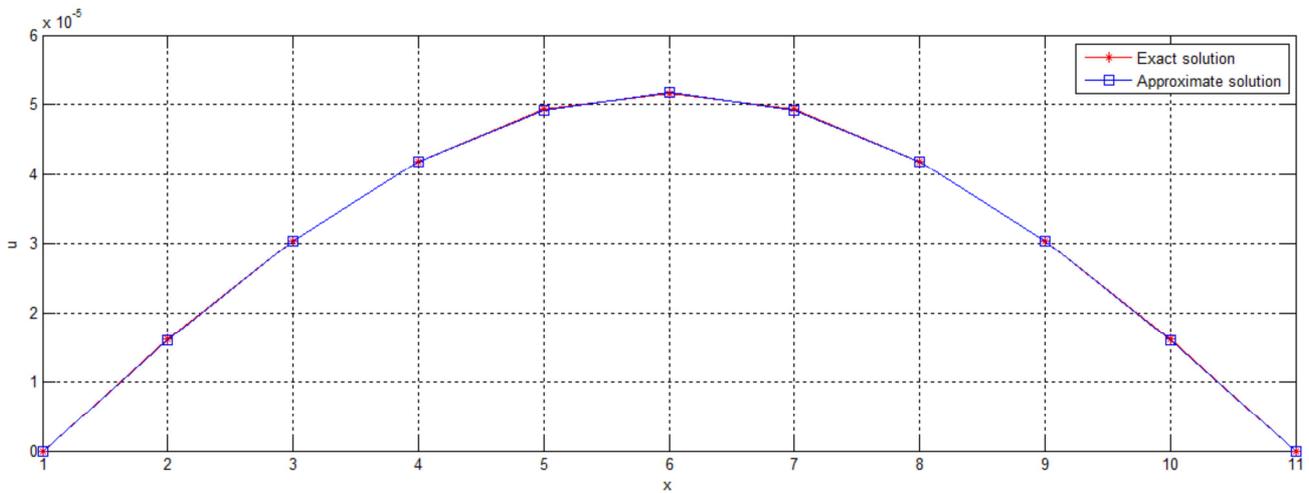


Figure 1. Profile of plotting graph of the Approximate versus Exact solution of example 1 for which $c = 4, h = 0.1, T = 1$ & $\Delta t = 0.01$.

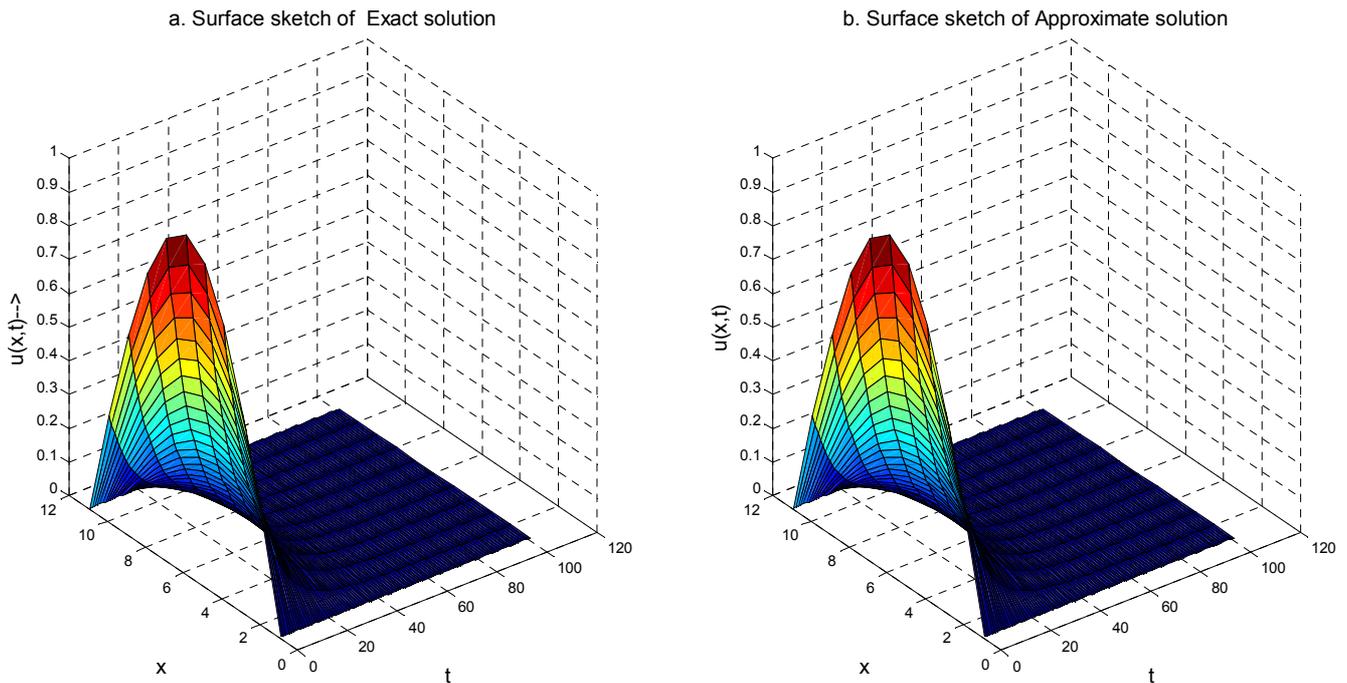


Figure 2. Surface graphs of example 1 showing the physical behavior of the one-dimensional heat equation when $c = 4$, $h = 0.1$, $T = 1$ & $\Delta t = 0.01$.

4. Discussion

As depicted in Table 1, the present method is able to generate convergent numerical solution when $c = 0.5, 1$, and 2 at which the method presented by Tatari and Dehghan, in [15] fails to produce convergent solution. The condition number of the system matrix of the present method is in the range $4.0762E+07 \leq \kappa(A) \leq 7.3104E+17$ whereas the condition number of the system matrix presented by Tatari and Dehghan, $20103.0267E+15 \leq \kappa(A) \leq 1.9676E+19$. Thus, the effect of the condition number on the accuracy of the numerical solution is more significant on the method presented by Tatari and Dehghan, than on the numerical solution of the present method. The value of E^∞ in Table 1 confirms this issue. That is the smaller the value of E^∞ the less the effect of the condition number on the accuracy of the approximate solution. As it can also be seen from Tables 1-Table 3, the condition number of the system matrix is kept constant for same values of the shape parameter c in each table. This is because the condition number depends only on the step length the spatial variable and the shape parameter c . Comparison among Table 1-Table 3 shows that more accurate result is generated when $\Delta t = 0.001$. This indicates that the present method is suitable for problems that require long time interval.

The simulations presented in Figures 1 and 2 shows that the approximate solution obtained by the present method is in a good agreement with the exact solution.

5. Conclusion

In this thesis, RBFS-DQM is used to solve one

dimensional Heat equation. First, the domain is discretized using the uniform step length and the derivative involving the spatial variable 'x' is replaced by the sum of weighting coefficient times the functional value at each grid point via Multiquadric radial basis functions. Then the resulting first order linear system of ODE is solved by fifth order Runge-Kutta method.

To validate the applicability of the method, one model example is considered and solved by varying the value of shape parameter c and time-step k , and keeping the step-length h fixed. As it can be seen from the numerical results presented in tables and graphs, the present method is superior over the method developed by Tatari and Dehghan (2010) and approximates the exact solution very well. In a nutshell, the present method is conceptually simple, easy to use and readily adaptable for computer implementation for solving one-dimensional heat equation.

Acknowledgements

The authors are grateful to Jimma University for fully funding the project work so that it has become reality.

References

- [1] Benyam Mabrata (2015). Numerical solution of a one dimensional Heat Equation with Dirichlet boundary conditions. *American Journal of Applied Mathematics*, 3 (4): 305-311.
- [2] Boling, E. F. (2013). Multi-GPU solutions of geophysical PDEs with radial basis Function-generated finite differences. *PhD thesis Florida State University*.

- [3] Chen, W., Fu, Z. J., and Chen, C. S. (2014). Recent advances in radial basis function collocation methods, Springer Briefs in Applied Sciences and Technology.
- [4] Chenoweth, M. N. (2012). A local radial basis function method for the numerical Solution of partial differential equations. PhD thesis, Marshall University.
- [5] Cheong, H. T. (2015). Parallel Localized differential quadrature approach in boundary value problem. *Journal of math. Analysis and appl.* 31 (2): 127-134.
- [6] Ding, H., Shu, C., and Tang, D. B. (2005). Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments. *International journal for numerical method in Engin.* 63 (11): 1513-1529.
- [7] Erfanian, M., and Kosari, S. (2017). Using Chebyshev polynomials zeros as point grid for numerical solution of nonlinear PDEs by differential quadrature-based radial basis functions. *Int. Journ. Of Math's. Modelling and Computations*, 7 (1): 67-77.
- [8] Fasshauer, G. (2007). Meshfree Application Method with Matlab. Interdisciplinary Mathematical Sciences.
- [9] Franke, R. (1982). Scattered data interpolation: tests of some methods. *Mathematics of computation*, 38 (157): 181-200.
- [10] Hooshmandasl M. R Haidari, M., and Maalek Ghaini F. M. (2012). Numerical solution of one dimensional heat equation by using Chebyshev Wavelets method. *JACM an open access journal*, 1 (6), 1-7.
- [11] Kalyanil, P., and Rao (2013). Numerical solution of heat equation through double Interpolation. *IOSR Journ. Of math.* 6 (6): 58-62.
- [12] Li, J. R., and Greengard, L. (2007). On the numerical solution of the heat equation I: Fast solvers in free space. *Journal of Computational Physics*, 226 (2): 1891-1901.
- [13] Micchelli, C. A (1984). Interpolation of scattered data: distance matrices and conditionally positive definite functions. In *Approximation theory and splinefunctions*: 143-145. Springer, Dordrecht.
- [14] Mongillo, M. (2011). Choosing basis functions and shape parameters for radial basis function methods. *SIAM undergraduate research online*, 4 (190-209): 2-6.
- [15] Tatari, M. and Dehghan, M., (2010). A method for solving partial differential equations via radial basis functions: Application to the heat equation *Engineering Analysis with Boundaryelement* 34 (3), 206-212.
- [16] Waston, D. (2017). Radial Basis Function Differential Quadrature Method for the numerical solution of partial differential equations. The Aquila Digital Community, Dissertations.
- [17] Yao, G. and Yu, Z. (2012). A Localized Mesh less Approach for Modeling Spatial temporal.
- [18] Masho Jima, Alemayehu Shiferaw, Ali Tsegaye. "Numerical Solution of the Coupled Viscous Burgers' Equation Using Differential Quadrature Method Based on Fourier. Expansion Basis", *Applied Mathematics*, 2018.
- [19] Zhang, Y., and Zong, Z. (2009). *Advanced differential quadrature methods*. Chapman and Hall/CRC. USA.
- [20] C. Shu, H. Ding, N. Zhao. "Numerical comparison of least square-based finite-difference (LSFD) and radial basis function-based finite-difference (RBFFD) methods, *Computers & Mathematics with Applications*, 2006.
- [21] Sen Li, Linqun Yao, Shichao Yi, Wei Wang. "A Meshless Radial Basis Function Based on Partition of Unity Method for Piezoelectric Structures", *Mathematical Problems in Engineering*, 2016.
- [22] Pouria Assari, Hojatollah Adibi, Mehdi Dehghan. "The numerical solution of weakly singular integral equations based on the mesh less product integration (MPI) method with error analysis", *Applied Numerical Mathematics*, 2014.