# A Meshless IRBFN-based Method for Transient Problems

L. Mai-Cao[1] and T. Tran-Cong[2]

**Abstract:** The Indirect Radial Basis Function Network (IRBFN) method has been reported to be a highly accurate tool for approximating multivariate functions and solving elliptic partial differential equations (PDEs). The present method is a truly meshless method as defined in [Atluri and Shen (2002a)]. A recent development of the method for solving transient problems is presented in this paper. Two numerical schemes combining the IRBFN method with different time integration techniques based on either fully or semi-discrete framework are proposed. The two schemes are implemented making use of Hardy's multiquadrics (MQ) and Duchon's thin plate splines (TPS). Some example problems are solved by the present method, and the results compare favorably in terms of accuracy and efficiency with those from other numerical methods such as finite difference (FDM), finite element (FEM), boundary element (BEM) and the Direct Radial Basis Function Network (DRBFN) methods.

**keyword:** meshless method; radial basis functions; IRBFN method; time-dependent PDEs.

## 1 Introduction

In recent years meshless methods based on radial basis functions (RBFs) have increasingly attracted much attention from researchers not only for interpolating multivariate scattered data and approximating functions but also for solving partial differential equations (PDEs). The idea of using RBFs for solving PDEs was first proposed in [Kansa (1990a,b)] where a global MQ scheme was used together with the collocation method to solve parabolic, hyperbolic and elliptic PDEs. This method is hereby referred to as the Direct Radial Basis Function Network (DRBFN) method. Since its introduction, the method has been widely applied to various transient problems. In [Moridis and Kansa (1994)] a hybrid scheme combining MQ with the numerical inversion of Laplace transforms was proposed for solving linear or linearized time-dependent PDEs. Some applications using MQ and TPS for heat transfer problems and linear advection-diffusion equations were reported in [Zerroukat, Power, and Chen (1998); Zerroukat, Djidjeli, and Charafi (2000)]. RBF-based methods were used for solving natural convection, porous media, and solid-liquid system problems in [Sarler, Perko, Chen, and Kuhn (2001); Perko, Chen, and Sarler (2001); Kovacevic, Poredos, and Sarler (2003); Sarler, Perko, and Chen (2004)]. In [Hon, Cheung, Mao, and Kansa (1999)] a computational algorithm using MQ was devised to solve the shallow-water equations. It is noted that, the global MQ and TPS are ranked to be the most accurate for scattered data approximation [Franke (1982)]. The accuracy of MQ, however, is influenced by the so-called shape parameter whose optimal value depends on data [Carlson and Foley (1991); Rippa (1999)].

In addition to the DRBFN method, other meshless approaches to solving PDEs based on RBFs have been developed in the last decade. Unlike Kansa's method where the RBF coefficient matrix is unsymmetric, the Hermite-type collocation method proposed in [Fasshauer (1996)] yields a symmetric positive definite coefficient matrix which is guaranteed to be non-singular. The meshless local Petrov-Galerkin (MLPG) was developed in [Atluri and Zhu (1998a,b)] where compactly supported RBFs among others can be used as trial functions. Various MLPG methods were compared and shown to be promising contenders to the FEM in [Atluri and Shen (2002b)]. The power and flexibility of the MLPG approach were further demonstrated recently in [Atluri, Han, and Rajendran (2004); Han and Atluri (2004)]. Another approach investigated in [Chen, Golberg, and Rashed (1998); Golberg and Chen (1999, 2001)] is to combine the method of fundamental solutions (MFS) with techniques from the dual reciprocity method (DRM) where the MFS is employed to find the homogeneous solution and the ap-

[1] Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba Qld 4350, Australia.

[2] Corresponding author, Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba Qld 4350, Australia, email: trancong@usq.edu.au, Fax: +61 7 46312526.

proximation of particular solutions can be found in the context of the DRM using RBFs. In [Hon (2002)] the quasi-radial basis function method was proposed where the quasi-interpolation [Beatson and Powell (1990)] and RBFs are combined so that the ill-conditioning problem resulted from a global RBFs scheme is eliminated.

Recently, a new method, namely the Indirect Radial Basis Function Network (IRBFN) was proposed in [Mai-Duy and Tran-Cong (2001a, 2003)] for approximating functions and their derivatives, and solving differential equations. In the DRBFN method, a function is first approximated by the RBFN and its derivatives are then calculated by differentiating such closed form RBFN approximation. In the IRBFN method, on the other hand, the highest derivative is first decomposed into radial basis functions. Lower derivatives and the function itself are then successively obtained via symbolic integrations. Recently, it was discovered that this indirect idea is similar to a Chebyshev method for solving boundary value problems proposed in [Zebib (1984)]. The IRBFN method is, however, unique in its way of using the basis functions and computing the constants of integration as described later in this paper.

As far as the IRBFN method is concerned, the method has been successfully applied for function approximations and solving elliptic PDEs as well as steady Navier-Stokes equations. It was reported in [Mai-Duy and Tran-Cong (2003)] that the indirect method performs better than the direct method in terms of accuracy for approximating both functions and their derivatives. Furthermore, in [Mai-Duy and Tran-Cong (2001a,b)] the indirect method showed its superiority over the direct method in solving 1-dimensional and 2-dimensional Poisson's equations.

In this paper, further developments of the IRBFN method for solving transient problems are presented. In particular, the method is extended to solve time-dependent parabolic PDEs, hyperbolic PDEs, and advection-diffusion equations. The remaining of this paper is organized as follows. Firstly, the approximation of a function and its derivatives by the IRBFN method in a time-dependent framework is formulated. Two numerical schemes based on the IRBFN method for solving transient problems are then presented, followed by some numerical examples for verification purposes. Accuracy and efficiency comparisons between the two schemes based on both MQ and TPS RBFs with other numerical methods, as well as sensitivity studies of the MQ-based schemes to the network parameters are also discussed.

## 2 Function approximation by the IRBFN method for time-dependent problems

The approximation $\tilde{u}(\mathbf{x},t)$ of a function $u(\mathbf{x},t)$ can be written as a linear combination of $N$ basis functions

$$\tilde{u}(\mathbf{x},t)=\sum_{i=1}^{N} w_i(t)\,\phi_i(\mathbf{x})=\boldsymbol{\phi}^T(\mathbf{x})\,\boldsymbol{w}(t), \tag{1}$$

where $\boldsymbol{\phi}(\mathbf{x})=[\phi_1(\mathbf{x}),\ldots,\phi_N(\mathbf{x})]^T$ is a given set of basis functions, and $\boldsymbol{w}(t)=[w_1(t),\ldots,w_N(t)]^T$ is the corresponding set of $N$ weight coefficients.

Given a set of $M$ data points and the corresponding nodal values of the function at certain point in time $t$, $\boldsymbol{U}(t)=[U_1(t),U_2(t),...,U_M(t)]^T$, the collocation method is implemented by applying (1) at every data point to give a system of equations as follows

$$\tilde{u}(\mathbf{x}_j,t)=\sum_{i=1}^{N} w_i(t)\,\phi_i(\mathbf{x}_j)=\boldsymbol{\phi}^T(\mathbf{x}_j)\,\boldsymbol{w}(t), \quad j=1,2,...,M, \tag{2}$$

or in a compact form

$$\boldsymbol{U}(t)=\boldsymbol{\Phi}\boldsymbol{w}(t), \tag{3}$$

where

$$\boldsymbol{\Phi}=\begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \ldots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \ldots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_M) & \phi_2(\mathbf{x}_M) & \ldots & \phi_N(\mathbf{x}_M) \end{bmatrix}. \tag{4}$$

The weight coefficients $\boldsymbol{w}(t)$ can be then determined by

$$\boldsymbol{w}(t)=\boldsymbol{\Phi}^{-1}\boldsymbol{U}(t). \tag{5}$$

Substituting (5) into (1) yields

$$\tilde{u}(\mathbf{x},t)=\boldsymbol{\phi}^T(\mathbf{x})\boldsymbol{\Phi}^{-1}\boldsymbol{U}(t)=\boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{U}(t), \tag{6}$$

where $\boldsymbol{\psi}^T(\mathbf{x})=\boldsymbol{\phi}^T(\mathbf{x})\boldsymbol{\Phi}^{-1}$ is analogous to a "shape function" in the FEM context. The first and higher derivatives of the function in terms of spatial variables can be approximated by

$$\tilde{u}_{,j}(\mathbf{x},t) = \boldsymbol{\phi}^T_{,j}(\mathbf{x})\boldsymbol{\Phi}^{-1}\boldsymbol{U}(t), \tag{7}$$

$$\tilde{u}_{,j...l}(\mathbf{x},t) = \boldsymbol{\phi}^T_{,j...l}(\mathbf{x})\boldsymbol{\Phi}^{-1}\boldsymbol{U}(t), \tag{8}$$

where

$$\begin{aligned}\boldsymbol{\phi}_{,j}(\mathbf{x}) &= [\phi_{1,j}(\mathbf{x}),\phi_{2,j}(\mathbf{x}),\ldots,\phi_{N,j}(\mathbf{x})]^T,\\ \boldsymbol{\phi}_{,j\ldots l}(\mathbf{x}) &= \left[\phi_{1,j\ldots l}(\mathbf{x}),\phi_{2,j\ldots l}(\mathbf{x}),\ldots,\phi_{N,j\ldots l}(\mathbf{x})\right]^T.\end{aligned}$$

In the DRBFN method, $\phi_i$'s are chosen to be radial basis functions. The IRBFN method, on the other hand, starts from the highest derivative of $\phi_i$

$$\phi_{i,j\ldots l}(\mathbf{x}) = \varphi_i(\mathbf{x}) = \varphi(||\mathbf{x}-\mathbf{x}_i||). \tag{9}$$

For all examples in this paper, the highest order derivative of interest is the second derivative

$$\phi_{i,jk}(\mathbf{x}) = \varphi_i(\mathbf{x}), \tag{10}$$

and the chosen basis function is Hardy's multiquadrics

$$\varphi_i = \sqrt{r_i^2+s_i^2}, \tag{11}$$

or Duchon's thin plate splines

$$\varphi_i = r_i^{2m}\log(r_i), \quad m=1,2,3,\ldots, \tag{12}$$

where $r_i = ||\mathbf{x}-\mathbf{x}_i||$ is the Euclidian norm of vector $(\mathbf{x}-\mathbf{x}_i)$; $s_i$ is the RBF shape parameter which can be calculated by

$$s_i = \beta d_i^{min}, \tag{13}$$

where $\beta$ is the user-defined parameter and $d_i^{min}$ is the distance from the $i^{th}$ center to its nearest neighboring center [Moody and Darken (1989)].

Based on the closed form of the highest order derivative of $\phi_i(\mathbf{x})$ in (9), the lower derivatives and the function $\phi_i(\mathbf{x})$ itself are then successively obtained via symbolic integrations. The formulations of the first and second order antiderivatives of Hardy's multiquadrics and Duchon's thin plate splines are presented in Appendices 1 and 2, respectively. It should be noted that the integration process gives rise to constants of integration which contribute to the formulation of $\boldsymbol{\phi}(\mathbf{x})$ in (6). The detailed procedure for calculating these constants of integration is presented in Appendix 3.

It is noted that in the case of the DRBFN method, the RBF must be sufficiently differentiable to satisfy the particular differential operator. The MQ is, therefore, always applicable as it is $C^\infty$ continuous, whereas the TPS is only $C^{2m-1}$ continuous, and $m$ must be chosen appropriately to suit the differential operator. The IRBFN method, on the other hand, can use the first order TPS ($m=1$) regardless of the order of the differential operator because the method starts from the highest derivative and the approximation of the corresponding function is obtained via successive integrations. Another note to be mentioned here is that, although the MQ and TPS are used with up to second order derivatives of a function in this paper, the IRBFN method can be used without changing its interface (6)-(8) with other kinds of RBFs and/or with higher order derivatives provided that symbolic integrations can be performed.

## 3 Numerical schemes based on the IRBFN method for transient problems

Consider a general initial boundary value problem

$$\begin{aligned}\frac{\partial u}{\partial t}+Lu = {} & f \quad \text{in } Q_T := (0,T)\times\Omega,\\ Bu = {} & g_1 \quad \text{on } \Sigma_T := (0,T)\times\partial\Omega,\\ u = {} & g_2 \quad \text{on } \Omega,\ t=0,\end{aligned} \tag{14}$$

where $\Omega$ is a bounded domain in $\mathbf{R}^d$, $d=1,2,3$, with boundary $\partial\Omega$; $T>0$ is a prescribed time-level; $f=f(\mathbf{x})$, $g_1=g_1(\mathbf{x})$ and $g_2=g_2(\mathbf{x})$ are given functions; $L$ is a linear differential operator; $B$ is an operator representing Dirichlet, Neumann or Robin boundary conditions as follows

*Dirichlet*:

$$Bu = u \tag{15}$$

*Neumann*:

$$\begin{aligned}Bu &= \mathbf{n}\cdot\nabla u\\ &= n_1\frac{\partial u}{\partial x_1}+n_2\frac{\partial u}{\partial x_2}+n_3\frac{\partial u}{\partial x_3}\end{aligned} \tag{16}$$

where $\mathbf{n}=[n_1,n_2,n_3]^T$ is the outward unit vector normal to the boundary.

*Robin*:

$$Bu = \mathbf{n}\cdot\nabla u+\kappa u \tag{17}$$

where $\kappa$ is a given function on the boundary.

### 3.1 Fully discrete schemes

In fully discrete schemes, problem (14) is discretized with respect to both time and space variables. Firstly, the time interval $[0,T]$ is partitioned into $N_T$ subintervals $[t_n, t_{n+1}]$ of length $\Delta t = T/N_T$ with $t_0 = 0$ and $t_{N_T} = T$. The discretization of the problem in time is then accomplished by a time-stepping scheme, followed by the spatial discretization based on the IRBFN method. Among many possible time-stepping schemes, the standard $\theta$-scheme, $0 \leq \theta \leq 1$ is used in this work. It should be noted that the extreme cases $\theta = 0$ and $\theta = 1$ correspond to the well-known forward (fully explicit) and backward (fully implicit) Euler methods, respectively. The scheme associated with the case $\theta = 1/2$ is equivalent to the (semi-implicit) Crank-Nicolson method which is second-order accurate. Applying the $\theta$-scheme to problem (14) gives

$$\frac{\tilde{u}^{n+1}-\tilde{u}^{n}}{\Delta t}+\theta L\tilde{u}^{n+1}+(1-\theta)L\tilde{u}^{n} = f, \quad (18)$$

$$B\tilde{u}^{n+1} = g_1, \quad (19)$$

where $t^{n+1} = t^n + \Delta t$, $\tilde{u}^{n+1} \approx u(\mathbf{x}, t^{n+1})$, and $\tilde{u}^0 = g_2(\mathbf{x})$. The time discrete system (18)-(19) is then discretized in space by using the IRBFN approximants in (6)-(8) at time $t^{n+1}$ and applied at every data point to obtain a system of equations to be solved for $\boldsymbol{U}^{n+1}$. Note that data points are partitioned into $M_I$ interior points and $M - M_I$ boundary points. The fully discrete version of (18) reads

$$\boldsymbol{\phi}^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}\boldsymbol{U}^{n+1} + \theta\Delta t\boldsymbol{\phi}_L^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}\boldsymbol{U}^{n+1} =$$
$$\boldsymbol{\phi}^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}\boldsymbol{U}^{n} - (1-\theta)\Delta t\boldsymbol{\phi}_L^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}\boldsymbol{U}^{n} + \Delta t\mathbf{f}(\mathbf{x}_j),$$
$$j = 1,\ldots,M_I,$$

or

$$\left(\boldsymbol{\psi}^T(\mathbf{x}_j) + \theta\Delta t\boldsymbol{\psi}_L^T(\mathbf{x}_j)\right)\boldsymbol{U}^{n+1} = \boldsymbol{\psi}^T(\mathbf{x}_j)\boldsymbol{U}^{n} -$$
$$(1-\theta)\Delta t\boldsymbol{\psi}_L^T(\mathbf{x}_j)\boldsymbol{U}^{n} + \Delta t\mathbf{f}(\mathbf{x}_j), \quad j = 1,\ldots,M_I, \quad (20)$$

where $\boldsymbol{U}^n = [U_1(t^n),\ldots,U_N(t^n)]^T$ is a known set of nodal values of the function at time $t^n$, and

$$\begin{aligned}
\boldsymbol{\phi}_L(\mathbf{x}) &= [L\phi_1(\mathbf{x}),\ldots,L\phi_N(\mathbf{x})]^T, \\
\boldsymbol{\psi}^T(\mathbf{x}_j) &= \boldsymbol{\phi}^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}, \quad j = 1,\ldots,M_I, \\
\boldsymbol{\psi}_L^T(\mathbf{x}_j) &= \boldsymbol{\phi}_L^T(\mathbf{x}_j)\boldsymbol{\Phi}^{-1}, \quad j = 1,\ldots,M_I.
\end{aligned}$$

Similarly, one has fully discrete versions of (19) corresponding to different boundary conditions defined in (15), (16), and (17) as follows

*Dirichlet*:

$$\boldsymbol{\psi}^T(\mathbf{x}_j)\boldsymbol{U}^{n+1} = \mathbf{g}_1(\mathbf{x}_j),$$
$$j = M_I+1,\ldots,M, \quad (21)$$

*Neumann*:

$$\left(n_1\boldsymbol{\psi}_{,1}^T(\mathbf{x}_j) + n_2\boldsymbol{\psi}_{,2}^T(\mathbf{x}_j) + n_3\boldsymbol{\psi}_{,3}^T(\mathbf{x}_j)\right)\boldsymbol{U}^{n+1}$$
$$= \mathbf{g}_1(\mathbf{x}_j), \quad (22)$$

where

$$\boldsymbol{\psi}_{,i}^T(\mathbf{x}_j) = \boldsymbol{\phi}_{,i}^T(\mathbf{x})\boldsymbol{\Phi}^{-1},$$

in which $i = 1,2,3$ and $j = M_I+1,\ldots,M,$.

*Robin*:

$$\left(n_1\boldsymbol{\psi}_{,1}^T(\mathbf{x}_j) + n_2\boldsymbol{\psi}_{,2}^T(\mathbf{x}_j) + n_3\boldsymbol{\psi}_{,3}^T(\mathbf{x}_j) + \kappa\boldsymbol{\psi}^T(\mathbf{x}_j)\right)\boldsymbol{U}^{n+1}$$
$$= \mathbf{g}_1(\mathbf{x}_j), \quad (23)$$

where $j = M_I+1,\ldots,M$.

A general form corresponding to any of the above boundary conditions can be written as follows

$$\boldsymbol{\psi}_B^T(\mathbf{x}_j)\boldsymbol{U}^{n+1} = \mathbf{g}_1(\mathbf{x}_j), \quad j = M_I+1,\ldots,M, \quad (24)$$

where $\boldsymbol{\psi}_B^T(\mathbf{x})$ is defined by either (21), (22) or (23).

System (20) and (24) can be written in a compact form as follows

$$\mathbf{A}\boldsymbol{U}^{n+1} = \mathbf{b}, \quad (25)$$

where

$$\mathbf{A} = \left[\begin{array}{cc} \mathbf{I}+\theta\Delta t\boldsymbol{\Psi}_L & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_B \end{array}\right],$$

and

$$\mathbf{b} = \left[\begin{array}{c} \boldsymbol{U}^n - (1-\theta)\Delta t\boldsymbol{\Psi}_L\boldsymbol{U}^n + \Delta t\mathbf{f} \\ \mathbf{g}_1 \end{array}\right],$$

in which $\boldsymbol{\Psi}_L$ and $\boldsymbol{\Psi}_B$ result from the application of $\boldsymbol{\psi}_L(\mathbf{x})$ and $\boldsymbol{\psi}_B(\mathbf{x})$ at interior and boundary points, respectively.

The system (25) is solved at each time step for $\boldsymbol{U}^{n+1}$ until the prescribed time $T$ is reached using the corresponding nodal values at the previous time $t = t^n$, $\boldsymbol{U}^n$, and the IRBFN coefficient matrices associated with function $u$, differential operator $L$ and boundary operator $B$. It is noted that an alternative system of equations in terms of the weight coefficients $\mathbf{w}$ can be set up by minimizing the sum-squared error (SSE) in the sense of the general least-squares principle [Mai-Duy and Tran-Cong (2001a)]. However, the SSE approach is not used here since a slight drawback might arise due to the fact that boundary conditions are not strictly satisfied within the least-squares context. As shown above, the fully discrete schemes have been derived and the system (25) has been set up without using the SSE approach.

### 3.2 Semi-discrete schemes

In semi-discrete schemes, problem (14) is first discretized in space while still continuously dependent on time. The IRBFN approximation of function $u(\mathbf{x},t)$ is written here as a linear combination of shape functions $\psi_i(\mathbf{x})$ and nodal values $U_i(t)$ of function $u(\mathbf{x},t)$ from (6)

$$\tilde{u}(\mathbf{x},t) = \boldsymbol{\psi}^T(\mathbf{x})\boldsymbol{U}(t) = \sum_{i=1}^{N} U_i(t)\psi_i(\mathbf{x}). \qquad (26)$$

Substituting (26) into (14) yields a discrete version of the problem in space

$$\begin{aligned}\sum_{i=1}^{N}\frac{\mathrm{d}U_i(t)}{\mathrm{d}t}\psi_i(\mathbf{x}) + \sum_{i=1}^{N}U_i(t)\,\psi_{L_i}(\mathbf{x}) &= f(\mathbf{x}), &(27)\\ \sum_{i=1}^{N}U_i(t)\,\psi_{B_i}(\mathbf{x}) &= g_1(\mathbf{x}). &(28)\end{aligned}$$

The collocation method is then used to set up a system of ordinary differential equations (ODEs) by applying (27)-(28) at every data point

$$\sum_{i=1}^{N}\frac{\mathrm{d}U_i(t)}{\mathrm{d}t}\psi_i(\mathbf{x}_j) + \sum_{i=1}^{N}U_i(t)\,\psi_{L_i}(\mathbf{x}_j) = f(\mathbf{x}_j),$$
$$j = 1,...,M_I,$$
$$\sum_{i=1}^{N}U_i(t)\,\psi_{B_i}(\mathbf{x}_j) = g_1(\mathbf{x}_j),$$
$$j = M_I+1,...,M,$$

or in a compact form

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} + \boldsymbol{\Psi}_L\boldsymbol{U} &= \mathbf{f}, &(29)\\ \boldsymbol{\Psi}_B\boldsymbol{U} &= \mathbf{g}_1, &(30)\end{aligned}$$

where $\boldsymbol{\Psi}_L$ and $\boldsymbol{\Psi}_B$ result from the application of $\boldsymbol{\psi}_L(\mathbf{x})$ and $\boldsymbol{\psi}_B(\mathbf{x})$ at every data point, respectively.

Solving the ODE system (29)-(30) for $\boldsymbol{U}$ with the initial conditions $\boldsymbol{U}^0 = \mathbf{g}_2$ yields the solution to the problem (14) at every data point within the time interval of interest. It is noted that various high-order ODE solvers can be applied to solve the system (29)-(30) depending on the problem at hand. For illustrative purposes, the fourth-order Runge-Kutta method is described here in the IRBFN framework to solve the problem (14). In that case, the ODE (29) can be rewritten as

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{F}(t,\boldsymbol{U}), \qquad (31)$$

where $\boldsymbol{F} = \mathbf{f} - \boldsymbol{\Psi}_L\boldsymbol{U}$. At each time step, four function evaluations of $\boldsymbol{F}$ are performed taking into account the boundary conditions described in (30). The updated value of $\boldsymbol{U}$ at time $t^{n+1}$ is calculated as follows

$$\begin{aligned}\boldsymbol{K}_1 &= \Delta t\boldsymbol{F}\left(t^n,\boldsymbol{U}^n\right),\\ \boldsymbol{K}_2 &= \Delta t\boldsymbol{F}\left(t^{n+1/2},\boldsymbol{U}^n+\boldsymbol{K}_1/2\right),\\ \boldsymbol{K}_3 &= \Delta t\boldsymbol{F}\left(t^{n+1/2},\boldsymbol{U}^n+\boldsymbol{K}_2/2\right),\\ \boldsymbol{K}_4 &= \Delta t\boldsymbol{F}\left(t^{n+1},\boldsymbol{U}^n+\boldsymbol{K}_3\right),\\ \boldsymbol{U}^{n+1} &= \boldsymbol{U}^n + \frac{\boldsymbol{K}_1+2\boldsymbol{K}_2+2\boldsymbol{K}_3+\boldsymbol{K}_4}{6}, &(32)\end{aligned}$$

where $\Delta t$ is the time step size; $t^{n+1/2} = t^n + \Delta t/2$; $t^{n+1} = t^n + \Delta t$; $\boldsymbol{U}^n$ is the set of nodal values of function $u(\mathbf{x},t)$ at time $t^n$. It is noted that other methods such as Adams methods, predictor-corrector methods, and backward differentiation formula (BDF) can be used for both non-stiff and stiff problems [Quarteroni, Sacco, and Saleri (2000)].

## 4 Numerical Examples

For verification purposes, the IRBFN method is applied to solve five example problems namely the 1D and 2D diffusion equations, the 1D and 2D wave equations, and the 1D advection-diffusion equation. The results are then compared in terms of accuracy and efficiency to those from other methods such as finite difference, finite element, boundary element and the DRBFN methods. Furthermore, three network parameters including the user-defined parameter β described in (13), the point density and the time-step size are taken into account to investigate their influence on the accuracy of the solutions by the MQ-IRBFN method.

In order to compare the present results with published works, it is necessary to use the same data format and hence most of the accuracy comparisons in this paper are made necessarily in a pointwise manner using absolute/relative errors at each time step. For each table of comparison, however, $L_\infty$-norm of error is shown for further information. In addition, root-mean-squared-error throughout the time domain are used for sensitivity analysis. The root-mean-squared error is calculated by

$$RMSE = \sqrt{\frac{1}{N_T}\sum_{i=1}^{N_T}(u_i - \tilde{u}_i)^2}, \qquad (33)$$

where $\tilde{u}$ and $u$ are numerical and analytical solutions, respectively; $N_T$ is the total number of time steps.

### 4.1 Example 1. One-dimensional heat equation

Two test problems are considered in this example where the numerical solutions by the present IRBFN method are compared to those by the FDM and DRBFN method.

**Example 1.1**

Consider an 1D heat equation

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < 1, \quad t > 0,$$

subject to boundary and initial conditions

$$\begin{aligned} u(0,t) &= u(1,t) = 0, \quad t > 0, \\ u(x,0) &= \sin \pi x, \quad 0 \le x \le 1, \end{aligned}$$

where $u(x,t)$ is the temperature at the position $x$ at time $t$. The problem has the analytical solution $u(x,t) = \sin \pi x \cos 2\pi t$.

The finite difference method (FDM) and the IRBFN method are used to solve the problem. For the FDM, Crank-Nicolson algorithm is used whereas the fully discrete scheme described in section (3.1) with θ=0.5 is used for the IRBFN method. The numerical solution by the IRBFN method is shown in Figure 1. Table 1 shows that the IRBFN method using either MQ or TPS RBFs gives more accurate result than the FDM with the same number of points $N = 11$ and the time step size $\Delta t = 0.01$. In particular, the present method yields the solution accurate to 4 significant digits with TPS and up to 5 significant digits with MQ. In Table 2, the IRBFN method shows its superiority over the FDM when the point density and the number of time steps it uses are only half as many as the FDM uses. With such a "crude" discretization in both time and space domain, the IRBFN method still yields solutions more accurate than that of the FDM.

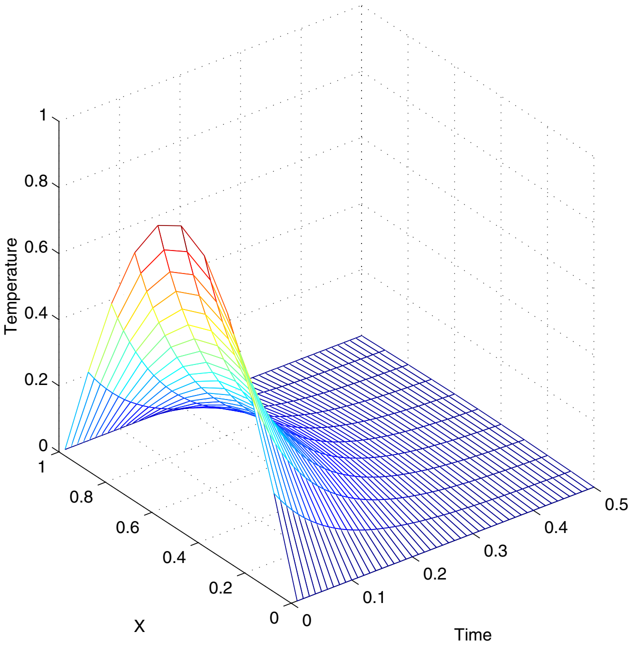

**Figure 1** : Solution profile in Example 1.1 by the MQ-IRBFN method.

For the purpose of sensitivity studies for the MQ-IRBFN method, four point densities are chosen to be 5, 7, 9, 11 points, and the numbers of time steps of interest are 25 and 50. In addition, a wide range of β's values from 1 to 10 are used to study its influence on the solution accuracy. Figures 2a, 2b, 2c, 2d show the maximum absolute errors by the MQ-IRBFN method over the entire set of collocation points, and figures 2e, 2f, 2g, 2h present the root-mean-squared errors over the time domain of 25 time steps. It can be seen from the figures that the errors of the proposed method do not vary significantly for this problem where β's are in the range from 1 to 10. In particular, these errors are of

**Table 1** : Accuracy comparison of the solution at time $t = 0.5$ between the FDM and the IRBFN method (using either MQ or TPS) in Example 1.1. Both methods use the same discretization ($N = 11, \Delta t = 0.01$). The $L_\infty$-norm of the error vector for FDM, TPS-IRBFN and MQ-IRBFN are $2.6766e-4$, $1.6887e-5$ and $8.3494e-6$, respectively.

| | | Numerical solution | | | Absolute Error | | |
|---|---|---|---|---|---|---|---|
| X | Exact | FDM | TPS | MQ | FDM | TPS | MQ |
| 0.1 | 0.002222 | 0.002305 | 0.002219 | 0.002224 | 8.271e-05 | 3.341e-06 | 1.908e-06 |
| 0.2 | 0.004227 | 0.004385 | 0.004218 | 0.004226 | 1.573e-04 | 8.856e-06 | 9.966e-07 |
| 0.3 | 0.005818 | 0.006035 | 0.005805 | 0.005815 | 2.165e-04 | 1.311e-05 | 3.066e-06 |
| 0.4 | 0.006840 | 0.007094 | 0.006824 | 0.006835 | 2.546e-04 | 1.591e-05 | 4.544e-06 |
| 0.5 | 0.007192 | 0.007460 | 0.007175 | 0.007187 | 2.677e-04 | 1.689e-05 | 5.035e-06 |
| 0.6 | 0.006840 | 0.007094 | 0.006824 | 0.006835 | 2.546e-04 | 1.592e-05 | 4.543e-06 |
| 0.7 | 0.005818 | 0.006035 | 0.005805 | 0.005815 | 2.165e-04 | 1.311e-05 | 3.064e-06 |
| 0.8 | 0.004227 | 0.004385 | 0.004218 | 0.004226 | 1.573e-04 | 8.861e-06 | 9.937e-07 |
| 0.9 | 0.002222 | 0.002305 | 0.002219 | 0.002224 | 8.271e-05 | 3.348e-06 | 1.912e-06 |

**Table 2** : Accuracy comparison of the solution at time $t = 0.5$ between the FDM and the IRBFN method (using either MQ or TPS) in Example 1.1. Although the number of collocation points and time steps the IRBFN method needs (6 and 25) are only half as many as the FDM uses (11 and 50), the IRBFN method still yields more accurate solution. The $L_\infty$-norm of the error vector for FDM, TPS-IRBFN and MQ-IRBFN are $2.5455e-4$, $4.3845e-5$ and $1.8211e-5$, respectively.

| | | Numerical solution | | | Absolute Error | | |
|---|---|---|---|---|---|---|---|
| X | Exact | FDM | TPS | MQ | FDM | TPS | MQ |
| 0.1 | 0.002222 | 0.002305 | - | - | 8.271e-05 | - | - |
| 0.2 | 0.004227 | 0.004385 | 0.004271 | 0.004245 | 1.573e-04 | 4.362e-05 | 1.821e-05 |
| 0.3 | 0.005818 | 0.006035 | - | - | 2.165e-04 | - | - |
| 0.4 | 0.006840 | 0.007094 | 0.006884 | 0.006852 | 2.546e-04 | 4.384e-05 | 1.176e-05 |
| 0.5 | 0.007192 | 0.007460 | - | - | 2.677e-04 | - | - |
| 0.6 | 0.006840 | 0.007094 | 0.006884 | 0.006852 | 2.546e-04 | 4.384e-05 | 1.176e-05 |
| 0.7 | 0.005818 | 0.006035 | - | - | 2.165e-04 | - | - |
| 0.8 | 0.004227 | 0.004385 | 0.004271 | 0.004245 | 1.573e-04 | 4.361e-05 | 1.821e-05 |
| 0.9 | 0.002222 | 0.002305 | - | - | 8.271e-05 | - | - |

the same order of magnitude. In addition, a "mesh" (or discretization) convergence can be observed from figures 2b, 2d and 2f, 2h where the accuracy of the numerical solution does not improve significantly when the number of points $nx$ increases from 7 to 11.

**Example 1.2**

Consider the one-dimensional heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1,\, t > 0,$$

subject to Dirichlet boundary conditions

$$u(0,t) = u(1,t) = 0, \quad t \geq 0,$$

and initial conditions

$$u(x,0) = \begin{cases} \mu x, & 0 \leq x \leq 0.5, \\ \mu(1-x), & 0.5 \leq x \leq 1. \end{cases}$$

The analytical solution to the problem is given by

$$u(x,t) = \frac{4\mu}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \sin\frac{n\pi x}{2} e^{-\kappa t n^2 \pi^2}.$$

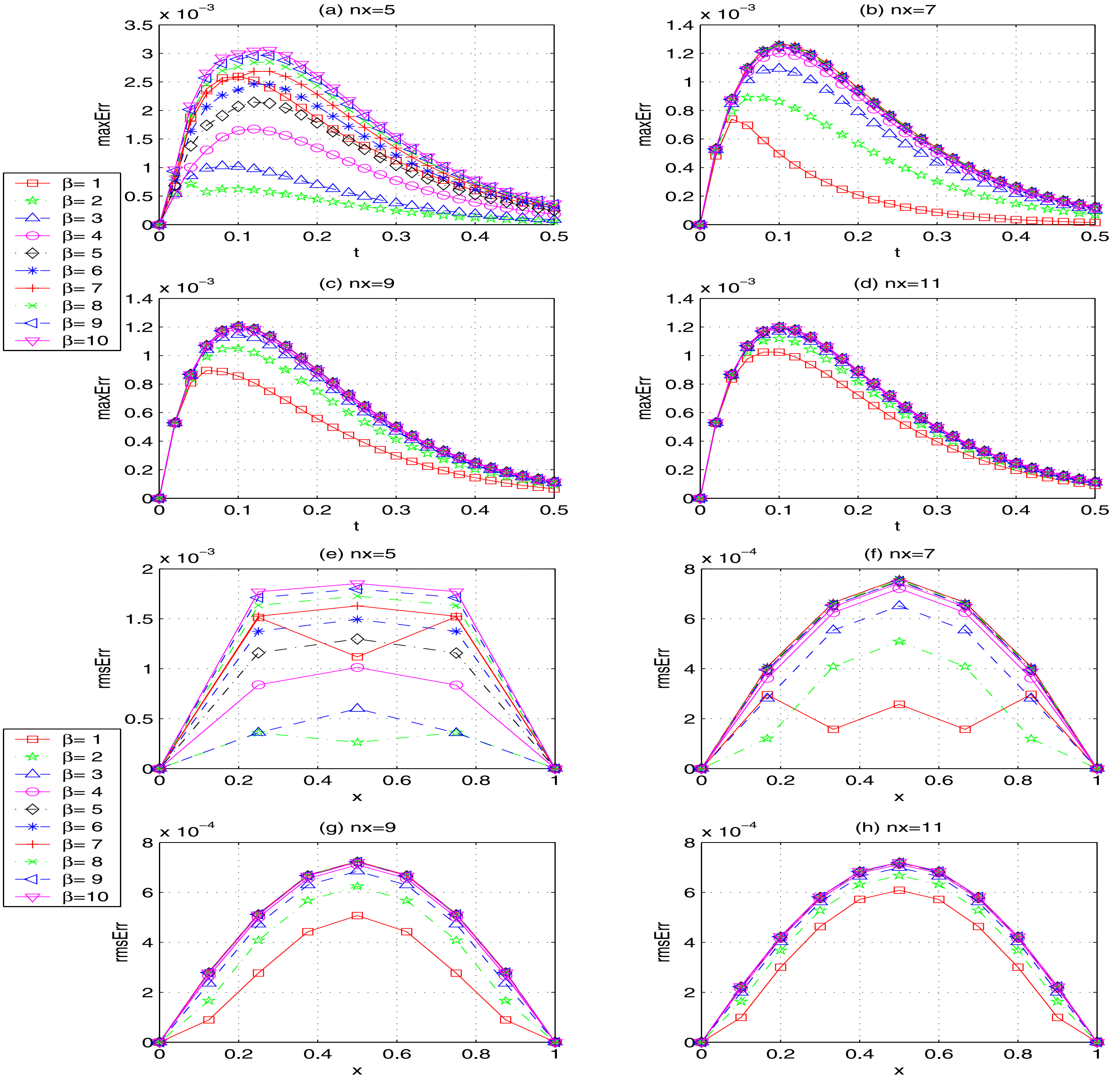

**Figure 2** : Sensitivity analysis of the MQ-IRBFN method with respect to the network parameters in Example 1.1 for the case where the number of time steps = 25. Figures a,b,c,d show the maximum absolute error over the entire set of collocation points, and figures e,f,g,h show the root-mean-squared errors over the whole time domain.

In this example, the problem is solved by both direct and indirect RBFN methods. The solution by the DRBFN method is obtained from [Zerroukat, Power, and Chen (1998)] where the explicit scheme, namely EEX-MQ, was used. The scheme used explicit exponential formulation with multiquadrics as RBFs and was reported to yield the most accurate results among others. For the IRBFN method, a semi-discrete scheme based on Adams predictor-corrector method [Quarteroni, Sacco, and Saleri (2000)] is used with either MQ or TPS to solve the problem. Figure 3 shows the solution profile of the problem given by the MQ-IRBFN method. Solutions at time

**Table 3** : Example 1.2: Accuracy comparison between the DRBFN using explicit exponential scheme with MQ [Zerroukat, Power, and Chen (1998)] and the IRBFN using semi-discrete scheme with MQ. With the same number of collocation points, the indirect method yields more accurate solution while the number of time steps it uses ($Nts = 10$) is one-tenth of that required by the direct method ($Nts = 100$). The $L_\infty$-norm of the error vector for MQ-Direct and MQ-Indirect are $5.6000e-4$ and $5.2220e-5$, respectively.

| | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|
| X | Exact | MQ-Direct | MQ-Indirect | MQ-Direct | MQ-Indirect |
| 0.10 | 0.00483 | 0.00458 | 0.00484 | 2.5000e-004 | 1.0235e-005 |
| 0.30 | 0.01265 | 0.01218 | 0.01262 | 4.7000e-004 | 2.5465e-005 |
| 0.50 | 0.01564 | 0.01508 | 0.01560 | 5.6000e-004 | 4.2446e-005 |
| 0.60 | 0.01488 | 0.01434 | 0.01483 | 5.4000e-004 | 5.2220e-005 |
| 0.80 | 0.00919 | 0.00882 | 0.00918 | 3.7000e-004 | 1.2937e-005 |

**Table 4** : Example 1.2: Accuracy comparison between the DRBFN using explicit exponential scheme with MQ [Zerroukat, Power, and Chen (1998)] and the IRBFN using semi-discrete scheme with TPS. Although the number of time steps it uses ($Nts = 10$) is one-tenth of that required by the direct method ($Nts = 100$), still the indirect method yields more accurate solution. The two methods use the same number of collocation points. The $L_\infty$-norm of the error vector for MQ-Direct and TPS-Indirect are $5.6000e-4$ and $1.6953e-4$, respectively.

| | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|
| X | Exact | MQ-Direct | TPS-Indirect | MQ-Direct | TPS-Indirect |
| 0.10 | 0.00483 | 0.00458 | 0.00489 | 2.5000e-004 | 6.0989e-005 |
| 0.30 | 0.01265 | 0.01218 | 0.01275 | 4.7000e-004 | 1.0490e-004 |
| 0.50 | 0.01564 | 0.01508 | 0.01576 | 5.6000e-004 | 1.1617e-004 |
| 0.60 | 0.01488 | 0.01434 | 0.01505 | 5.4000e-004 | 1.6953e-004 |
| 0.80 | 0.00919 | 0.00882 | 0.00929 | 3.7000e-004 | 1.0127e-004 |

$t = 1$ by direct and indirect methods are shown in Tables 3 and 4. As can be seen from the tables, with the same number of collocation points, the IRBFN method using either MQ or TPS outperforms the DRBFN method using MQ while the number of time steps it uses ($Nts = 10$) is only one-tenth of that required by the direct method with MQ ($Nts = 100$). It is also noted that the solutions by the IRBFN near the boundary points are of high accuracy without any special treatment.

### *4.2 Example 2. Two-dimensional diffusion equation*

Consider the 2D diffusion equation

$$\frac{\partial u}{\partial t} = \nabla^2 u + f(x,y,t),$$

to be solved in the domain $0 \le x \le 1,\ 0 \le y \le 1$. The forcing function is given by

$$f(x,y,t) = \sin x \sin y (2\sin t + \cos t).$$

The initial and boundary conditions are appropriate to the analytical solution

$$u = \sin x \sin y \sin t.$$

In this example, the problem is solved by the BEM [Ingber and Phan-Thien (1992)] and the present IRBFN method. For the BEM, Ingber and Phan-Thien used two numerical schemes, namely $M1$ and $M2$, where "generalized" forcing functions can be approximated by radial basis functions. For the IRBFN method, the fully discrete scheme presented in section 3.1 is used with $\theta = 0.5$. For the purpose of comparing the two methods, the solution to the problem at point (0.8,0.8) throughout the whole time domain is considered. The solution by the BEM is obtained by scheme $M2$ (which was reported to be slightly more accurate than scheme $M1$) with a fine mesh of $17 \times 17$ boundary points and $9 \times 9$ interior points. As the IRBFN method uses a set of randomly generated

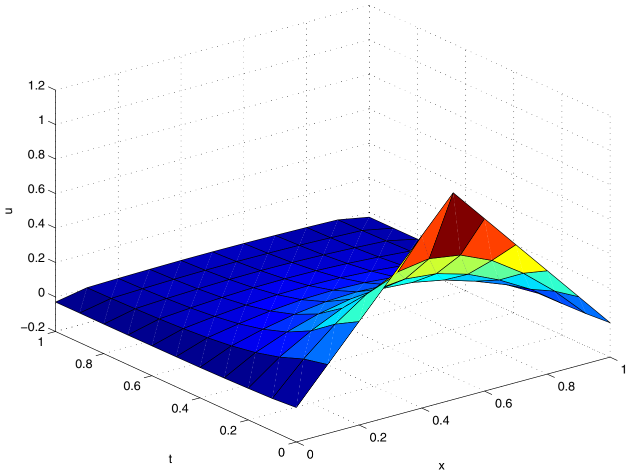

**Figure 3** : Solution profile of the problem given by the MQ-IRBFN method in Example 1.2.

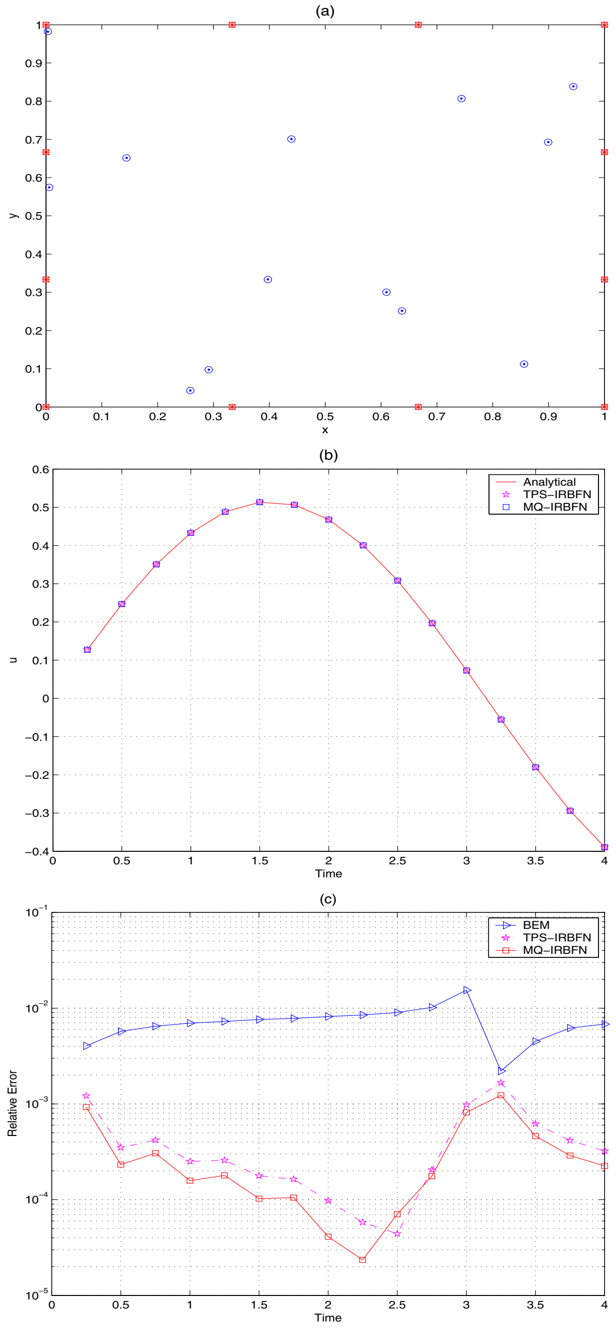

**Figure 4** : (a) A set of randomly generated points used by the IRBFN method; (b) Solution at point (0.8,0.8) by the IRBFN method; (c) An accuracy comparison between the IRBFN method and the BEM [Ingber and Phan-Thien (1992)] in Example 2. Using a coarser point density, the IRBFN method maintains its higher accuracy compared to the BEM throughout the whole time domain.

points, the solution at the point of interest can be post-processed by simple function evaluation based on (6). Figure 4.a presents the set of randomly generated collocation points, and Figure 4.b shows the solution at point (0.8,0.8) by the IRBFN method. The results in this example show that the present IRBFN method yields accurate solution not only at grid points but also at arbitrary points in the domain. Table 5 shows that the IRBFN method using either MQ or TPS outperforms the BEM in terms of accuracy and efficiency. As can be seen from the table, the IRBFN method yields more accurate result even with a coarser discretization (12 boundary points and 13 interior points) than the BEM does with a much finer mesh (289 boundary points and 81 interior points). Furthermore, Figure 4.b shows that the IRBFN method maintains its higher accuracy compared to the BEM throughout the whole time domain. In this example, both methods use the same time step size $\Delta t$=0.25.

For the purpose of studying the sensitivity of the MQ-IRBFN method to the network parameters, four regular point densities $(5 \times 5)$, $(7 \times 7)$, $(9 \times 9)$ and $(11 \times 11)$ are used together with a range of values of the user-defined parameter $\beta$ from 1 to 10 and two different numbers of time steps (16 and 32). Figure 5 shows that the maximum absolute errors by the IRBFN method do not change significantly within a range of $\beta$'s values from 1 to 10. Furthermore, it can be seen from the figures that while "mesh" convergence can be observed at a very coarse point density, the accuracy of the solution can

**Table 5** : Solution to the problem in Example 2 for the interior point $x = 0.8$, $y = 0.8$. With a coarser point density (12 boundary points and 13 interior points), still the IRBFN method (based on either MQ or TPS) yields more accurate result compared to the BEM [Ingber and Phan-Thien (1992)] using a much finer mesh (289 boundary points and 81 interior points). Both methods use the same time step size $\Delta t$=0.25. The $L_\infty$-norm of the error vector for BEM, TPS-IRBFN and MQ-IRBFN are $3.9589e-3$, $1.5474e-4$ and $1.1788e-4$, respectively.

| | | Numerical solution | | | Relative Error | | |
|---|---|---|---|---|---|---|---|
| T | Exact | BEM | TPS | MQ | BEM | TPS | MQ |
| 0.3 | 0.127314 | 0.126800 | 0.127159 | 0.127196 | 4.037e-03 | 1.215e-03 | 9.259e-04 |
| 0.5 | 0.246712 | 0.245300 | 0.246626 | 0.246655 | 5.724e-03 | 3.497e-04 | 2.326e-04 |
| 0.8 | 0.350771 | 0.348500 | 0.350624 | 0.350664 | 6.475e-03 | 4.187e-04 | 3.059e-04 |
| 1.0 | 0.433021 | 0.430000 | 0.432912 | 0.432952 | 6.976e-03 | 2.512e-04 | 1.583e-04 |
| 1.3 | 0.488347 | 0.484800 | 0.488221 | 0.488260 | 7.264e-03 | 2.584e-04 | 1.796e-04 |
| 1.5 | 0.513311 | 0.509400 | 0.513220 | 0.513258 | 7.619e-03 | 1.776e-04 | 1.027e-04 |
| 1.8 | 0.506359 | 0.502400 | 0.506276 | 0.506306 | 7.818e-03 | 1.639e-04 | 1.053e-04 |
| 2.0 | 0.467924 | 0.464100 | 0.467878 | 0.467905 | 8.173e-03 | 9.778e-05 | 4.108e-05 |
| 2.3 | 0.400396 | 0.397000 | 0.400373 | 0.400387 | 8.482e-03 | 5.801e-05 | 2.351e-05 |
| 2.5 | 0.307974 | 0.305200 | 0.307987 | 0.307995 | 9.006e-03 | 4.404e-05 | 7.055e-05 |
| 2.8 | 0.196403 | 0.194400 | 0.196443 | 0.196437 | 1.020e-02 | 2.049e-04 | 1.761e-04 |
| 3.0 | 0.072620 | 0.071500 | 0.072691 | 0.072679 | 1.543e-02 | 9.775e-04 | 8.141e-04 |
| 3.3 | -0.055677 | -0.055800 | -0.055584 | -0.055608 | 2.206e-03 | 1.667e-03 | 1.234e-03 |
| 3.5 | -0.180513 | -0.179700 | -0.180401 | -0.180430 | 4.504e-03 | 6.193e-04 | 4.609e-04 |
| 3.8 | -0.294125 | -0.292300 | -0.294003 | -0.294040 | 6.206e-03 | 4.152e-04 | 2.891e-04 |
| 4.0 | -0.389450 | -0.386800 | -0.389325 | -0.389363 | 6.805e-03 | 3.226e-04 | 2.251e-04 |

be further improved by increasing the number of time steps. In this particular example, as the number of collocation points increases from Figures 5(a)(c) to more than 2 times denser in Figures 5(b)(d), the maximum errors do not decrease noticeably, indicating "mesh" convergence. However, as the time step size decreases from Figures 5(a)(b) to half the value in Figures 5(c)(d), the errors decrease by an order of magnitude. It can also be seen from Figure 6 that root-mean-squared errors by the IRBFN method in Example 2 do not vary significantly within a range of β's values from 1 to 10 in figures (a)-(g) for the case where the number of time steps $Nts = 16$, and in figures (h)-(n) for $Nts = 32$.

### 4.3 Example 3. One-dimensional wave equation

Consider the 1D wave equation

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < 1, \quad t > 0,$$

subject to Dirichlet boundary conditions

$$u(0,t) = u(1,t) = 0, \quad t > 0,$$

and initial conditions

$$\begin{aligned} u(x,0) &= f(x), \quad 0 \le x \le 1, \\ \frac{\partial u}{\partial t}(x,0) &= g(x), \quad 0 \le x \le 1. \end{aligned}$$

With $f(x) = \sin \pi x$, $g(x) = 0$, $\alpha = 2$ the problem has the analytical solution

$$u(x,t) = \sin \pi x \cos 2\pi t.$$

The problem is solved by semi-discrete schemes described in section (3.2) using $N = 11$ collocation points in space. In order to apply the semi-discrete scheme, the problem is first reformulated as a system of first order ODEs

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) &= v(x,t), \\ \frac{\partial v}{\partial t}(x,t) &= \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t), \end{aligned}$$

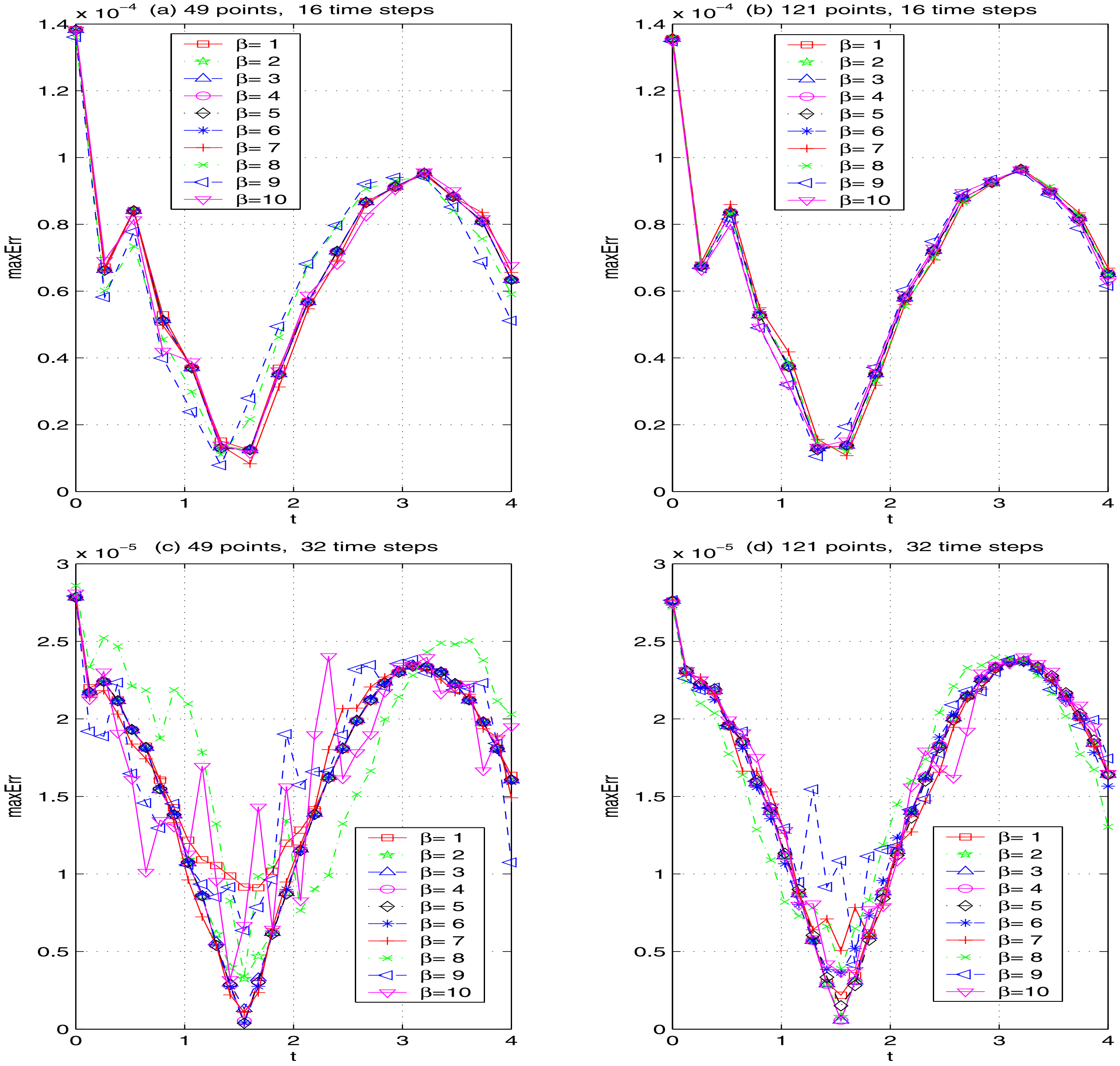

**Figure 5** : In Example 2, as the point density increases from figures (a)(c) to more than 2 times denser in figures (b)(d), the maximum errors by the MQ-IRBFN method do not decrease noticeably, indicating "mesh" convergence. The accuracy of the solution can be, however, improved by an order of magnitude by decreasing the time step size in figures (a)(b) to half the value in figures (c)(d). In addition, the errors by the MQ-IRBFN method do not to vary significantly within the range of β's values from 1 to 10.

with initial conditions

$$u(x,0) = f(x),$$
$$v(x,0) = g(x),$$

and subject to the same boundary conditions as in the original problem.

In this example, the one-step fourth-order Runge-Kutta (RK) with adaptive step control and the Adams predictor-corrector (PC) schemes [Quarteroni, Sacco, and Saleri (2000)] are applied to solve the problem in the IRBFN

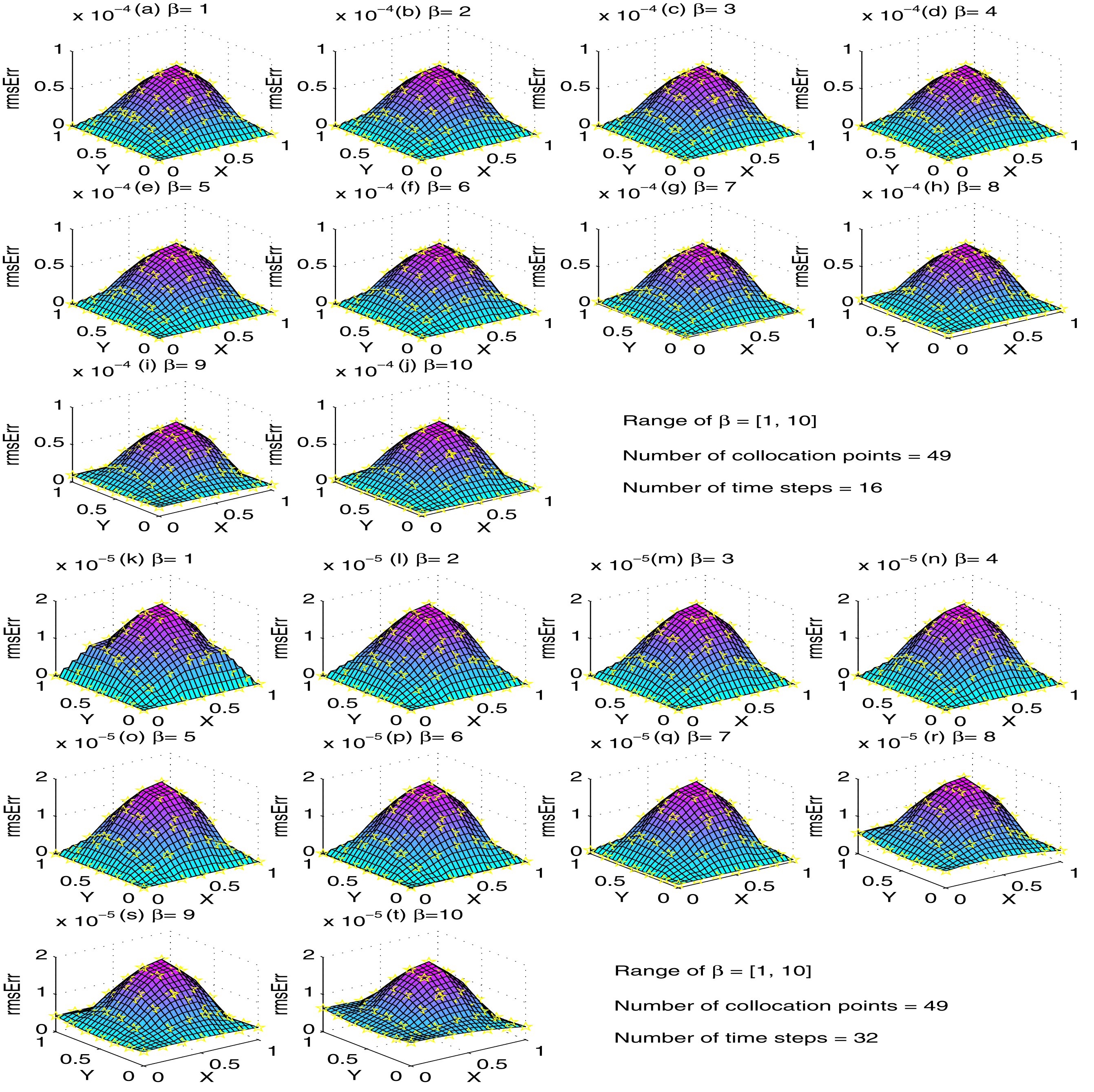

**Figure 6** : The root-mean-squared errors by the MQ-IRBFN method in Example 2 do not vary significantly within a range of β's values from 1 to 10 in figures (a)-(j) where the number of time steps $Nts = 16$, and in figures (k)-(t) for $Nts = 32$. An increase of the number of time steps from 16 to 32 results in a noticeable decrease in the root-mean-squared errors.

framework. The solution profile is shown in Figure 7. Tables 6 and 7 show the solution and absolute errors by the two semi-discrete schemes using MQ and TPS, respectively. There is no significant difference in the solutions by RK and PC schemes using the same type of RBF (MQ or TPS). The predictor-corrector scheme, however, proved to be more efficient than Runge-Kutta for non-stiff problems with stringent tolerances and when the

**Table 6** : Accuracy comparison in Example 3 between the two semi-discrete schemes, Runge-Kutta (RK) and Predictor-Corrector (PC), based on MQ-IRBFN. The numerical and analytical solutions are shown at time t=1.0. The $L_\infty$-norm of the error vector for MQ-IRBFN using RK and PC are $4.8788e-3$ and $4.9064e-3$, respectively.

| | | MQ-IRBFN | | Absolute Error | |
|---|---|---|---|---|---|
| T | Exact | RK | PC | RK | PC |
| 0.10 | 0.30901699 | 0.30901605 | 0.30899770 | 9.48658E-007 | 1.92943E-005 |
| 0.20 | 0.58778525 | 0.58776384 | 0.58779166 | 2.14102E-005 | 6.40824E-006 |
| 0.30 | 0.80901699 | 0.80902993 | 0.80902490 | 1.29361E-005 | 7.90858E-006 |
| 0.40 | 0.95105652 | 0.95106126 | 0.95104420 | 4.74844E-006 | 1.23145E-005 |
| 0.50 | 1.00000000 | 0.99997783 | 1.00001933 | 2.21743E-005 | 1.93331E-005 |
| 0.60 | 0.95105652 | 0.95106128 | 0.95104421 | 4.75879E-006 | 1.23049E-005 |
| 0.70 | 0.80901699 | 0.80902993 | 0.80902490 | 1.29347E-005 | 7.90928E-006 |
| 0.80 | 0.58778525 | 0.58776383 | 0.58779165 | 2.14231E-005 | 6.39342E-006 |
| 0.90 | 0.30901699 | 0.30901605 | 0.30899771 | 9.40765E-007 | 1.92876E-005 |

**Table 7** : Accuracy comparison in Example 3 between the two semi-discrete schemes, Runge-Kutta (RK) and Predictor-Corrector (PC), based on TPS-IRBFN. The numerical and analytical solutions are shown at time t=1.0. The $L_\infty$-norm of the error vector for TPS-IRBFN using RK and PC are $5.2349e-3$ and $5.2791e-3$, respectively.

| | | TPS-IRBFN | | Absolute Error | |
|---|---|---|---|---|---|
| T | Exact | RK | PC | RK | PC |
| 0.10 | 0.30901699 | 0.30905179 | 0.30906749 | 3.47971E-005 | 5.04963E-005 |
| 0.20 | 0.58778525 | 0.58768745 | 0.58769273 | 9.78054E-005 | 9.25245E-005 |
| 0.30 | 0.80901699 | 0.80906449 | 0.80903911 | 4.74931E-005 | 2.21147E-005 |
| 0.40 | 0.95105652 | 0.95106175 | 0.95111521 | 5.23780E-006 | 5.86968E-005 |
| 0.50 | 1.00000000 | 0.99997721 | 0.99992776 | 2.27937E-005 | 7.22411E-005 |
| 0.60 | 0.95105652 | 0.95106176 | 0.95111531 | 5.24051E-006 | 5.87906E-005 |
| 0.70 | 0.80901699 | 0.80906450 | 0.80903899 | 4.75097E-005 | 2.19910E-005 |
| 0.80 | 0.58778525 | 0.58768744 | 0.58769281 | 9.78157E-005 | 9.24443E-005 |
| 0.90 | 0.30901699 | 0.30905177 | 0.30906750 | 3.47800E-005 | 5.05023E-005 |

right-hand-side functions are particularly expensive to evaluate [Quarteroni, Sacco, and Saleri (2000)]. As can be seen from the tables, MQ-IRBFN with $\beta = 2$ yields more accurate solution than TPS-IRBFN in this example.

A range of the user-defined parameter $\beta$ is used together with four point densities (5,7,9,11) and two different numbers of time steps (25 and 50) to investigate their influence on the accuracy of the numerical solution. Figure 8 shows that the root-mean-squared errors by the IRBFN method do not change significantly within a range of $\beta$'s values from 1 to 3. Also, it can be seen from the figures that while "mesh" convergence can be observed at a very coarse point density, the accuracy of the solution can be further improved by increasing the number of time steps.

### *4.4 Example 4. Two-dimensional wave equation*

Consider the 2D wave equation

$$\frac{\partial^2 u}{\partial t^2}(\mathbf{x},t) - \alpha^2 \nabla^2 u(\mathbf{x},t) = 0, \quad \mathbf{x} \in \Omega \subset \mathbf{R}^2, \quad t > 0,$$

subject to Dirichlet boundary conditions

$$u(\mathbf{x},t) = 0, \mathbf{x} \in \partial\Omega, \quad t > 0,$$

and initial conditions

$$\begin{aligned} u(\mathbf{x},0) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ \frac{\partial u}{\partial t}(\mathbf{x},0) &= g(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{aligned}$$

**Table 8** : Comparison of accuracy and efficiency between the FEM and the MQ-IRBFN method in Example 4 at points (0.8,:) at time $t = 0.16$. The FEM uses 177 nodes and 312 triangles while the MQ-IRBFN method uses 121 collocation points. The $L_\infty$-norm of the error vector for FEM and MQ-IRBFN are $7.9915e-4$ and $2.3697e-4$, respectively.

| | | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|---|
| X | Y | Exact | FEM | MQ | FEM | MQ |
| 0.80 | 0.10 | 0.001874 | 0.002213 | 0.001720 | 3.3933e-004 | 1.5402e-004 |
| 0.80 | 0.20 | 0.004222 | 0.004796 | 0.004229 | 5.7389e-004 | 6.9258e-006 |
| 0.80 | 0.30 | 0.007521 | 0.007719 | 0.007583 | 1.9771e-004 | 6.2266e-005 |
| 0.80 | 0.40 | 0.010127 | 0.010329 | 0.010149 | 2.0203e-004 | 2.2110e-005 |
| 0.80 | 0.50 | 0.010995 | 0.011122 | 0.011021 | 1.2658e-004 | 2.5854e-005 |
| 0.80 | 0.60 | 0.010127 | 0.010218 | 0.010149 | 9.1194e-005 | 2.2559e-005 |
| 0.80 | 0.70 | 0.007521 | 0.007961 | 0.007580 | 4.4046e-004 | 5.9521e-005 |
| 0.80 | 0.80 | 0.004222 | 0.004797 | 0.004231 | 5.7448e-004 | 8.3148e-006 |
| 0.80 | 0.90 | 0.001874 | 0.002177 | 0.001728 | 3.0315e-004 | 1.4563e-004 |

**Table 9** : Comparison of accuracy and efficiency between the FEM and the TPS-IRBFN method in Example 4 at points (0.8,:) at the point of time $t = 0.16$. The FEM uses 177 nodes and 312 triangles while the TPS-IRBFN method uses 121 collocation points. The $L_\infty$-norm of the error vector for FEM and TPS-IRBFN are $7.9915e-4$ and $2.4562e-4$, respectively.

| | | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|---|
| X | Y | Exact | FEM | TPS | FEM | TPS |
| 0.80 | 0.10 | 0.001874 | 0.002213 | 0.001702 | 3.3933e-004 | 1.7164e-004 |
| 0.80 | 0.20 | 0.004222 | 0.004796 | 0.004245 | 5.7389e-004 | 2.2934e-005 |
| 0.80 | 0.30 | 0.007521 | 0.007719 | 0.007588 | 1.9771e-004 | 6.6888e-005 |
| 0.80 | 0.40 | 0.010127 | 0.010329 | 0.010176 | 2.0203e-004 | 4.8762e-005 |
| 0.80 | 0.50 | 0.010995 | 0.011122 | 0.011045 | 1.2658e-004 | 4.9373e-005 |
| 0.80 | 0.60 | 0.010127 | 0.010218 | 0.010169 | 9.1194e-005 | 4.1971e-005 |
| 0.80 | 0.70 | 0.007521 | 0.007961 | 0.007548 | 4.4046e-004 | 2.7258e-005 |
| 0.80 | 0.80 | 0.004222 | 0.004797 | 0.004247 | 5.7448e-004 | 2.5058e-005 |
| 0.80 | 0.90 | 0.001874 | 0.002177 | 0.001769 | 3.0315e-004 | 1.0419e-004 |

where $\Omega = \{(x,y)\,|\,0 \le x \le a, 0 \le y \le b\}$. With $f(x) = cx(a-x)y(b-y)$ and $g(x) = 0$, the problem has the analytical solution

$$u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \mathbf{B}_{mn} \sin\frac{m\pi x}{a} \cos\left(\alpha\pi t\sqrt{\frac{m^2}{a^2}+\frac{n^2}{b^2}}\right),$$

where

$$\mathbf{B}_{mn} = \frac{16a^2b^2c}{\pi^6 m^3 n^3}(1-\cos m\pi)(1-\cos n\pi).$$

The IRBFN method with a semi-discrete scheme based on fourth-order Runge-Kutta scheme is used to solve the problem, and the numerical solution at time step = 5 with time step size $\Delta t = 0.04$ is shown in Figure 9 where $\alpha^2 = 3$, $a = 1$, $b = 1$ and $c = 1$. It is noted that in order to apply the semi-discrete scheme, the problem is reformulated in to a system of first order ODEs as in Example 3. Comparisons of accuracy and efficiency between the FEM and the IRBFN method using both MQ and TPS are also performed in this example. With the FEM, a mesh of 177 nodes and 312 triangles is generated to solve the problem whereas for the IRBFN method, 121 collocation points are used. The accuracy comparison between the two methods is shown in Table 8 and 9 for the case of MQ

**Table 10** : Example 5: Comparison of accuracy and efficiency between the MQ-IRBFN method and the TPS-DRBFN method [Zerroukat, Djidjeli, and Charafi (2000)] at $t = 1$. With the same number of collocation points, the MQ-IRBFN method outperforms the TPS-DRBFN method while the number of time steps it uses ($Nts$=25) is half as many as that required by the TPS-DRBFN method. The $L_\infty$-norm of the error vector for TPS-DRBFN and MQ-IRBFN are $9.8935e-3$ and $5.9243e-5$, respectively.

| | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|
| X | Analytical | DRBFN | MQ-IRBFN | DRBFN | MQ-IRBFN |
| 0.2500 | 0.737486 | 0.747380 | 0.737494 | 9.89e-003 | 7.76e-006 |
| 0.3125 | 0.666554 | 0.675000 | 0.666578 | 8.45e-003 | 2.38e-005 |
| 0.3750 | 0.602444 | 0.608810 | 0.602454 | 6.37e-003 | 9.92e-006 |
| 0.5000 | 0.492129 | 0.493910 | 0.492116 | 1.78e-003 | 1.22e-005 |
| 0.6250 | 0.402014 | 0.400190 | 0.401988 | 1.82e-003 | 2.59e-005 |
| 0.7500 | 0.328400 | 0.324740 | 0.328358 | 3.66e-003 | 4.19e-005 |
| 0.8750 | 0.268266 | 0.264260 | 0.268207 | 4.01e-003 | 5.92e-005 |

**Table 11** : Example 5: Comparison of accuracy and efficiency between the TPS-IRBFN method and the TPS-DRBFN method [Zerroukat, Djidjeli, and Charafi (2000)] at $t = 1$. With the same number of collocation points, the TPS-IRBFN method outperforms the TPS-DRBFN method while the number of time steps it uses ($Nts$=25) is half as many as that required by the TPS-DRBFN method. The $L_\infty$-norm of the error vector for TPS-DRBFN and TPS-IRBFN are $9.8935e-3$ and $1.2150e-3$, respectively.

| | | Numerical solution | | Absolute Error | |
|---|---|---|---|---|---|
| X | Analytical | DRBFN | TPS-IRBFN | DRBFN | TPS-IRBFN |
| 0.2500 | 0.737486 | 0.747380 | 0.737512 | 9.89e-003 | 2.50e-005 |
| 0.3125 | 0.666554 | 0.675000 | 0.666701 | 8.45e-003 | 1.47e-004 |
| 0.3750 | 0.602444 | 0.608810 | 0.602286 | 6.37e-003 | 1.58e-004 |
| 0.5000 | 0.492129 | 0.493910 | 0.491800 | 1.78e-003 | 3.28e-004 |
| 0.6250 | 0.402014 | 0.400190 | 0.401410 | 1.82e-003 | 6.03e-004 |
| 0.7500 | 0.328400 | 0.324740 | 0.327574 | 3.66e-003 | 8.26e-004 |
| 0.8750 | 0.268266 | 0.264260 | 0.267051 | 4.01e-003 | 1.22e-003 |

and TPS, respectively. As can be seen from the tables, the IRBFN method gives more accurate result with a smaller number of points. It should be noted that for the FEM, a numerical interpolation is required to calculate the solution at some non-nodal points after obtaining the solution on the triangulated mesh. For the IRBFN method, with its unstructured configuration, particular points of interest can be inserted into the computational grid at early stages before solving the PDEs so that interpolation is not needed. This can be done with the IRBFN method at no additional cost. Alternatively, the solution at any point **x** at time $t^n$ can be obtained simply by evaluating (6) after the solving process has been completed for time level $t^n$.

### 4.5 Example 5. One-dimensional advection-diffusion equation

Consider the 1D advection-diffusion equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + v \frac{\partial u}{\partial x}, \quad 0 < x < 1, \quad t > 0,$$

subject to boundary and initial conditions

$$\begin{aligned} u(0,t) &= a e^{bt}, \quad t > 0, \\ u(1,t) &= a e^{bt-c}, \quad t > 0, \\ u(x,0) &= a e^{-cx}. \end{aligned}$$

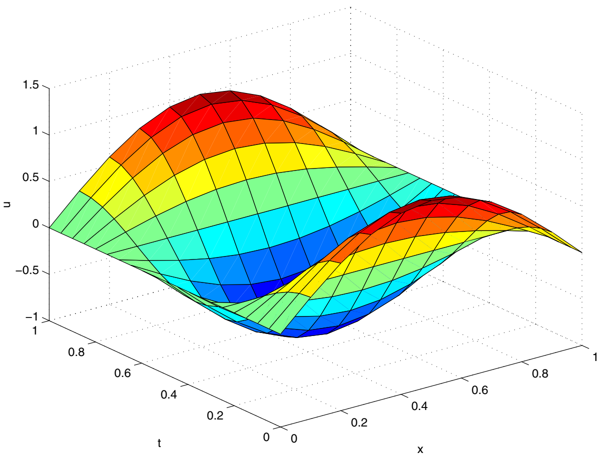

**Figure 7** : Solution profile of the problem in Example 3 by the MQ-IRBFN method.

The analytical solution is given by

$$\begin{aligned} u(x,t) &= ae^{bt-cx}, \\ c &= \frac{v \pm \sqrt{v^2+4\kappa b}}{2\kappa} > 0. \end{aligned}$$

The DRBFN and IRBFN methods are used to solve the problem in this example. For comparison purposes, the solution to the problem by the DRBFN method is obtained from [Zerroukat, Djidjeli, and Charafi (2000)] where the implicit scheme, namely ICNTPS, was reported to be the most efficient among the other schemes. The ICNTPS scheme uses TPS for spatial discretization and $\theta$-scheme with $\theta = 0.5$ for time integration. For the IRBFN method, a semi-discrete scheme based on the fourth-order Runge-Kutta scheme is used together with either MQ or TPS to solve this problem. Solutions at time $t = 1.0$ by direct and indirect methods are shown in Tables 10 and 11 where $\kappa = 0.1$, $b = 0.1$, $c = 1.61803$, $a = 1.0$, and $v = 0.1$. It can be seen from the tables that, with the same number of collocation points, the semi-discrete scheme using either MQ or TPS outperforms the ICNTPS scheme while the number of time steps used in the present IRBFN method is half as many as that required by the DRBFN method.

## 5 Conclusions

A recent development of the IRBFN method for solving transient problems has been discussed in this paper. Two numerical schemes combining the IRBFN method with different time integration schemes based on either fully or semi-discrete framework have been proposed. The IRBFN method is implemented and verified with Hardy's multiquadrics and Duchon's thin plate splines. For the problems considered in this work, the method shows its superiority over other numerical methods such as FDM, FEM, BEM and the DRBFN method in term of accuracy and efficiency. In particular, the IRBFN method can still give more accurate solutions even with coarser discretization. It is also observed that no special treatment is required in order to capture the boundary layers. In other words, the point density near the boundary is the same as in the rest of the domain and the solution there is well behaved.

Although the MQ scheme can yield better results than TPS, its accuracy depends on a user-defined parameter, $\beta$. Sensitivity studies of the IRBFN method to the shape parameter as well as other network parameters such as point density and time step size, have been performed extensively in this work. It is recommended that the TPS-IRBFN method be used for most problems, and the MQ-IRBFN method be used for problems with more stringent tolerances.

The results obtained in this work show the present IRBFN method is promising and further development is being undertaken. As in DRBFN method, the present method might face difficulties with ill-conditioned system in large scale problems. An implementation of the IRBFN method using domain decomposition for transient problems is in progress in which the original computational domain can be decomposed into a number of subdomains with associated system matrices having smaller condition number and the problem can be solved in a parallel manner. In addition, a combination between the IRBFN and level set method is preliminarily implemented for tracking moving interfaces [Mai-Cao and Tran-Cong (2003)] and simulating multi-phase flows.

**Acknowledgement:** L. Mai-Cao is supported by a USQ scholarship, USQ Faculty of Engineering and Surveying scholarship and a Moldflow scholarship. The authors would like to thank Professor C.S. Chen for bringing Reference [Zebib (1984)] to their attention and the referees for their helpful comments.

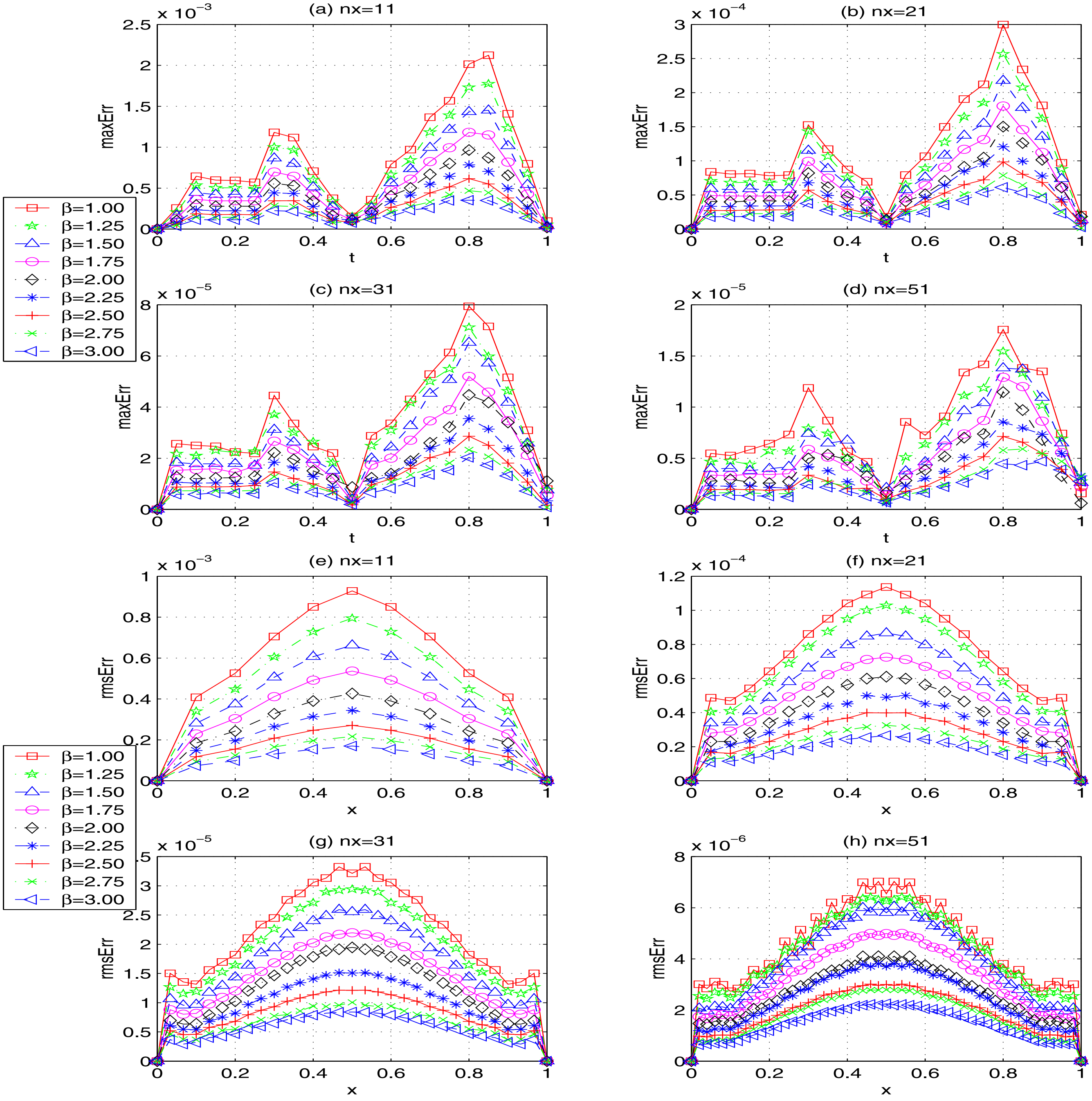

**Figure 8** : Sensitivity analysis of the MQ-IRBFN method for the problem in Example 3. The numerical solution yielded by the proposed method maintains its high accuracy with the root-mean-squared error remaining within an order of magnitude for the range of β's values from 1 to 3.

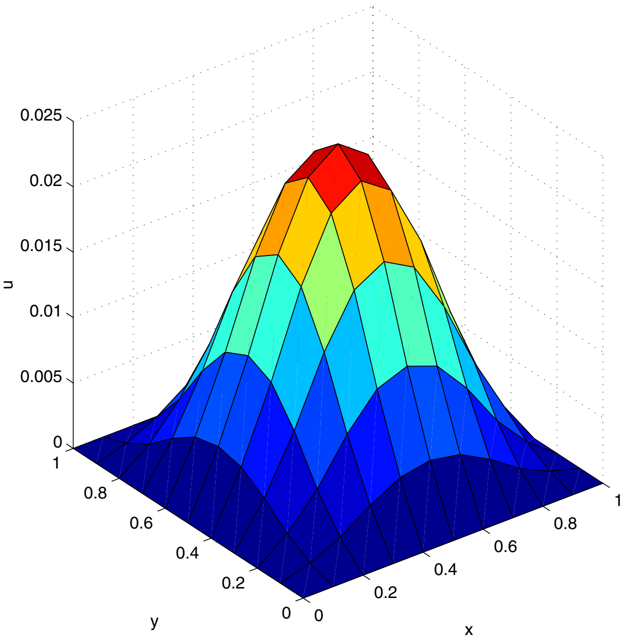

**Figure 9** : Numerical solution by the MQ-IRBFN method at time step 5 for the problem in Example 4.

## References

**Atluri, S.; Han, Z.; Rajendran, A.** (2004): A new implementation of the meshless finite volume method, through the mlpg ”mixed” approach. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6, no. 6, pp. 491–513.

**Atluri, S.; Shen, S.** (2002): *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Encino, USA.

**Atluri, S.; Shen, S.** (2002): The meshless local Petrov-Galerkin (MLPG) method: A simple and less-costly alternative to the finite element and boundary element methods. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, no. 1, pp. 11–52.

**Atluri, S.; Zhu, T.** (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, vol. 22, pp. 117–127.

**Atluri, S.; Zhu, T.** (1998): A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computational modeling and simulation. *Modeling Simulation in Engineering*, vol. 3, pp. 187–196.

**Beatson, R.; Powell, M.** (1990): Univariate multiquadric approximation: Quasi-interpolation to scattered data. Technical Report DAMTP 1990/NA7, University of Cambridge, 1990.

**Carlson, R.; Foley, T.** (1991): The parameter $r^2$ in multiquadric interpolation. *Computers and Mathematics with Applications*, vol. 21, no. 9, pp. 29–42.

**Chen, C.; Golberg, M.; Rashed, Y.** (1998): A meshfree method for linear diffusion equations. *Numerical Heat Transfer, Part B*, vol. 33, pp. 469–486.

**Fasshauer, G.** (1996): Soving partial differential equations by collocation with radial basis functions. In Le Mehaute, A.; Rabut, C.; Schumaker, L.(Eds): *Proceedings of Chamonix*, pp. 1–8, Nashville. Vanderbilt University Press.

**Franke, R.** (1982): Scattered data interpolation: Test of some methods. *Mathematics of Computation*, vol. 38, no. 157, pp. 181–200.

**Golberg, M.; Chen, C.** (1999): *The Method of Fundamental Solutions for Potential, Helmholtz and Diffusion Problems*, chapter 4, Boundary Integral Methods: Numerical and Mathematical Aspects, pp. 105–176. WIT Press and Computational Mechanics Publications, Boston, Southampton, 1999.

**Golberg, M.; Chen, C.** (2001): An efficient meshfree method for nonlinear reaction-diffusion equations. *CMES: Computer Modeling in Engineering and Sciences*, vol. 2, no. 1, pp. 87–95.

**Han, Z.; Atluri, S.** (2004): A meshless local petrov-galerkin (mlpg) approach for 3-dimensional elastodynamics. *CMC:Computer Materials and Continua*, vol. 1, no. 2, pp. 129–140.

**Hon, Y.** (2002): A quasi-radial basis functions method for American options pricing. *Computer and Mathematics with Applications*, vol. 43, no. 3/5, pp. 513–524.

**Hon, Y.; Cheung, K.; Mao, X.; Kansa, E.** (1999): Multiquadric solution for shallow water equations. *ASCE Journal of Hydraulic Engineering*, vol. 125, no. 5, pp. 524–533.

**Ingber, M.; Phan-Thien, N.** (1992): A boundary element approach for parabolic differential equations using a class of particular solutions. *Applied Mathematical Modelling*, vol. 16, pp. 124–132.

**Kansa, E.** (1990): Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics–I. Surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, vol. 19, no. 8/9, pp. 127–145.

**Kansa, E.** (1990): Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics–II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, vol. 19, no. 8/9, pp. 147–161.

**Kovacevic, I.; Poredos, A.; Sarler, B.** (2003): Solving the Stefan problem with the radial basis function collocation method. *Numerical Heat Transfer, Part B: Fundamentals*, vol. 44, pp. 1–24.

**Mai-Cao, L.; Tran-Cong, T.** (2003): A meshless level set approach to interface capturing. In Atluri, S.; Beskos, D.; Polyzos, D.(Eds): *Proceedings of the 2003 International Conference on Computational & Experimental Engineering & Sciences*, Encino. Tech Science Press.

**Mai-Duy, N.; Tran-Cong, T.** (2001): Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks*, vol. 14, pp. 185–199.

**Mai-Duy, N.; Tran-Cong, T.** (2001): Numerical solution of navier-stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids*, vol. 37, pp. 65–86.

**Mai-Duy, N.; Tran-Cong, T.** (2003): Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling*, vol. 27, pp. 197–220.

**Moody, J.; Darken, C.** (1989): Fast learning in networks of locally-tuned processing units. *Neural Computations*, vol. 1, pp. 281–294.

**Moridis, G.; Kansa, E.** (1994): The Laplace transform multiquadradric method: A highly accurate scheme for the numerical solution of linear partial differential equations. *Journal of Applied Science and Computations*, vol. 1, no. 2, pp. 375–407.

**Perko, J.; Chen, C.; Sarler, B.** (2001): A polygon-free numerical solution of steady natural convection in solid-liquid systems. In Sarler, B.; Brebbia, C.(Eds): *Moving Boundaries VI: Computational Modelling of Free and Moving Boundary Problems, (Computational and Experimental Methods, Vol. 4)*, pp. 111–122, Boston. WIT Press.

**Quarteroni, A.; Sacco, R.; Saleri, F.** (2000): *Numerical Mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, New York, USA.

**Rippa, S.** (1999): An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, vol. 11, pp. 193–210.

**Sarler, B.; Perko, J.; Chen, C.** (2004): Radial basis function collocation method solution of natural convection in porous media. *International journal of numerical methods for heat and fluid flow*, vol. 14, no. 2, pp. 187–212.

**Sarler, B.; Perko, J.; Chen, C.; Kuhn, G.** (2001): A meshless approach to natural convection. In Atluri, S.; Nishioka, T.; Kikuchi, M.(Eds): *Advances in Computational Engineering & Sciences, ICES'01: [Proceedings of the International Conference on Computational Engineering & Sciences, 19-25 August 2001, Puerto Vallarta, Mexico]*, Palmdale, CA. Tech Science Press.

**Zebib, A.** (1984): A Chebyshev method for the solution of boundary value problems. *Journal of Computational Physics*, vol. 53, pp. 443–455.

**Zerroukat, M.; Djidjeli, K.; Charafi, A.** (2000): Explicit and implicit meshless methods for linear advection-diffusion-type partial differential equations. *International Journal for Numerical Methods in Engineering*, vol. 48, pp. 19–35.

**Zerroukat, M.; Power, H.; Chen, C.** (1998): A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering*, pp. 1263–1278.

### *Appendix 1: The first and second order antiderivatives of Hardy's multiquadrics*

This appendix presents the symbolic integrations of 2D MQ-RBF

$\varphi_i = \sqrt{r_i^2 + s_i^2}$,

where $r_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ and $s_i$ is a shape parameter associated with the $i^{th}$ center. Only the antiderivatives are described here. The constants of integration are presented in Appendix 3.

$$\int \varphi_i \, dx = \frac{1}{2}(x-x_i)\sqrt{r_i^2+s_i^2} + \frac{1}{2}\left((y-y_i)^2+s_i^2\right) \ln\left((x-x_i)+\sqrt{r_i^2+s_i^2}\right),$$

$$\int \varphi_i \, dy = \frac{1}{2}(y-y_i)\sqrt{r_i^2+s_i^2} + \frac{1}{2}\left((x-x_i)^2+s_i^2\right) \ln\left((y-y_i)+\sqrt{r_i^2+s_i^2}\right),$$

$$\iint \varphi_i \, dx dx = \frac{1}{6}\left((x-x_i)^2-2(y-y_i)^2-2s_i^2\right)\sqrt{r_i^2+s_i^2}+ \frac{1}{2}(x-x_i)\left((y-y_i)^2+s_i^2\right)\ln\left((x-x_i)+\sqrt{r_i^2+s_i^2}\right),$$

$$\iint \varphi_i \, dy dy = \frac{1}{6}\left((y-y_i)^2-2(x-x_i)^2-2s_i^2\right)\sqrt{r_i^2+s_i^2}+ \frac{1}{2}(y-y_i)\left((x-x_i)^2+s_i^2\right)\ln\left((y-y_i)+\sqrt{r_i^2+s_i^2}\right),$$

$$\iint \varphi_i \, dx dy = -\frac{1}{18}(y-y_i)\left((y-y_i)^2+6s_i^2-6(x-x_i)\sqrt{r_i^2+s_i^2}\right) + \frac{1}{3}s_i^3\left(\tan^{-1}\frac{y-y_i}{s_i}-\tan^{-1}\frac{(x-x_i)(y-y_i)}{s_i\sqrt{r_i^2+s_i^2}}\right) + \frac{1}{6}(y-y_i)\left((y-y_i)^2+3s_i^2\right)\ln\left((x-x_i)+\sqrt{r_i^2+s_i^2}\right) + \frac{1}{6}(x-x_i)\left((x-x_i)^2+3s_i^2\right)\ln\left((y-y_i)+\sqrt{r_i^2+s_i^2}\right)$$

### *Appendix 2: The first and second order antiderivatives of Duchon's thin plate splines*

This appendix presents the symbolic integrations of 2-dimensional first order TPS-RBF

$$\varphi_i = r_i^2 \ln(r_i),$$

where $r_i = \sqrt{(x-x_i)^2+(y-y_i)^2}$. Only the antiderivatives are described here. The constants of integration are presented in Appendix 3.

$$\int \varphi_i \, dx = \frac{2}{3}(y-y_i)^3\tan^{-1}\left(\frac{x-x_i}{y-y_i}\right) - \frac{1}{9}(x-x_i)\left((x-x_i)^2+6(y-y_i)^2\right) + \frac{1}{3}(x-x_i)\left((x-x_i)^2+3(y-y_i)^2\right)\ln(r_i),$$

$$\int \varphi_i \, dy = \frac{2}{3}(x-x_i)^3\tan^{-1}\left(\frac{y-y_i}{x-x_i}\right) - \frac{1}{9}(y-y_i)\left(6(x-x_i)^2+(y-y_i)^2\right) + \frac{1}{3}(y-y_i)\left(3(x-x_i)^2+(y-y_i)^2\right)\ln(r_i),$$

$$\iint \varphi_i \, dx dx = -\frac{1}{144}(x-x_i)^2\left(7(x-x_i)^2+78(y-y_i)^2\right) + \frac{2}{3}(x-x_i)(y-y_i)^3\tan^{-1}\left(\frac{x-x_i}{y-y_i}\right) - \frac{1}{3}(y-y_i)^4\ln\left(1+\frac{(x-x_i)^2}{(y-y_i)^2}\right) + \frac{1}{12}r_i^2\left((x-x_i)^2+5(y-y_i)^2\right)\ln(r_i),$$

$$\iint \varphi_i \, dy dy = -\frac{1}{144}(y-y_i)^2\left(78(x-x_i)^2+7(y-y_i)^2\right) + \frac{2}{3}(x-x_i)^3(y-y_i)\tan^{-1}\left(\frac{y-y_i}{x-x_i}\right) - \frac{1}{3}(x-x_i)^4\ln\left(1+\frac{(y-y_i)^2}{(x-x_i)^2}\right) + \frac{1}{12}r_i^2\left(5(x-x_i)^2+(y-y_i)^2\right)\ln(r_i),$$

$$\iint \varphi_i \, dx dy = \frac{1}{6}(x-x_i)^4\tan^{-1}\left(\frac{y-y_i}{x-x_i}\right) + \frac{1}{6}(y-y_i)^4\tan^{-1}\left(\frac{x-x_i}{y-y_i}\right) + \frac{1}{18}(x-x_i)(y-y_i)r_i^2\left(6\ln(r_i)-5\right).$$

### *Appendix 3: Procedure for calculating constants of integration*

The computational procedure is described for 3-dimensional problems which can be applied easily to lower-dimensional problems. The computation is started with (10) where the second-order derivative of $\phi$ is chosen to be a radial basis function. In order to describe the computation in full details where the constants of integration and their corresponding intermediate values can be obtained, (8) is rewritten in the form of a combination of basis functions taking into account (5) and (10) to give

$$\tilde{u}^n_{,jk}(\xi_1,\xi_2,\xi_3)=\sum_{i=1}^{N} w_i^n\,\varphi_i(\xi_1,\xi_2,\xi_3), \quad j,k=1,2,3, \tag{34}$$

Integrating (34) twice in $j$ and $k$ directions yields

$$\tilde{u}^n_{,k}(\xi_1,\xi_2,\xi_3)=\sum_{i=1}^{N} w_i^n\int\varphi_i(\xi_1,\xi_2,\xi_3)d\xi_j + C_1^n(\xi_p,\xi_q), \quad p\neq q,\ p\neq j,\ q\neq j, \tag{35}$$

$$\tilde{u}^n(\xi_1,\xi_2,\xi_3)=\sum_{i=1}^{N} w_i^n\iint\varphi_i(\xi_1,\xi_2,\xi_3)d\xi_j d\xi_k + C_1^n(\xi_p,\xi_q)\,\xi_k+C_2^n(\xi_r,\xi_s), \quad r\neq s,\ r\neq k,\ s\neq k, \tag{36}$$

where $\xi_i$, $i=1,2,3$ is the $i^{th}$ coordinate (for example, $(\xi_1,\xi_2,\xi_3)\equiv(x,y,z)$ in Cartesian coordinate system); $\tilde{u}^n$, $\tilde{u}^n_{,k}$ and $\tilde{u}^n_{,jk}$ are the IRBFN approximations to the function $u$ and its derivatives at time $t=t^n$, respectively. Formulations for the antiderivatives of Hardy's multiquadrics and Duchon's thin plate splines are presented in Appendix 1 and 2. It should be noted that $C_1^n$, $C_2^n$ are the constants of integration along the $j^{th}$ and $k^{th}$ directions at time $t^n$, respectively. As they can be calculated in the same way, only $C_1^n$ is presented here. Since $C_1^n$ is a function in terms of $(\xi_p,\xi_q)$, this "constant", in turn, can be approximated by integrating twice in direction $l,m\neq j$ using IRBFN approximants as follows

$$C^n_{1,lm}(\xi_p,\xi_q)=\sum_{i=1}^{N_{pq}}\hat{w}_i^n\,\varphi_i(\xi_p,\xi_q), \quad l,m=1,2,3;\ l,m\neq j, \tag{37}$$

$$C^n_{1,m}(\xi_p,\xi_q)=\sum_{i=1}^{N_{pq}}\hat{w}_i^n\int\varphi_i(\xi_p,\xi_q)d\xi_l + \hat{C}_1^n(\xi_v), \quad v\neq j,l, \tag{38}$$

$$C^n_{1}(\xi_p,\xi_q)=\sum_{i=1}^{N_{pq}}\hat{w}_i^n\iint\varphi_i(\xi_p,\xi_q)d\xi_l d\xi_m + \hat{C}_1^n(\xi_v)\,\xi_m+\hat{C}_2^n(\xi_w), \quad w\neq j,m, \tag{39}$$

where $N_{pq}$ is the number of distinct points in $(\xi_p,\xi_q)$ coordinates.

The computation of $\hat{C}_1^n$ and $\hat{C}_2^n$ is performed in the same manner as in steps (37)-(39). This time however, because $\hat{C}_1^n$ (or $\hat{C}_2^n$) depends on only one spatial variable $\xi_v$ (or $\xi_w$), it will be determined by integrating twice in direction $v$ (or $w$) as follows

$$\begin{aligned}
\hat{C}^n_{1,vv}(\xi_v) &= \sum_{i=1}^{N_v}\tilde{w}_i^n\,\varphi_i(\xi_v), \quad v=1,2,3;\ v\neq j,l, && (40)\\
\hat{C}^n_{1,v}(\xi_v) &= \sum_{i=1}^{N_v}\tilde{w}_i^n\int\varphi_i(\xi_v)d\xi_v+\tilde{C}_1^n, && (41)\\
\hat{C}^n_{1}(\xi_v) &= \sum_{i=1}^{N_v}\tilde{w}_i^n\iint\varphi_i(\xi_v)d\xi_v d\xi_v+\tilde{C}_1^n\xi_v+\tilde{C}_2^n, && (42)
\end{aligned}$$

where $N_v$ is the number of distinct points in $\xi_v$ coordinate, $v\neq j,l$.

It is noted that for 3D problems, $\tilde{C}_1^n$ and $\tilde{C}_2^n$ are really constants having arbitrary values at time $t^n$. For simplicity, one can choose $\tilde{C}_1^n=\tilde{C}_2^n=0$. In that case, $\hat{C}_1^n$ is reduced to

$$\hat{C}_1^n=\sum_{i=1}^{N_v}\tilde{w}_i^n\iint\varphi_i(\xi_v)d\xi_v d\xi_v. \tag{43}$$

$\hat{C}_2^n$ is approximated in the same manner as $\hat{C}_1^n$ is. As a result, $C_1^n$ can be calculated by substituting $\hat{C}_1^n$ and $\hat{C}_2^n$ into (39). The same procedure can be applied to calculate $C_2^n$. Finally, $C_1^n$ and $C_2^n$ are substituted into (36) which can be rewritten into the compact form (6).

It should be noted that for lower-dimensional problems, the computational procedure is much simpler. In particular, for 2-D problems, $C_1^n$ and $C_2^n$ can be approximated directly from (39) in the same manner as in (43) whereas for 1-D problems, one doesn't have to do any computation for $\tilde{C}_1^n$, $\tilde{C}_2^n$, $\hat{C}_1^n$ and $\hat{C}_2^n$ because $C_1^n$ and $C_2^n$ are already constants at time $t^n$.