JZS-A Volume 22, Issue 2, December 2020

Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)
http://jzs.univsul.edu.iq

Solving the forward-backward heat equation in two-dimension by Radial basis functions mesh free method

Siamak Banei[1*], Kamal Shanazari[1], Yaqub Azari[2], and Yones Esmaeelzade Aghdam[2]

[1] *Department of Applied Mathematics, University of Kurdistan, Sanandaj, Iran*
[2] *Faculty of Science, Shahid Rajaee Teacher Training University, Lavizan, Tehran, 16785-163*
**Corresponding author's e-mail: sbaneh@gmail.com*

Article info

Original: 21 February 2020
Revised: 18 August 2020
Accepted: 26 September 2020
Published online: 20 December 2020

Key Words: Radial basis functions, Meshless method, Thin plate spline, Domain decomposition, Finite difference

Abstract

This study is specially designed to solve the forward-backward heat equation in two dimension to make use of thin plate splines and finite difference method. It should be noted that the problem had more than one part categorization. The results as a whole covered a large proportion solutions by a radial basis function method for spatial and finite difference scheme along with interface boundry usage . This method was presented based on updating interface boundry. In addition , the interpolation and the collcation methods have supported through using of thin plate splines and finite difference method to update interface boundary. Furthermore, it showed that the time dicretization scheme was unconditionally stable and convergent. . Finally, some numerical examples were assigned to graded methods to support their efficiency.

1. Introduction

In this article, the numerical solutions of forward-backward heat equation (FBHE) were found by presenting a mesh free method in the cases of two dimensions. A two dimensional FBHE is introduced as the following,

$$\begin{aligned}
&a(x)u_t - \nabla^2 u = f(x,y,t) && (x,y,t) \in \Omega = (-1,1)\times(0,1)\times(0,1),\\
&u(-1,y,t) = X_{-1}(y,t) && y \in [0,1], t \in [0,1],\\
&u(1,y,t) = X_1(y,t) && y \in [0,1], t \in [0,1],\\
&u(x,0,t) = Y_0(x,t) && x \in [-1,1], t \in [0,1],\\
&u(x,1,t) = Y_1(x,t) && x \in [-1,1], t \in [0,1],\\
&u(x,y,0) = T_0(x,y) && x \in [0,1], y \in [0,1],\\
&u(x,y,1) = T_1(x,y) && x \in [-1,0], y \in [0,1],
\end{aligned} \tag{1}$$

where $X_{-1}(y,t)$, $X_1(y,t)$, $Y_0(x,t)$, $Y_1(x,t)$, $T_0(x,y)$, $T_1(x,y)$ are known functions, $a(x) > 0$ for $x > 0$, $a(x) < 0$ for $x < 0$ and $a(0) = 0$.

The applications of this equation were seen in many problems such as computational fluid dynamics and randomly accelerated particle problems. Moreover, the numerical solution of the forward-backward problem in the case of one dimensional series had been associated through variety of methods including finite differences method (FDM) [30, 12, 28, 25], transformation to a system of first order differential eqs. [14], least square approach[3], finite element method (FEM) [2, 5, 6, 13] and radical basis function (RBF) [26]. However, consideration of this equation in the case of 2D is rarely seen in the literature [19, 24, 33]. All the above mentioned methods have been applied to the FBHE by the use of a domain decomposition method (DDM) [29].

Having mesh-dependant methods such as FEM and FDM were computationally expensive so that it is required both time and spatial domains as well as discretization. This is mentioned to remind the point that in mesh free method, no mesh is needed. Only a set of scattered points in domain and on the boundary to promote a truly mesh free approach is used. Also, based on the radial basis functions (RBFs) in which mesh construction, neither in the domain representation nor in the solution procedure was required. In addition, lack of the mesh would easily applied to the complicated geometric problems. The RBF methods were being given increasingly as evidences that Kansa employed the multiquadric (MQ) to the numerical solution of elliptic and parabolic eqs. [7, 8] in growth. There was no doubt that RBF mesh free method was one of the most developed methods among numerical solution of partial differential eqs. (PDEs) since 1990 [10, 1, 21, 20]. Indeed, it was declared that these basis functions have depended either on the distance between field nodes or some fixed points. They were significantly convenient to use for higher dimensional problems. However, the primary attempt on RBFs was restricted in this series, development of this method have occurred across other approaches such as local weak forms, global weak forms and weak strong forms of mesh free methods [11]. Illustrations for this work have provided to solve FBHE in two- dimensional case numerically by thin plate splines in mesh free method based on the usage of strong form. To generalize the cubic splined to the case of 2D , TPS augmentation by a special polynomial was referred. It was increasing in smoothness optimal properties value [17, 16]. To make a direct consideration, the domain is divided into two subdomains splitting the problem into two standard forward and backward subproblems. While underlying mesh free method, an iterative process had to improve the initial approximate solution on the interface boundary and suit it to the division of subdomains. Apart from that coefficient matrices of the final linear systems of eqs. could be prosecuted on the distance between the points that under selecting the colocation points appropriately, also, the same matrices could be used enough, both subproblems have been solved. Furthermore, the highly structured solvers could be applied to increase the computational efficiency since the matrices remained invariant during the iteration. This paper was organized based on the following characteristics: The domain partitioning and time discretization are considered in section 2. The formulation of the RBF mesh free method for the spatial variable and the solution of the algebraic eqs. will be presented in section 3. Finally, some numerical results will be given in section 5.

2 Time discretization and domain partitioning

Let the domain Ω is divided into two subdomains, $\Omega_1 = (-1,0) \times (0,1) \times (0,1)$ and $\Omega_2 = (0,1) \times (0,1) \times (0,1)$ with the real boundaries Γ_1 and Γ_2 and an artificial boundary $\Gamma_I = \{(x,y) | x = 0,\ 0 < y < 1\}$ in between. This partitioning is dividing the problem (1) into two subproblems as the following,

Subproblem 1:

$$\begin{array}{ll} a(x)u_t - \nabla^2 u = f(x,y,t) & (x,y,t) \in \Omega_1 = (-1,0) \times (0,1) \times (0,1), \\ u(-1,y,t) = X_{-1}(y,t) & y \in [0,1], t \in [0,1], \\ u(x,0,t) = Y_0(x,t) & x \in [-1,0], t \in [0,1], \\ u(x,1,t) = Y_1(x,t) & x \in [-1,0], t \in [0,1], \\ u(x,y,1) = T_1(x,y) & x \in [-1,0], y \in [0,1], \end{array} \tag{2}$$

and

Subproblem 2:

$$\begin{aligned} &a(x)u_t - \nabla^2 u = f(x,y,t) && (x,y,t) \in \Omega_2 = (0,1)\times(0,1)\times(0,1),\\ &u(1,y,t) = X_1(y,t) && y\in[0,1], t\in[0,1],\\ &u(x,0,t) = Y_0(x,t) && x\in[0,1], t\in[0,1],\\ &u(x,1,t) = Y_1(x,t) && x\in[0,1], t\in[0,1],\\ &u(x,y,0) = T_0(x,y) && x\in[0,1], y\in[0,1]. \end{aligned} \tag{3}$$

As previously mentioned, the spatial variable is treated by RBF method and the time derivative is approximated by FDM. Firstly, we discretize the time by the first order backward and forward schemes for subproblems 1 and 2, respectively.
Let $\delta t = 1/M$ and $t_j = j\delta t$ for $j = 0,\ldots,M$. Using a forward difference scheme for the time derivative, we have

$$\frac{\partial u(x,y,t_n)}{\partial t} = \frac{u^{n+1}(x,y)-u^n(x,y)}{\delta t}, \quad n = 0,\ldots,M-1, \tag{4}$$

where $u^n(x,y) = u(x,y,t_n)$. Substituting the approximation (4) into (2), for subproblem 1, we obtain

$$\begin{aligned} &a(x)u^n(x,y) + \delta t\nabla^2 u^n(x,y) = \\ &\qquad a(x)u^{n+1}(x,y) - \delta t f^n(x,y)), && (x,y)\in(-1,0)\times(0,1),\\ &u^n(-1,y) = X_{-1}(y,t_n), && y\in[0,1],\\ &u^n(x,0) = Y_0(x,t_n), && x\in[-1,0],\\ &u^n(x,1) = Y_1(x,t_n), && x\in[-1,0],\\ &u^M(x,y) = T_1(x,y), && x\in[-1,0], y\in[0,1], \end{aligned} \tag{5}$$

where $f^n(x,y) = f(x,y,t_n)$.

Similarly, for subproblem 2, we use a backward difference approximation and substitute it into eq. (3). This conclude in

$$\begin{aligned} &a(x)u^{n+1}(x,y) - \delta t\nabla^2 u^{n+1}(x,y) = \\ &\qquad a(x)u^n + \delta t f^{n+1}(x,y), && (x,y)\in(0,1)\times(0,1),\\ &u^n(1,y) = X_1(y,t_n), && y\in[0,1],\\ &u^n(x,0) = Y_0(x,t_n), && x\in[0,1],\\ &u^n(x,1) = Y_1(x,t_n), && x\in[0,1],\\ &u^0(x,y) = T_0(x,y), && x\in[0,1], y\in[0,1]. \end{aligned} \tag{6}$$

We have constructed two separate subproblems (5) and (6) considering at the time step t_n. In the next section, we will suggest an iterative way to solve the local problems.

3 Meshfree method

Solving subproblems (5) and (6), in the current situation, is not possible, since the boundary condition on the interface,Γ_I, is not available. To treat this issue, we assume initial values $u^n(0,y) = \psi^{0,n}(y)$ on the virtual boundary. Adding these values to the boundary conditions of the subproblems, they can be dealt with underlying RBF method.

Before describing our method, we give some introductory information about RBFs. RBF interpolation uses a set of points $\{X_j\}_{j=1}^N \subset \mathcal{R}^d$ to approximate a given function f by an interpolant $S(X)$ as

$$S(X) = \sum_{j=1}^N \lambda_j \phi(||X - X_j||) + P(X), \quad X\in\mathcal{R}^d \tag{7}$$

where $||.||$ is Euclidean norm, ϕ is a radial function, $P(X)$ is a polynomial and $\{\lambda_j\}_{j=1}^N$ are the unknown coefficients being determined. This is possible using collocation method in all of the points and constructing the coefficient matrix. The function ϕ depends only on the distance between X and the fixed points X_j.

If $P_1,\ldots,P_m$ be the basis of the d-variate polynomials of degree at most q on $\mathcal{R}^d$, then $P(X)$ can be written as

$$P(X)=\sum_{i=1}^{m} c_i P_i(X),$$

where $m=(q-1+d)/(d!\,(q-1)!)$ [22].

Some common used RBFs are listed in Table 1. These basis functions contain a shape parameter ϵ whose values affect the quality of the approximation. There has been a lot of theoretical and practical attempt to find suitable values for these parameters [23, 9, 4, 27], however; finding optimal values for ϵ is still an opening problem. Consequently, RBFs with no shape parameters, are more convenient to use. Some of these RBFs are shown in Table 2. To find more details about these RBFs see [22, 15, 18, 32, 31].

Table 1: Some well-known RBFs with a shape parameter

Type of the basis function	$\phi(r,\epsilon)$
Guassian (GA)	$e^{-(\epsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1+(\epsilon r)^2}$
Inverse Multiquadric (IMQ)	$1/\sqrt{1+(\epsilon r)^2}$
Inverse Quadratic (IQ)	$1/1+(\epsilon r)^2$

Table 2: Some well-known RBFs without a shape parameter

Type of the basis function	$\phi(r)$
Linear	r
Cubic	r^3
Thine Plate Spline (TPS)	$r^2\ln r$

The generalized TPS is introduced as $\phi(r_j)=r_j^{2k}\ln r_j$, k=1, 2, 3, ..., where $r_j=||X-X_j||$. The smoothness order of this function is $2k-1$. The order of used TPS depends on the order of the differential operator. For the forward-backward heat equation, the second order thin plate splines have sufficient smoothness property, that is, $k=2$, $\phi(r_j)=r_j^4\ln r_j$.

3.1 The spatial discretization

Let $\{X_i\}_{i=1}^{N}$ be a set of nodes in $[-1,0]\times[0,1]$ where $\{X_i\}_{i=1}^{N_I}$ are the interior points and the next N_B and N_{IB} points are located on the real and virtual boundaries, respectively, that is, $N=N_I+N_B+N_{IB}$. The function $u^n(x,y)$ in Ω_1 can be approximated by a set of TPS functions, that is augmented by a linear polynomial as the following,

$$u^n(X)\approx\sum_{j=1}^{N}\lambda_j^n\phi(r_j)+\lambda_{N+1}^n x+\lambda_{N+2}^n y+\lambda_{N+3}^n, \qquad (8)$$

where $\phi(r_j)=r_j^4\ln r_j$ and $r_j=||X-X_j||$, $X=(x,y)$, $X_j=(x_j,y_j)$, $j=1,\dots,N$. Applying the collocation method at all points X_i, $i=1,\dots N$, we obtain

$$u^n(X_i)\approx\sum_{j=1}^{N}\lambda_j^n\Phi_{ij}+\lambda_{N+1}^n x_i+\lambda_{N+2}^n y_i+\lambda_{N+3}^n, \qquad (9)$$

where $\Phi_{ij}=\phi(r_{ij})$ and $r_{ij}=||X_i-X_j||$. The following additional conditions are required in order to obtain a square system of eqs.,

$$\sum_{j=1}^{N}\lambda_j^n=\sum_{j=1}^{N}\lambda_j^n x_j=\sum_{j=1}^{N}\lambda_j^n y_j=0. \qquad (10)$$

Assembling eqs. (9) and (10) leads to a linear system of eqs. with the following matrix form

$$[\mathbf{u}]^{\mathbf{n}} = A[\boldsymbol{\lambda}]^{\mathbf{n}}, \tag{11}$$

where $[\mathbf{u}]^{\mathbf{n}} = [u_1^n, u_2^n, \ldots, u_N^n, 0,0,0]^T$, $[\boldsymbol{\lambda}]^{\mathbf{n}} = [\lambda_1^n, \lambda_2^n, \ldots, \lambda_{N+3}^n]^T$ and A is an $(N+3) \times (N+3)$ matrix with

$$A = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1N} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \Phi_{N1} & \cdots & \Phi_{NN} & x_N & y_N & 1 \\ x_1 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & \cdots & y_N & 0 & 0 & 0 \\ 1 & \cdots & 1 & 0 & 0 & 0 \end{bmatrix}$$

This matrix can be written as a summation of four matrices A_I, A_B A_{IB} and A_A as the following,

$$\begin{aligned} A_I^{(1)} &= [a_{ij}, \text{ for } 1 \le i \le N_I,\ 1 \le j \le N+3, \text{ and } 0 \text{ elsewhere}] \\ A_B^{(1)} &= [a_{ij}, \text{ for } N_I + 1 \le i \le N_I + N_B,\ 1 \le j \le N+3, \text{ and } 0 \text{ elsewhere}] \\ A_{IB}^{(1)} &= [a_{ij}, \text{ for } N_I + N_B + 1 \le i \le N,\ 1 \le j \le N+3, \text{ and } 0 \text{ elsewhere}] \\ A_A^{(1)} &= [a_{ij}, \text{ for } N + 1 \le i \le N+3,\ 1 \le j \le N+3, \text{ and } 0 \text{ elsewhere}] \end{aligned} \tag{12}$$

where the superscripts indicate the association of the matrices with subproblem 1.

To deal with eqs. (5), we use (8) to approximate $\nabla^2 u^n(X)$ as

$$\nabla^2 u^n(X) \approx \sum_{j=1}^{N} \lambda_j^n \nabla^2 \phi\left(\left|\left|X - X_j\right|\right|\right). \tag{13}$$

Applying the collocation method at all nodes for eqs. (5) and using (11), (12) and (13), the following eqs. are derived in a matrix form,

$$E^{(1)}[\boldsymbol{\lambda}]^{\mathbf{n}} = D^{(1)}[U_I^{(1)}]_I^{n+1} + [U_B^{(1)}]^n + [U_{IB}^{(1)}]^n - \delta t[\mathbf{F}^{(1)}]^{\mathbf{n}}, \qquad n = M-1, \ldots, 0, \tag{14}$$

where

$$E^{(1)} = DA_I^{(1)} + \delta t(\nabla^2 A_I^{(1)}) + A_B^{(1)} + A_{IB}^{(1)} + A_A^{(1)}, \tag{15}$$

$D^{(1)}$ is an $(N+3) \times (N+3)$ diagonal matrix with $D_{ii}^{(1)} = a(x_i)$, for $i = 1, \ldots, N_I$, and zero elsewhere, $\left(\nabla^2 A_I^{(1)}\right)_{ij}$ is derived by evaluating $\nabla^2 \phi(||X - X_j||)$ at interior points $X = X_i$,

$$[U_I^{(1)}]^n = \begin{bmatrix} D^{(1)} \mathbf{u}_\mathbf{I}^\mathbf{n} \\ 0 \end{bmatrix}, \quad [U_B^{(1)}]^n = \begin{bmatrix} 0 \\ \mathbf{u}_\mathbf{B}^\mathbf{n} \\ 0 \end{bmatrix}, \quad [U_{IB}^{(1)}]^n = \begin{bmatrix} 0 \\ \mathbf{u}_\mathbf{IB}^\mathbf{n} \\ 0 \end{bmatrix},$$

$$[\mathbf{F}^{(1)}]^{\mathbf{n}} = [f_1^n, \ldots, f_{N_I}^n, 0, \ldots, 0]^T,$$

where $\mathbf{u}_\mathbf{I}^\mathbf{n}$, $\mathbf{u}_\mathbf{B}^\mathbf{n}$ and $\mathbf{u}_\mathbf{IB}^\mathbf{n}$ denote the associated solution vectors with the interior nodal values, real boundary and virtual boundary values respectively. Also $\mathbf{u}_\mathbf{B}^\mathbf{n}$ is representing the real boundary values, that is calculated by the fuctions $X_1(y, t_n)$, $Y_0(x, t_n)$, $Y_1(x, t_n)$ and the function $\psi^{0,n}(y)$ is used to evaluate the interface boundary values $\mathbf{u}_\mathbf{IB}^\mathbf{n}$ in the starting point of iteration at each time step.

Solving the linear system (14) for $[\boldsymbol{\lambda}]^{\mathbf{n}}$ and using (8), the solution of subproblem (6) can be found.

Similarly, subproblem 2 can be dealt with underlying method. To this end, the same number of nodes may be used to obtain a similar system of eqs. as the following,

$$E^{(2)}[\gamma]^{n+1} = [U_I^{(2)}]^n + [U_B^{(2)}]^{n+1} + [U_{IB}^{(2)}]^{n+1} + \delta t[\mathbf{F}^{(2)}]^{\mathbf{n+1}}, \tag{16}$$

where

$$E^{(2)} = D^{(2)} A_I^{(2)} - \delta t(\nabla^2 A_I^{(2)}) + A_B^{(2)} + A_{IB}^{(2)} + A_A^{(2)}. \tag{17}$$

$$[U_I^{(2)}]^n = \begin{bmatrix} D^{(2)}\mathbf{u_I^n} \\ 0 \end{bmatrix}, \quad [U_B^{(2)}]^n = \begin{bmatrix} 0 \\ \mathbf{u_B^n} \\ 0 \end{bmatrix}, \quad [U_{IB}^{(2)}]^n = \begin{bmatrix} 0 \\ \mathbf{u_{IB}^n} \\ 0 \end{bmatrix},$$

$$[\mathbf{F^{(2)}}]^{\mathbf{n}} = [f_1^n, \dots, f_{N_I}^n, 0, \dots, 0]^T.$$

All the matrices with superscript (2) have the same duties as those of the matrices with superscript (1) in subproblem 1.

It should be noted that the vectors $\mathbf{u_I}$ and $\mathbf{u_B}$ are local solutions and different in the subproblems but, the interface boundary solution $\mathbf{u_{IB}}$ is the same for the two subproblems.

3.2 The iterative method

Before starting the iterations, firstly, we need to obtain $[\boldsymbol{\lambda}]^{\mathbf{M}}$ and $[\boldsymbol{\lambda}]^{\mathbf{0}}$, respectively, for the subproblems (1) and (2). This can be carried out by applying the initial conditions and using (8). To do it, we need to solve the following system of eqs.:

$$T_1(x_i, y_i) = \sum_{j=1}^{N} \lambda_j^M \phi(r_{ij}) + \lambda_{N+1}^M x_i + \lambda_{N+2}^M y_i + \lambda_{N+3}^M, \ i = 1, \dots, N.$$

Similarly, $T_0(x, y)$ can be used for the other subproblem to find $[\boldsymbol{\lambda}]^{\mathbf{0}}$. Now inserting the initial approximate solution $\psi^{0,n}$ on the interface, the linear systems (14) and (16) can be readily solved for the next time step according to the sort of the subproblem either in a forward or a backward manner.

Having solved the local problems at the nth time step, the interface boundary solution may be updated. This can be accomplished by interpolating the solution function using a number of interior points close to the virtual boundary in the two subdomains. Let $\{(x_{i_j}, y_{i_j})\}_{j=1}^{L}$ be a set of the interior points close to the virtual boundary and $\{(x_{i_j}, y_{i_j})\}_{j=L+1}^{L+n_B}$ be the interface boundary points. In fact $x_{i_j} = 0$, for $j = L+1, \dots, L+n_B$. We suggest interpolating $u^n(x, y)$ at the above points as

$$u^n(x, y) \approx \sum_{j=1}^{L+n_B} \alpha_j^n \phi(r_j), \tag{18}$$

where ϕ is the TPS function, and α_j^ns are time-dependent coefficients to be found. To solve the above interpolation problem, the following combination use of an interpolation and the collocation is suggested.

I) The interpolation conditions may be imposed on the internal points whose corresponding solutions are available, that is,

$$u^n(x_{i_k}, y_{i_k}) = \sum_{j=1}^{L+n_B} \alpha_j^n \phi(r_{kj}), \quad k = 1, \dots, L, \tag{19}$$

where $r_{kj} = \sqrt{(x_{i_k} - x_{i_j})^2 + (y_{i_k} - y_{i_j})^2}$.

II) Since, in problem (1), $a(x) = 0$ for $x = 0$, the interface boundary solution satisfies

$$-\nabla^2 u^n(x, y) = f(x, y, t_n). \tag{20}$$

Using (18), $\nabla^2 u^n(x, y)$ can be approximated by

$$\nabla^2 u^n(x, y) \approx \sum_{j=1}^{L+n_B} \alpha_j^n \nabla^2 \phi(r_j), \tag{21}$$

Substituting (21) into (20) and applying the collocation method at the virtual boundary points, we obtain

$$-\sum_{j=1}^{L+n_B} \alpha_j^n \nabla^2 \phi(r_{kj}) = f(x_{i_k}, y_{i_k}, t_n), \quad k = L+1, \dots, L+n_B. \tag{22}$$

Eqs. (19) and (22) form a square system whose solution will determine the unknown coefficients α_j^n's followed by using (18) to evaluate $u^n(x_{i_k}, y_{i_k})$, for $k = L+1, \dots, L+n_B$ and updating $\mathbf{u_{IB}^n}$. The new solution can be used for the next iteration and this procedure continues until a desired accuracy is achieved.

There are some computational advantages with using eqs. (14) and (16). Since all the matrices in (15) and (17), except the diagonal matrices $D^{(1)}$ and $D^{(2)}$ which, in fact, do not need to be explicitly constructed, the coefficient matrices $E^{(1)}$ and $E^{(2)}$ are the same and they can be built only once for the two subproblems. In fact, if a constant δt is used, these matrices do not change during the time marching, and also in the iterative procedure. As a result, the systems (14) and (16) can be dealt with efficient solvers such as LU decomposition method.

4 Stability and Convergence

First we write the time disretization for subproblem Ω_1 in point (x, t_{n+1}),

$$a(x)\frac{u^{n+1}-u^n}{k} - \triangle u^n = f(x, t_n) + R, \quad n = M, \ldots, 0, \tag{23}$$

which $k = \triangle t$ and $|R| < Ck$ that C is a positive constant. By simplification We can write eq. ?? in the form

$$a(x)u^n + k \triangle u^n = a(x)u^{n+1} - kf(x, t_n) + kR, \quad n = M, \ldots, 0, \tag{24}$$

By eliminating the small term R we have

$$a(x)U^n + k \triangle U^n = a(x)U^{n+1} - kf(x, t_n), \quad n = M, \ldots, 0, \tag{25}$$

Now by these equations that represent the time discrete scheme for the subproblem Ω_1 we present two theorems for stability and convergence due to time.

Theorem 1 *The time discrete scheme (25) is stable in weighted L_2 norm.*

Proof. Let define the exact and approximate solutions of (25) respectively by U^{n+1} and $\widetilde{U}^{n+1}$ and define the roundoff error by $e^{n+1} = U^{n+1} - \widetilde{U}^{n+1}$. Now by applying them, we have

$$a(x)e^n + \delta t \triangle e^n = a(x)e^{n+1}. \tag{26}$$

Multiplying both sides of eq. (26) by e^n and integrating over Ω_1, yields

$$(a(x)e^n, e^n) + \delta t(\triangle e^n, e^n) = (a(x)e^n, e^{n+1}),$$

which can be rewritten as

$$(a(x)e^n, e^n) - \delta t(\nabla. e^n, \nabla. e^n) = (a(x)e^n, e^{n+1}),$$

By inequality $\delta t(\nabla. e^n, \nabla. e^n) > 0$ and using the properties of norms, we have

$$\| e^n \|^2_{a(x)} < \| e^{n+1} \|_{a(x)} \| e^n \|_{a(x)}.$$

Therefore,

$$\| e^0 \|_{a(x)} < \| e^1 \|_{a(x)} < \ldots < \| e^{M+1} \|_{a(x)}.$$

which is the desired result.

Theorem 2 *Let u^n and U^n be the exact solution and approximate solution respectively for equation of subdomain Ω_1 with considered initial and boundary conditions, then the time discrete solution is convergent with order $O(k)$ in weighted L_2 norm.*

Proof. By the same method for theorem 1 and by helping of article [20], we can prove the above theorem.

5 Numerical results

In this section, we demonstrate the effectiveness of the proposed method by giving some experimental results. The maximum error and the root-mean-square error (RMSE) are used to measure the accuracy of the numerical solutions as the following,

$$Max\ error = \max_{j=1}^{N} |\hat{u}_j - u_j|, \qquad RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N} (\hat{u}_j - u_j)^2},$$

where N is the number of nodes, u_j and $\hat{u}_j$ denote the exact and the approximate solutions at the jth node.

Example 1 *Consider eqs. (1) with*

$$f(x, y, t) = x(2t-1)(1-x^2)(y^2-y) + 2(t^2-t)(y^2-y) - 2(1-x^2)(t^2-t) - 4, \tag{27}$$

with the initial conditions

$$u(x,y,0) = x^2 + y^2, \quad u(x,y,1) = x^2 + y^2,$$

and the dirichlet condition for rectangular domain is

$$u(x,y,t) = (t^2 - t)(1 - x^2)(y^2 - y) + x^2 + y^2, \qquad (x,y) \in \partial\Omega, \ 0 < t < 1.$$

The exact solution of (1), in this case, is given by:

$$u(x,y,t) = (t^2 - t)(1 - x^2)(y^2 - y) + x^2 + y^2. \tag{28}$$

Table 3: Error values of the new method

$N^{(1)}$	$N^{(2)}$	M	k	$Max\ error$	MSRE
26	26	10	9	1.21E-3	2.87E-4
		20	8	9.56E-4	2.40E-4
		30	8	8.98E-4	2.12E-4
65	65	20	16	5.119E-4	2.56E-4
		40	19	4.42E-4	1.39E-4
		60	19	4.35E-4	1.21E-4
145	145	40	20	5.37E-4	1.76E-4
		40	24	2.72E-4	1.26E-4
		40	28	2.19E-4	1.11E-4

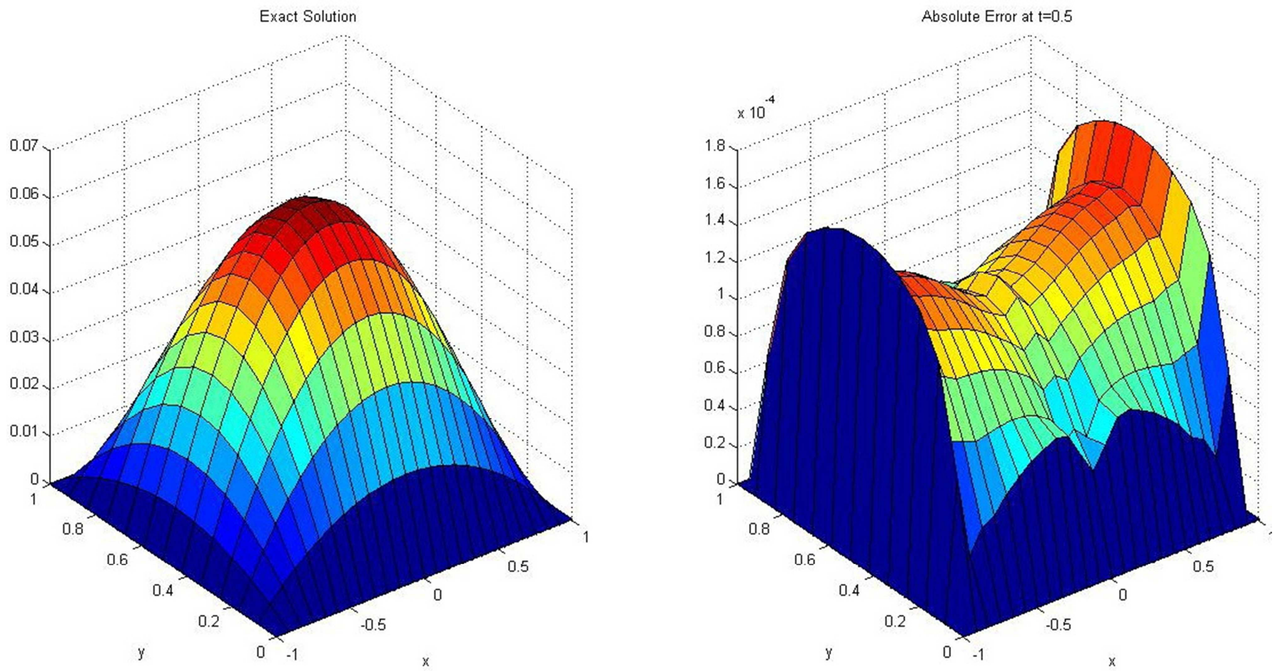

Figure 1: Configerations of the exact solution and the absolute errors with $N^{(1)} = 145$, $N^{(2)} = 145$, $M = 80$ and $k = 28$ at $t = 0.5$ for the example 1.

We have solved the above problem and have obtained the numerical solution for different numbers of points, time steps and iterations which represented by $N^{(i)}$, $i = 1,2$, M and k, respectively. The error values are measured by Max error and RMSE are presented in Tables 3. Also the exact and approximate solutions are compared in Figure 1 and the absolute error for t=0.2, 0.4, 0.6 and 0.8 are shown in Figure 2.

Example 2 *Now consider eqs. (1) with*

$$f(x,y,t) = x\cos(x)\sin(y)\cos(t) + 2\cos(x)\sin(y)\sin(t), \tag{29}$$

with the initial conditions

$$u(x,y,0) = x^2, \quad u(x,y,1) = \cos(x)\sin(y)\sin(1) + x^2,$$

and the dirichlet condition is

$$u(x,y,t) = \cos(x)\sin(y)\sin(t) + x^2, \qquad (x,y) \in \partial\Omega,\ 0 < t < 1.$$

The exact solution of (1), in this case, is given by:

$$u(x,y,t) = \cos(x)\sin(y)\sin(t). \tag{30}$$

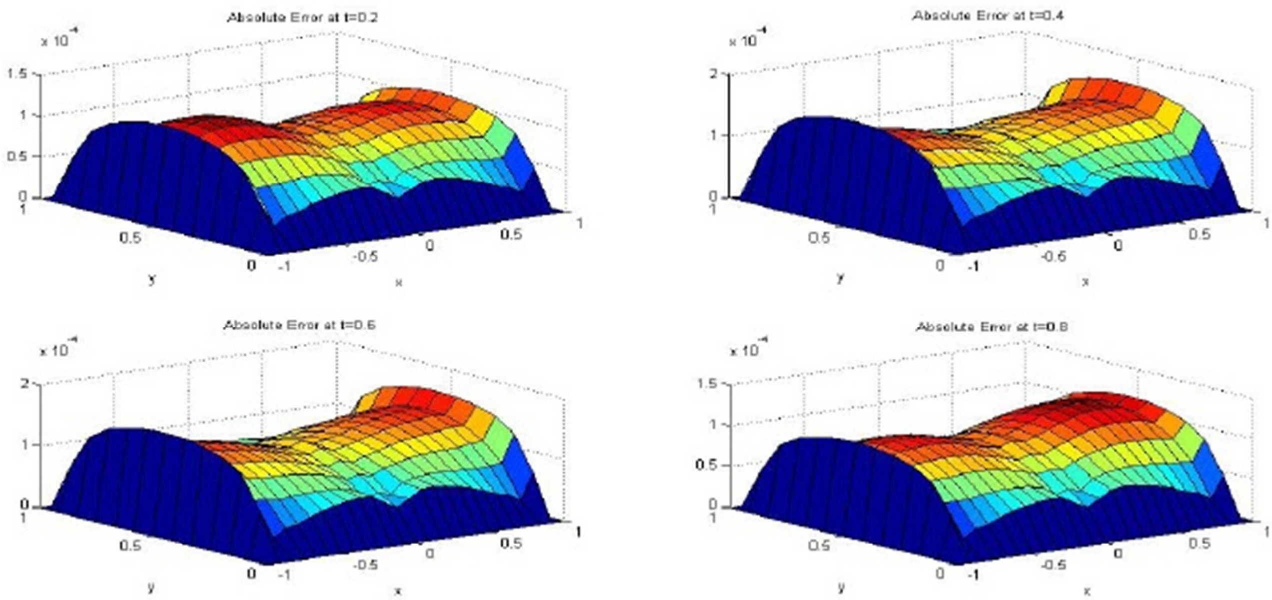

Figure 2: Configerations of the absolute errors with $N^{(1)} = 145$, $N^{(2)} = 145$, $M = 80$ and $k = 28$ at $t = 0.5$ for the example 1.

In this example some numerical solutions have been collected in Table 4 and 5 for two different subdomais and in Figure 3 the exact and estimated solutions are campared. in Figure 4 have shown the absolute errors for t=0.2, 0.4, 0.6 and 0.8 like example 1.

Table 4: Error values of the new method

$N^{(1)}$	$N^{(2)}$	M	k	$Max\ error$	MSRE
26	26		6	6.90E-3	1.30E-3
37	37	10	9	2.20E-3	5.19E-4
50	50		11	1.10E-3	2.29E-4
65	65		14	7.21E-4	1.35E-4
82	82	20	17	4.38E-4	8.82E-5
122	122		24	2.28E-4	6.87E-5
			26	6.55E-4	1.34E-4
170	170	30	28	2.89E-4	7.68E-5
			30	1.61E-4	5.20E-5
			26	6.24E-4	1.18E-4
170	170	40	28	2.54E-4	6.00E-5
			30	1.96E-4	3.87E-5

One can observe that the method offers a high accurate solution while using a reasonable number of points. Moreover, increasing the number of collocation nodes leads to improve the accuracy of the solutions which show the convergence of our method.

It should be noted that the number of interpolation points, L, in (18) has been taken the interior points in the whole domain as twenty percent of the total number. Also the initial values $\psi^{0,n}(y) = 0$ were used as the interface boundary solution for starting the iterations.

Table 5: Error values of the new method

$N^{(1)}$	$N^{(2)}$	M	k	$Max\ error$	MSRE
50	50	10	9	3.20E-2	5.70E-3
79	79		15	5.20E-3	9.42E-4
113	113		17	4.33E-3	6.58E-4
79	79	25	16	4.70E-3	7.96E-4
113	113		17	3.51E-3	2.03E-4
152	152		19	1.20E-3	9.96E-5
251	251	30	40	8.82E-4	1.21E-4
			44	8.40E-4	7.49E-5
			48	8.21E-4	5.61E-5
251	251	60	40	8.81E-4	1.03E-5
			44	8.36E-4	5.85E-5
			48	8.15E-4	4.42E-5

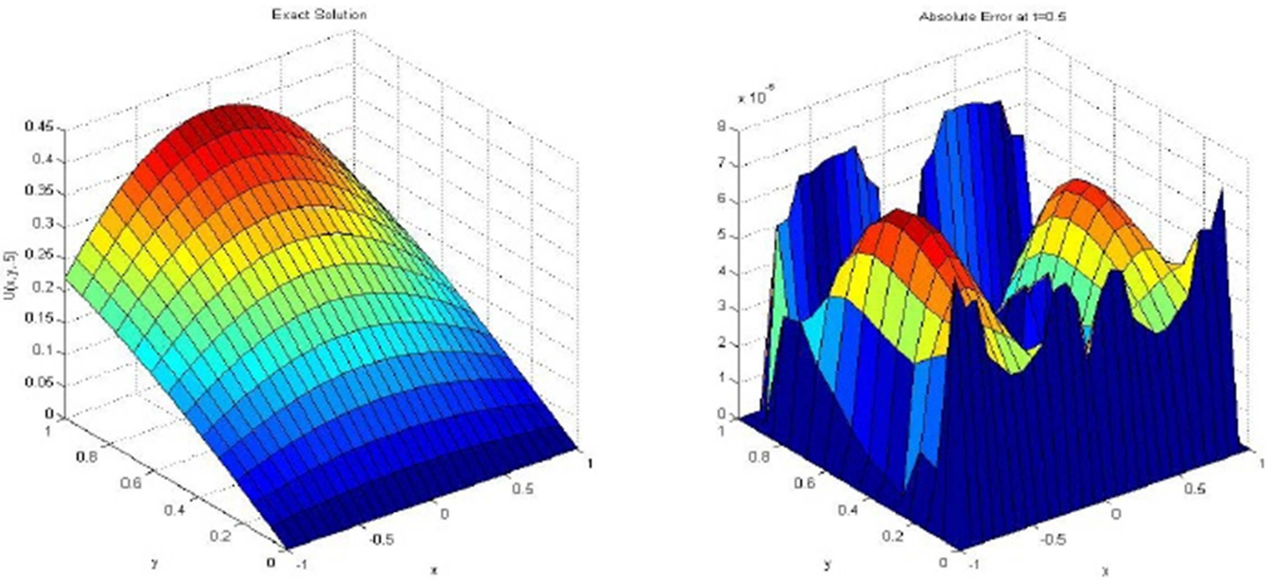

Figure 3: Configerations of the exact solution and the absolute errors with $N^{(1)} = 170$, $N^{(2)} = 170$, $M = 50$ and $k = 42$ at $t = 0.5$ for the example 2.

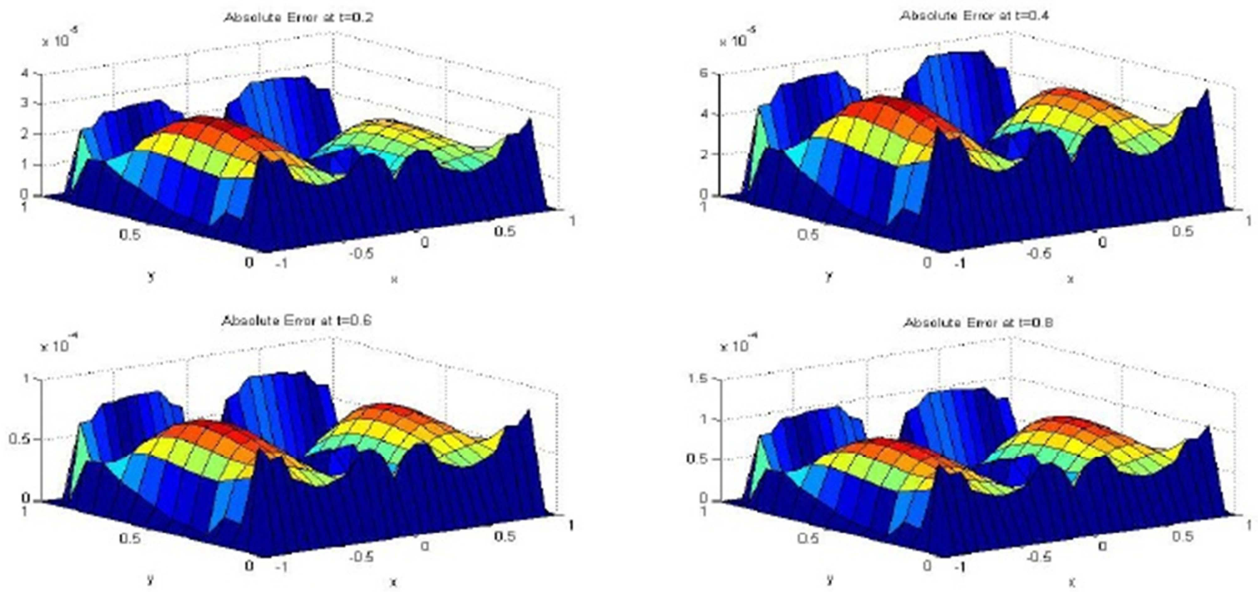

Figure 4: Configerations of the absolute errors with $N^{(1)} = 170$, $N^{(2)} = 170$, $M = 50$ and $k = 42$ at $t = 0.2, 0.4, 0.6$ and 0.8 for the example 2.

6 Conclusion

At the present study , applying TPS radial basis function based on a non-overlaping domain decomposition method to the numerical solution of a forward-backward heat equation in two dimensional cases was very straightforward. The work was accomplished by splitting the problem into two standard forward and backward subproblems, solving each by a meshless method and using an iterative approach to make interaction between the subproblems. The obtained results were sensible. The scope of geometric- dependance of the matrices and their invariance during time marching have greatly shown improvement in efficiency.

References

[1] A. J. Ferreira, P. A. Martins and C. M. Roque. *"Solving time-dependent engineering problems with multiquadrics*". Journal of Sound and Vibration. Vol. 280, pp. 595- 610. (2005).

[2] A. K. Aziz and J. L. Liu. *"A Galerkin method for the forward-backward heat eq"*. Math. Comp. Vol. 56, pp. 35-44. (1991).

[3] A. K. Aziz and J. L. Liu. *"A weighted least squares method for the backward-forward heat eq"*. SIAM J. Numer. Anal. Vol. 28, pp. 156-167. (1991).

[4] C. K. Lee, X. Liu and S. C. Fan. "*Local multiquadric approximation for solving boundary value problems*". Computational Mechanics. Vol. 30, pp. 396-409. (2003).

[5] D. A. French. "*Continuous Galerkin finite element methods for a forward-backward heat eq*". Numer. Methods Partial Differential eq.s. Vol. 15, pp. 257-265. (1999).

[6] D. A. French. *"Discontinuous Galerkin finite element methods for a forward-backward heat eq*". Appl. Numer. Math. Vol. 28, pp. 37-44. (1998).

[7] E. J. Kansa. *"Multiquadrics, a scattered data approximation scheme with applications to computational fluid dynamics-I. Surface approximations and partial derivatives estimates"*. Comput. Math. Appl. Vol. 19, pp. 127-145. (1990).

[8] E. J. Kansa. *"Multiquadrics, a scattered data approximation scheme with applications to computational fluid dynamics-II. Solutions to parabolic, hyperbolic and elliptic partial differential eq.s"*. Comput. Math. Appl. Vol. 19, pp. 147-161. (1990).

[9] G. E. Fasshuaer. *"On choosing optimale shape parameters for RBF approximation"*. Numerical Algorithms. Vol. 45, No. 1, pp. 345- 368. (2007).

[10] G. E. Fasshuaer. *"Solving differential eq.s with radial basis functions: multilevel methods and smoothing*". Advances in Computational Mathematics. Vol. 11, No.2, pp. 139-159. (1999).

[11] G. U. Liu and Y. T. Gu. "*An introduction to meshfree methods and their programming"*. Springer, Netherlands. (2005).

[12] H. D. Han and D. S. Yin. *"A non-overlap domain decomposition method for the forward-backward heat eq*". J. Comput. Appl. Math. Vol. 159, pp. 35-44. (2003).

[13] H. Lu. *"Galerkin and weighted Galerkin methods for a forward-backward heat eq*". Numer. Math. Vol. 75, pp. 338-356. (1997).

[14] H. Lu and J. Maubach. *"A finite element method and variable transformations for a forward-backward heat eq"*. Numer. Math. Vol. 81, pp. 249-272. (1998).

[15] H. Mesgarani , Y. Azari. *"Numerical investigation of Fredholm integral equation of the first kind with noisy data"*. Mathematical Sciences. Vol. 13, No. 3, pp. 267- 278. (2019).

[16] I. Boztosunand etal. *"Thin-Plate Spline Radial Basis Fucntion Scheme for Advection-Diffusion problems"*. Electronic Journal of Boundary Elements, Vol. Beteq (2001), No. 2, pp.267-282. (2002).

[17] J. Duchon. *"Splines minimizing rotation-invariant semi-norms in Sobolev spaces"*. In: Schempp W, Zeller K. (eds) Constructive Theory of Functions of Several Variables, Lecture Notes in Mathematics. Vol. 571, pp. 85-100, Springer, Berlin, Heidelberg. (1977).

[18] J. Rashidinia, Y. Azari and Gh. Garmanjani. *"Numerical solution of the nonlinear Fredholm integral equations of the second kind by radial basis functions"*. (2012).

[19] J. Sun and X. L. Cheng. *"Iterrative methods for a forward-backward heat eq. in two-dimension*". Appl. Math. J. Chinese Univ. Vol. 25, No. 1, pp. 101-111. (2010).

[20] M. Dehghan and A. Shokri. *"A numerical method for solution of the two-dimensional sine-Gordon eq. using the radial basis functions"*. Math. Comput. Simul. Vol. 79, No. 3, pp. 700-715. (2008).

[21] M. A. Golberg, C. S. Chen and S. R. Karur. *"Improved multiquadrics approximation for partial differential eq.s"*. Engineering Analysis with Boundary Elements. Vol. 18, pp. 9-17. (1996).

[22] M. D. Buhmann. *"Radial Basis Functions: Theory and Implementations"*. Cambridge University Press, Cambridge. (2003).

[23] R. E. Carlson and T. A. Foley. *"The parameter R2 in multiquadric interpolation"*. Comput. Math. Appl. Vol. 21, pp. 29- 42. (1991).

[24] S. Banei, and K. Shanazari. *"A meshfree method with a non-overlapping domain decomposition method based on TPS for solving the forward-backward heat equation in two-dimension*". Numer. Algorithms. Vol. 85, No. 1, pp. 1-22. (2020).

[25] S. Banei, and K. Shanazari. *"A non-iterative domain decomposition method for the forward-backward heat equation"*. Iranian Journal of Numerical Analysis and Optimization. Vol. 10, No. 2, pp. 155-167. (2020).

[26] S. Banei, and K. Shanazari. *"Solving the forward-backward heat equation with a non-overlapping domain decomposition method based on multiquadric RBF meshfree method"*, Computational Methods for Differential Equations. (2020).

[27] S. Rippa. *"An algorithm for selecting a good value for the parameter c in radial basis function interpolation*". Adv. Comput. Math. Vol. 11, pp. 193- 210. (1999).

[28] V. Vanaja and R. B. Kellogg. *"Iterative methods for a forward-backward heat equation"*. SIAM J. Numer. Anal. Vol. 27, pp. 622-635. (1990).

[29] W. Beatson, J. B. Cherrie and T. Mouat. *"Fast solution of the radial basis function interpolation methods: domain decomposition methods".* SIAM J. Sci. Comp. Vol. 22, pp. 1717-1740. (2000).

[30] X. L. Cheng and J. Sun. *"Iterative methods for the forward-backward heat eq".* J Comput. Math. Vol. 23, pp. 419-424. (2005).

[31] Y. Azari, Gh. Garmanjani, H. Rabiei. *"Numerical of nonlinear Sine-Gordon equation with local RBF-based finite difference collocation method*". (2013).

[32] Y. Esmaeelzade Aghdam, H. Safdari, Y. Azari, H. Jafari, D. Baleanu. *"Numerical investigation of space fractional order diffusion equation by the Chebyshev collocation method of the fourth kind and compact finite difference scheme".* Discrete Continuous Dynamical Systems-S. (2018).

[33] S. Banei, and K. Shanazari. *"On the convergence analysis and stability of the RBF-adaptive method for the forward-backward heat problem in 2D".* Applied Numerical Mathematics. Vol 159, pp. 297-310. (January 2021).