FDMFS for Diffusion Equation with Unsteady Forcing Function

S.P. Hu[1] , D.L. Young[2] and C.M. Fan[1]

Abstract: In this paper, a novel numerical scheme called (FDMFS), which combines the finite difference method (FDM) and the method of fundamental solutions (MFS), is proposed to simulate the nonhomogeneous diffusion problem with an unsteady forcing function. Most meshless methods are confined to the investigations of nonhomogeneous diffusion equations with steady forcing functions due to the difficulty to find an unsteady particular solution. Therefore, we proposed a FDM with Cartesian grid to handle the unsteady nonhomogeneous term of the equations. The numerical solution in FDMFS is decomposed into a particular solution and a homogeneous solution. The particular solution is constructed using the FDM in an artificial regular domain which contains the real irregular domain without boundary conditions, and the homogeneous solution can be obtained by the time-space unification MFS in the irregular domain with boundary conditions. Besides, the Cartesian grid for particular solution is very simple to generate automatically. Our paper is the first time to propose an algorithm to solve nonhomogeneous diffusion equations with unsteady forcing functions using MFS to solve homogeneous solutions and FDM to calculate the particular solutions. Numerical experiments are presented for 2D problems in regular and irregular domains to show the high performance of this proposed scheme. Moreover, the stabilities of explicit and implicit FDM for particular solution are analyzed. Numerical studies suggest that the proposed FDMFS can speed up the simulation and save the CPU time and memory storage substantially.

Keyword: meshless; unsteady forcing function; nonhomogeneous diffusion equation; method of fundamental solutions; finite difference method; FDMFS.

1 Introduction

The diffusion equations are very important subjects for sciences and engineering and usually applied to describe the problems of heat transfer, pollution transports, chemical processes, etc. Some external input of energy is represented in the form of a forcing function by the nonhomogeneous term. Classical numerical methods, such as FDM, finite element method (FEM) and finite volume method (FVM), had been extensively adopted to simulate the diffusion equations. However, all of them are mesh-dependent methods which need mesh generation. The FDM needs the coordinate transformations to treat irregular domain problems. The associated bookkeeping of the elements and nodes is also cumbersome and expensive in the CPU time and computer memory for the FEM and FVM. Recently, there are a lot of researchers developed the simpler meshless or meshfree methods to solve the diffusion equations.

Meshless numerical method is a new developed tool for solving irregular domain and homogeneous problems. As the name implies, the meshless methods only require nodes for boundary and initial conditions instead of mesh. The MFS, also known as F-Treffz method or singularity method was originally presented by Kupradze and Aleksidze (1964). In previous studies, the MFS was applied widely to simulate a lot of physical problems, for example the Helmholtz equations [Chen, Fan, Young, Murugesan and Tsai (2005), Young and Ruan (2005), Chen, Chen and Kao (2006)]; potential problems [Liu, Nishimura and

[1] Department of Civil Engineering & Hydrotech Research Institute, National Taiwan University, Taipei, Taiwan.

[2] Correspondence to: Prof. D.L. Young, Fax: +886-2-23626114, E-mail: dlyoung@ntu.edu.tw

Yao (2005), Young, Chen, Chen and Kao (2007)]; Stokes flows [Tsai, Young and Cheng (2002), Young, Chiu, Fan, Tsai and Lin (2006), Young, Jane, Fan, Murugesan and Tsai (2006)] and diffusion problems [Young, Tsai and Fan (2004), Hu, Fan, Chen and Young (2005)].

There are a lot of researches on meshless methods for diffusion problems, such as radial basis functions (RBFs) collocation method or the Kansa's method [Kansa (1990)]; indirect radial basis function network (IRBFN) method [Mai-Cao and Tran-Cong (2005)]; meshless local Petrov-Galerkin (MLPG) method [Lin and Atluri (2000), Sladek, Sladek and Tanaka (2005)] and the MFS [Chen, Golberg and Hon (1998a), Young, Tsai, Murugesan, Fan and Chen (2004), Cho, Golberg, Muleshkov and Li (2004)]. MFS utilizes the fundamental solutions of PDEs and can reduce one dimensionality of the problem. Therefore, it is very popular and powerful in the realm of computational sciences.

For solving homogeneous diffusion equations by MFS, it is usual to employ the finite difference discretization or Laplace transform to deal with the time derivative. Golberg and Chen (1998) used the MFS based on modified Helmholtz fundamental solution to simulate the nonhomogeneous term via associating with the dual reciprocity method (DRM). The Chebyshev interpolation functions [Golberg, Muleshkov, Chen and Cheng (2003)] are also suggested to approximate the right-hand side of modified Helmholtz equations for diffusion problems. Instead of using the finite difference scheme or the Laplace transform to deal with the time derivative term in diffusion equation, the time-dependent diffusion fundamental solution can be used directly by the MFS for the homogeneous diffusion solutions [Young, Tsai, Murugesan, Fan and Chen (2004), Hon and Wei (2005)].

In fact, the MFS is only effective at solving homogeneous PDEs. In order to extend the MFS to solve nonhomogeneous PDEs, we have to combine MFS with other discretized numerical schemes such as stated in the followings. When the PDEs have steady forcing functions, Burgess and Mahaherin (1987) constructed the particular solutions by direct numerical domain integration. Besides, Chen (1995) and Chen, Golberg and Hon (1998b) employed the quasi-Monte Carlo (QMC) quadrature as numerical integration to find the particular solutions. Golberg (1995) suggested the MFS to solve Poisson's equation by approximating the forcing function using thin plate splines (TPSs). Thereafter the MFS is extended to nonhomogeneous PDEs with steady forcing function commonly by combining the DRM where the steady forcing function is approximated by a finite series of RBFs [Balakrishnan and Ramachandran (2001), Alves and Chen (2005), Wang, Qin and Kang (2005)]. Young, Tsai and Fan (2004) extended the time-dependent diffusion MFS-DRM model to solve multidimensional nonhomogeneous diffusion problems and they also gave a comparison between their proposed scheme and Golberg and Chen's researches (1998). Recently, Young, Chen, Fan and Tsai (2006) proposed another numerical scheme (MFS-MPS-EEM), which combines the MFS, the method of particular solutions (MPS) and eigenfunction expansion method (EEM), to simulate the diffusion equation. Unfortunately, all of these methods can not be applied directly for nonhomogeneous diffusion problems with time-dependent forcing functions.

As far as the unsteady force functions are concerned there is still no literature available. We thereby in this paper first propose the time-dependent MFS for nonhomogeneous diffusion equation and combined with FDM to take care unsteady forcing functions. The solution of a nonhomogeneous PDE can be split into the summation of a particular solution and a homogeneous solution by the linear superposition theory. We will use the FDM to solve the particular solution in an artificial Cartesian grid which contains the physical domain without considering boundary conditions; and the time-dependent diffusion MFS to solve the homogeneous solutions with boundary conditions in physical domain. In other words, the FDM is performed in an artificial regular domain to handle the unsteady forcing function; in the meantime, the homogeneous diffusion equation can be analyzed by the MFS free from mesh generation and numerical quadrature in an irregular physical domain. Those are the strong features respectively for both the meshless MFS and mesh FDM. The concept is similar to the research of Chantasiriwan (2004) who combined the FDM and MFS to solve the steady Poisson problem. If we directly use the FDM to solve the nonhomogeneous diffusion equations we have to face a difficulty to handle the irregular domain problem which is not a trivial task as far as coordinate transform is concerned, see the Section 4.2 and appendix for details. In other words we have made use of both the advantages of meshless method to solve the homogeneous solution and the discretized FDM to calculate the particular solution in order to deal with the more complicated nonhomogeneous diffusion equations with unsteady forcing functions.

The aim of this study is to demonstrate the capability of the proposed FDMFS for nonhomogeneous diffusion equation with unsteady forcing functions. The governing equations and numerical methods will be explained in sections 2 and 3, respectively. We also give a detailed discussion on processing for forcing function for irregular-domain problem in section 3.2. The numerical results and conclusions will be provided separately in sections 4 and 5. There are six problems adopted in the paper and the numerical results are compared well with the analytical solutions.

2 Governing Equations

The diffusion equation with unsteady forcing function over the problem domain Ω with boundary Γ can be written as follows:

$$\frac{\partial T(\vec{x},t)}{\partial t} = k\nabla^2 T(\vec{x},t) + F(\vec{x},t) \tag{1}$$

where $\vec{x}$ is the general spatial coordinate, t is the time coordinate, k is the diffusion coefficient, $F(\vec{x},t)$ is the unsteady forcing function, and $T(\vec{x},t)$ is the scalar variable to be determined. The initial condition of the problem is

$$T(\vec{x},t_0) = f_1(\vec{x}) \text{ in } \Omega \tag{2}$$

with the Dirichlet and Neumann boundary conditions.

$$T(\vec{x},t) = f_2(\vec{x},t) \text{ in } \Gamma_1 \tag{3}$$

$$\frac{\partial}{\partial n} T(\vec{x},t) = f_3(\vec{x},t) \text{ in } \Gamma_2 \tag{4}$$

where Ω is the problem domain, $\Gamma_1 + \Gamma_2$ is equal to the boundary Γ, n is the outward normal direction and $f_1(\vec{x})$, $f_2(\vec{x},t)$, $f_3(\vec{x},t)$ are known functions. t_0 is the initial time.

Through the MPSMFS [Young, Tsai and Fan (2004)] or the MFS-MPS-EEM models [Young, Chen, Fan and Tsai (2006)], the diffusion equation with steady forcing function can be solved directly. However, both of this two time-dependent MFS schemes can not simulate the diffusion equation with unsteady forcing function. Therefore, we propose the FDMFS to analyze the diffusion problem with unsteady forcing function.

3 Numerical Method

3.1 Basic numerical scheme

The solution $T(\vec{x},t)$ can be written as the linear combination of a homogeneous solution $T_h(\vec{x},t)$ and a particular solution $T_p(\vec{x},t)$ shown as follows:

$$T(\vec{x},t) = T_h(\vec{x},t) + T_p(\vec{x},t) \tag{5}$$

The particular solution is obtained from the nonhomogeneous equation as shown below:

$$\frac{\partial T_p(\vec{x},t)}{\partial t} - k\nabla^2 T_p(\vec{x},t) = F(\vec{x},t) \tag{6}$$

No boundary condition has to be satisfied and initial condition can be set as an arbitrary function; the homogeneous solution is obtained from the following homogeneous equation:

$$\frac{\partial T_h(\vec{x},t)}{\partial t} - k\nabla^2 T_h(\vec{x},t) = 0 \tag{7}$$

with the modified initial condition

$$T_h(\vec{x},t_0) = f_1(\vec{x}) - T_p(\vec{x},t_0) \text{ in } \Omega \tag{8}$$

and the modified Dirichlet and Neumann boundary conditions.

$$T_h(\vec{x},t) = f_2(\vec{x},t) - T_p(\vec{x},t) \text{ in } \Gamma_1 \tag{9}$$

$$\frac{\partial}{\partial n}T_h(\vec{x},t) = f_3(\vec{x},t) - \frac{\partial}{\partial n}T_p(\vec{x},t) \text{ in } \Gamma_2 \tag{10}$$

The numerical procedures start from the particular solution. First of all, as shown in Fig. 1 (a)-(b), we need to distribute a Cartesian grid (Ω^C) which has to contain the problem domain (Ω).

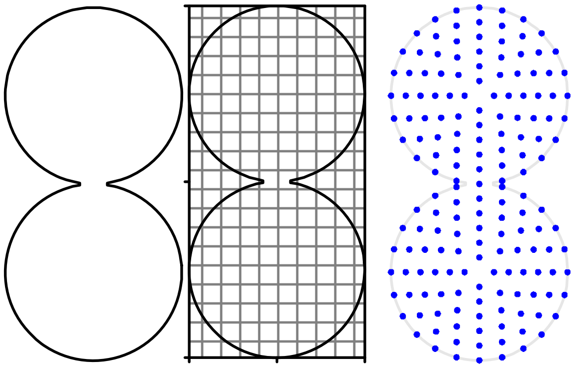

(a) (b) (c)

Figure 1: (a) Problem domain (Ω), (b) Cartesian mesh for FDM (Ω^C), (c) node distribution for MFS

Next, we can use the fully explicit FDM to simulate the particular solution by Eq. (6).

$$\begin{aligned} T_p^{n+1}(\vec{x},t) &= T_p^n(\vec{x},t) + \Delta t \cdot \\ &\left[k\left(\frac{T_{p,i+1,j}^n(\vec{x},t) - 2T_{p,i,j}^n(\vec{x},t) + T_{p,i-1,j}^n(\vec{x},t)}{(\Delta x)^2}\right)\right] \\ &+ \Delta t \cdot \\ &\left[k\left(\frac{T_{p,i,j+1}^n(\vec{x},t) - 2T_{p,i,j}^n(\vec{x},t) + T_{p,i,j-1}^n(\vec{x},t)}{(\Delta y)^2}\right)\right] \\ &+ \Delta t\left(F^n(\vec{x},t)\right) \end{aligned} \tag{11}$$

where $\Delta t = \Delta t_p$ is the time step size for particular solution, Δx is the mesh size in x direction and Δy is the mesh size in y direction. Indeed, the implicit FDM also can be adopted to obtain the particular solution

$$\begin{aligned} &T_p^{n+1}(\vec{x},t) - \Delta t \cdot \\ &\left\{k\left(\frac{T_{p,i+1,j}^{n+1}(\vec{x},t) - 2T_{p,i,j}^{n+1}(\vec{x},t) + T_{p,i-1,j}^{n+1}(\vec{x},t)}{(\Delta x)^2}\right)\right\} \\ &- \Delta t \cdot \\ &\left\{k\left(\frac{T_{p,i,j+1}^{n+1}(\vec{x},t) - 2T_{p,i,j}^{n+1}(\vec{x},t) + T_{p,i,j-1}^{n+1}(\vec{x},t)}{(\Delta y)^2}\right)\right\} \\ &= T_p^n(\vec{x},t) + \Delta t\left(F^n(\vec{x},t)\right) \end{aligned} \tag{12}$$

The initial condition of the particular solution is assumed to be an arbitrary function and the boundary condition of the particular solutions is not required. By the advantages of explicit FDM, the particular solution can be obtained in a very short time and no matrix solver is needed.

After the particular solution is obtained, the meshless MFS is considered to solve the homogenous solution. The homogeneous solution satisfies the linear diffusion equations, Eq. (7), and the modified initial and boundary conditions, Eqs. (8)-(10). In the MFS, the diffusion solution can be represented as the linear combination of the diffusion fundamental solutions with different intensities. The fundamental solution of the linear diffusion equation is governed by

$$\frac{\partial G\left(\vec{x},t;\vec{\xi},\tau\right)}{\partial t} = k\nabla^2 G\left(\vec{x},t;\vec{\xi},\tau\right) + \delta\left(\vec{x}-\vec{\xi}\right)\delta(t-\tau) \tag{13}$$

where $G\left(\vec{x},t;\vec{\xi},\tau\right)$ is the fundamental solution of the diffusion equation. $\vec{x} = (x,y)$ and $\vec{\xi} = (\xi,\eta)$ are the spatial coordinates of the field point and source point, as t and τ are the temporal coordinates of the field point and source point. $\delta()$ is the well-known Dirac delta function.

By using the integral transform theory of the above equation, the free-space Green's function or the fundamental solution of the diffusion equation can be obtained [Kythe (1996)]:

$$G\left(\vec{x},t;\vec{\xi},\tau\right) = \frac{e^{-\frac{\left(\left|\vec{x}-\vec{\xi}\right|\right)^2}{4k(t-\tau)}}}{\left(4\pi k(t-\tau)\right)^{d/2}}H(t-\tau) \tag{14}$$

where d is the spatial dimension and equal to two in this study. $H\,()$ is the Heaviside step function.

Based on the time-dependent MFS, the homogeneous solution can be expressed as the linear combination of the diffusion fundamental solutions

$$T_h(\vec{x},t) = \sum_{j=1}^{N_h} \alpha_j G\left(\vec{x},t;\vec{\xi}_j,\tau_j\right) \tag{15}$$

where N_h is the number of source points. In our numerical experiments, the numbers of field points and source points are chosen as the same, $N = N_h$, so that a square matrix equation can be formed. α_j are the unknown coefficients which denote the source intensities of the corresponding fundamental solutions.

The initial and boundary conditions of homogeneous solution are modified by the particular solution:

$$\begin{aligned} &T_h(\vec{x},t_0) = \\ &\quad T(\vec{x},t_0) - T_p(\vec{x},t_0) = f_1(\vec{x}) - T_p(\vec{x},t_0) \text{ in } \Omega \end{aligned} \tag{16}$$

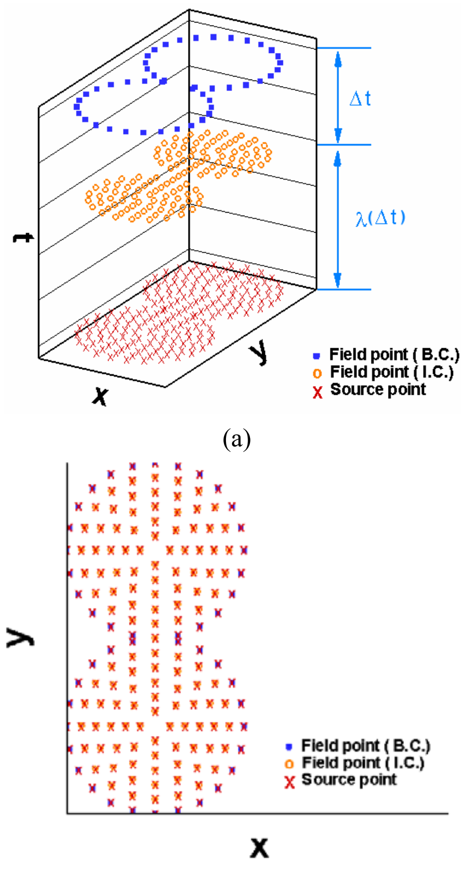
$$\begin{aligned} &T_h(\vec{x},t) = \\ &\quad T(\vec{x},t) - T_p(\vec{x},t) = f_2(\vec{x},t) - T_p(\vec{x},t) \text{ in } \Gamma_1 \end{aligned} \tag{17}$$

$$\begin{aligned} \frac{\partial}{\partial n}T_h(\vec{x},t) &= \frac{\partial}{\partial n}T(\vec{x},t) - \frac{\partial}{\partial n}T_p(\vec{x},t) \\ &= f_3(\vec{x},t) - \frac{\partial}{\partial n}T_p(\vec{x},t) \text{ in } \Gamma_2 \end{aligned} \tag{18}$$

Applying the concept of the MFS, we obtain a matrix equation as follows:

$$\begin{bmatrix} G\left(\vec{x},t_0;\vec{\xi}_j,\tau_j\right) \\ G\left(\vec{x},t;\vec{\xi}_j,\tau_j\right) \\ \frac{\partial}{\partial n}G\left(\vec{x},t;\vec{\xi}_j,\tau_j\right) \end{bmatrix} \{\alpha_j\} = \left\{ \begin{array}{c} f_1(\vec{x}) - T_p(\vec{x},t_0) \\ f_2(\vec{x},t) - T_p(\vec{x},t) \\ f_3(\vec{x},t) - \frac{\partial}{\partial n}T_p(\vec{x},t) \end{array} \right\} \tag{19}$$

Solving the above matrix equation, the coefficients α_j are obtained, and then the homogeneous solutions can be acquired by Eq. (15). Finally, the numerical solutions can be obtained by summing up the particular and homogeneous solutions of Eq. (5).



(a)

(b)

Figure 2: Schematic diagram of source and field points for the MFS based on diffusion fundamental solution (a) in a time-space coordinate (b) in a space coordinate

As shown in Fig. 1 (c), the meshless MFS requires only field points for boundary and initial conditions without mesh. The locations of field and source points of MFS are illustrated as Fig. 2, and the field and source points are located at the same spatial positions but different time levels. In Fig. 2 (a), the parameter, λ, is chosen as a function of the maximum distance of the spatial domain (R) and it can be expressed as $\lambda\,(\Delta t) = \mu R$. μ is an adaptive parameter which can be chosen by the trial and error process and is equal to 0.5 in this study. More detailed discussions on this formula can be found in a previous research [Young, Fan, Hu and Atluri (2007)].

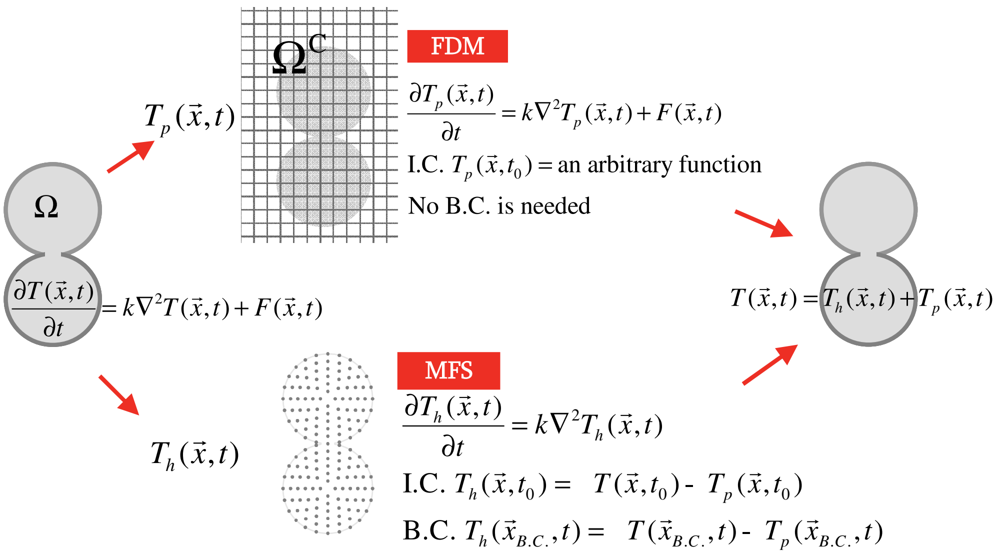

Ω
$\frac{\partial T(\vec{x},t)}{\partial t} = k\nabla^2 T(\vec{x},t) + F(\vec{x},t)$
$T_p(\vec{x},t)$
Ω^C
FDM
$\frac{\partial T_p(\vec{x},t)}{\partial t} = k\nabla^2 T_p(\vec{x},t) + F(\vec{x},t)$
I.C. $T_p(\vec{x},t_0)$ = an arbitrary function
No B.C. is needed
$T(\vec{x},t) = T_h(\vec{x},t) + T_p(\vec{x},t)$
$T_h(\vec{x},t)$
MFS
$\frac{\partial T_h(\vec{x},t)}{\partial t} = k\nabla^2 T_h(\vec{x},t)$
I.C. $T_h(\vec{x},t_0) = T(\vec{x},t_0) - T_p(\vec{x},t_0)$
B.C. $T_h(\vec{x}_{B.C.},t) = T(\vec{x}_{B.C.},t) - T_p(\vec{x}_{B.C.},t)$

Figure 3: The illustration of the proposed numerical scheme (FDMFS)

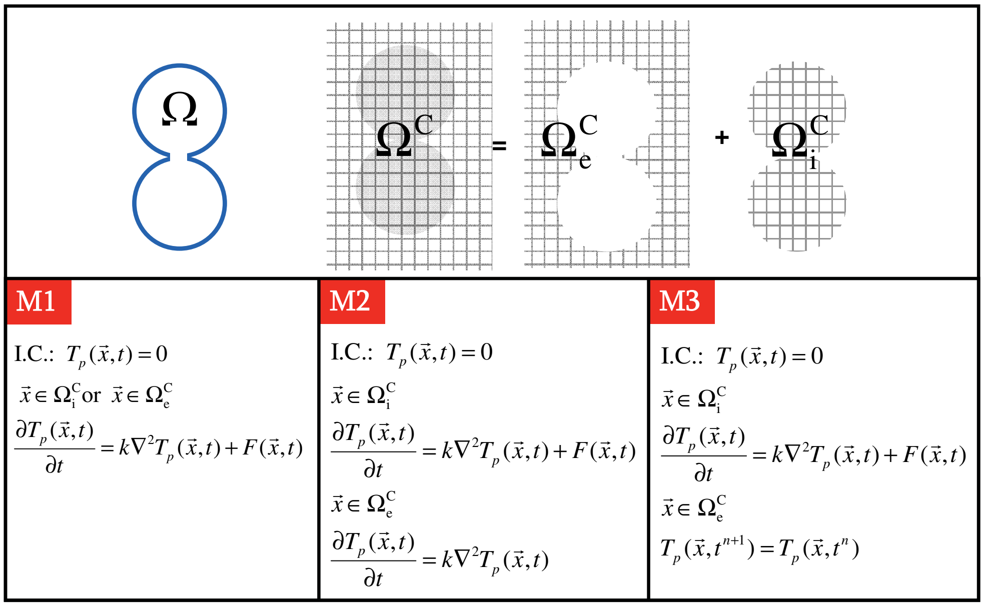

Ω
Ω^C
=
Ω_e^C
+
Ω_i^C
M1
I.C.: $T_p(\vec{x},t)=0$
$\vec{x}\in\Omega_i^C$ or $\vec{x}\in\Omega_e^C$
$\frac{\partial T_p(\vec{x},t)}{\partial t} = k\nabla^2 T_p(\vec{x},t) + F(\vec{x},t)$
M2
I.C.: $T_p(\vec{x},t)=0$
$\vec{x}\in\Omega_i^C$
$\frac{\partial T_p(\vec{x},t)}{\partial t} = k\nabla^2 T_p(\vec{x},t) + F(\vec{x},t)$
$\vec{x}\in\Omega_e^C$
$\frac{\partial T_p(\vec{x},t)}{\partial t} = k\nabla^2 T_p(\vec{x},t)$
M3
I.C.: $T_p(\vec{x},t)=0$
$\vec{x}\in\Omega_i^C$
$\frac{\partial T_p(\vec{x},t)}{\partial t} = k\nabla^2 T_p(\vec{x},t) + F(\vec{x},t)$
$\vec{x}\in\Omega_e^C$
$T_p(\vec{x},t^{n+1}) = T_p(\vec{x},t^n)$

Figure 4: The illustration of three suggestions (M1, M2 and M3) for dealing with the particular solutions

Although the time increment of FDM scheme Δt_p should satisfy the stability condition, the homogeneous solution does not need to be solved at each time step. In other words, the time interval for the MFS can use a large one. In this paper, we adopt $\Delta t_h = 10\Delta t_p$, $\Delta t_h = 100\Delta t_p$ or $\Delta t_h = 1000\Delta t_p$. Therefore, the CPU time of the simulation can be shortened. In order to demonstrate the idea of FDMFS more simply and clearly, the illustration of the proposed numerical procedures is shown as Fig. 3.

3.2 Manipulation in irregular domain

The present numerical scheme can solve problems with irregular domain directly. For solving particular solutions by FDM, we set a simple uniform Cartesian grid (Ω^C) which covers the whole problem domain (Ω) as a computational domain for particular solution. The Cartesian grid (Ω^C) can be divided into two parts. One (Ω_i^C) is the interior of the problem domain (Ω). The other one (Ω_e^C) is exterior of the problem domain (Ω).The particular solution at each point is governed by Eq. (6). However, we will encounter a problem that the forcing function on Ω_e^C is unknown in most practical problems. Therefore, we proposed three different schemes illustrated as Fig. 4 to overcome this issue. The accuracy and efficiency of these methods are compared with each other in section 4.2.

3.2.1 Method 1 (M1)

Provided that the forcing function of the problem is spatial-independent or the forcing function on Ω_e^C is known, the governing equation, Eq. (6), can be applied without problem. In other words, we use the same FDM to discretize the nonhomogeneous diffusion equation to obtain the particular solution on both Ω_e^C and Ω_i^C. The FDM scheme is the same as the FDM for a nonhomogeneous diffusion problem with a rectangular domain.

The idea of this method is simple and no extra computer code is required. This method has high accuracy due to no unreasonable assumptions. The virtual Cartesian grid for particular solution can be constructed automatically by known maximum and minimum value on each coordinate. This work will not affect the efficiency of the numerical scheme. However, this method is only suitable for the case which the forcing function is spatial-independent or the forcing function on Ω_e^C is known. Otherwise, we need to consider M2 or M3.

3.2.2 Method 2 (M2)

We can assume that this forcing function exterior the problem domain is equal to a constant, zero or a known physical value on the boundary. Therefore, the forcing function will appear a discontinuity near the boundary. Due to the nature of the FDM, the discontinuity will produce errors near the boundary and pollute the numerical results. Hence, we interpolate the forcing functions on the exterior domain (Ω_e^C) in order to smooth the forcing function on the Cartesian grid (Ω^C). If the forcing function on the computational domain for particular solution (Ω^C) is discontinuous, numerical error would be produced. To put it simply enough, the basic idea of M2 is to create a smooth forcing function to make sure that the numerical scheme for particular solution can be applied successfully.

In most practical problem, the problem domain is usually irregular and the forcing function outside the problem domain is unknown definitely. The M2 is developed to deal with this kind of problems. The unknown forcing function on Ω_e^C is set as a constant. The constant value can be defined through the system program and interpolations can be adopted to smooth the forcing function. This work needs to add computer code and set a rule to avoid discontinuous forcing function on the computational domain (Ω^C) for the particular solution. The scheme would increase the CPU time of the simulation. However, this approach conforms to the practice problem mostly.

3.2.3 Method 3 (M3)

In M3, we divide the numerical scheme into two parts. One is for nodes on the interior domain (Ω_i^C). The particular solution is obtained by the original governing equation, Eq. (6). The other one is for nodes on the exterior domain (Ω_e^C). The particular solution is equal to previous step without calculating on Ω_e^C. We do not care the particular solution on the non-real domain (Ω_e^C) in this method. However, some numerical error is caused by the inaccurate particular solution on the boundary. This method can speed up the numerical scheme due to not calculating the particular solution on the non-real domain (Ω_e^C).

After the brief introduction of the three methods, we can expect that the M1 has high accuracy; M2 is the most useful one; M3 can obtain the rough numerical results in a short time. The detailed comparisons of these three methods will be performed in the following section.

4 Numerical Experiments and Results

To illustrate the performance of the numerical scheme, we performed several numerical experiments listed in Tab. 1. There are six testing case studies in this paper. The former three cases are regular domain problems. In case 1, we considered a diffusion problem with a steady forcing function. Case 2 shows the convergence of this proposed numerical scheme and case 3 applies this method to simulate problems of k=10, 1 and 0.01. To show the advantages of meshless method, the last three cases are irregular domain problems. All irregular domain problems adopted these three different schemes mentioned on section 3.2 for particular solution and the detailed discussions of the three different methods are drawn based on the numerical results. The efficiency of FDM and FDMFS are compared with each other in problems of regular and irregular domains.

4.1 Regular domains

4.1.1 Case 1: Steady forcing function

In this case, the forcing function only depends on the spatial coordinates.

$$F(\vec{x},t) = -\frac{6x+6y+2}{12} \tag{20}$$

The analytical solution is

$$\begin{aligned} T(\vec{x},t) =& 2\left[\cos(\pi x)+\sin(\pi y)\right]e^{-k\pi^2 t} \\ &+\frac{x^3+y^3+x^2+y}{12}+10 \end{aligned} \tag{21}$$

Here, the diffusion coefficient k is equal to 1. Figure 5 (a) shows the time variations of the maximum and minimum temperature in the square domain with the explicit FDMFS (with 11 × 11 Cartesian grid for particular solution, 121 MFS nodes, time step size for particular solution $\Delta t_p = 10^{-3}$ and time step size for homogeneous solution $\Delta t_h = 10^{-2}$). This simple case can be solved by conventional FDM and another meshless method, MPSMFS [Young, Tsai and Fan (2004)]. Figure 5 (b) depicts the maximum relative errors of five different numerical methods. According to those results, all of those methods can obtain reasonable solutions. No matter explicit or fully implicit FDMFS scheme, the numerical solutions have high accuracy and the error distributions are very small. The efficiency of the five numerical methods is listed in Tab. 2. Explicit FDM can give numerical solution in a very short time; nevertheless, the MPSMFS spends a lot of time on solving full matrix. In addition, the meshless MPSMFS only can be used to solve PDEs with steady forcing function. Therefore it can not be applied for other unsteady-forcing-function cases. Although the meshless FDMFS also needs to solve the full matrix, the domain decomposition technique can overcome this issue. This simple case proves that the present numerical scheme can solve the nonhomogeneous diffusion problem successfully and accurately.

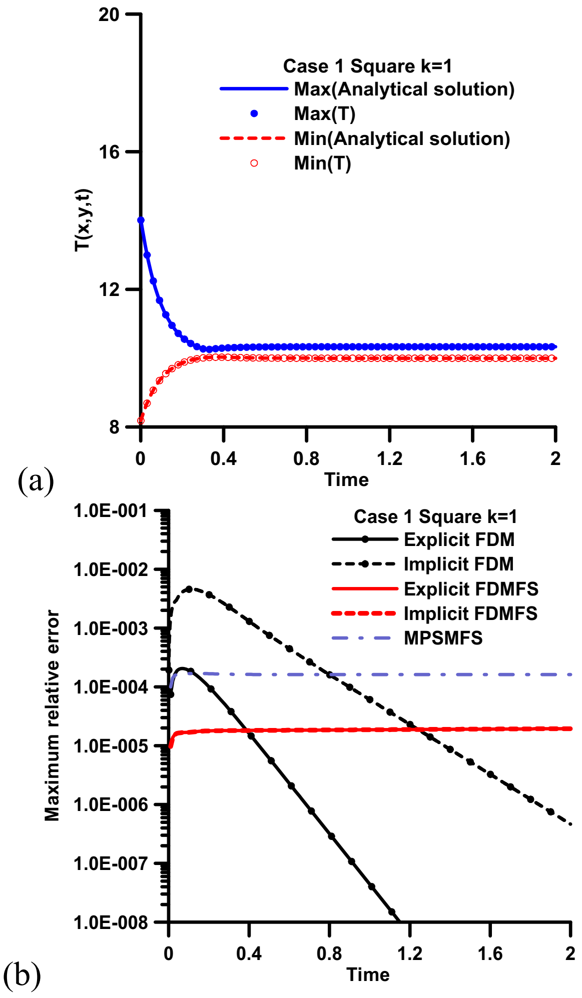


Figure 5: Time history of (a) maximum and minimum values of T and (b) maximum relative error for case 1

Table 1: All numerical experiments

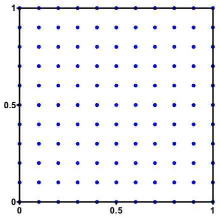
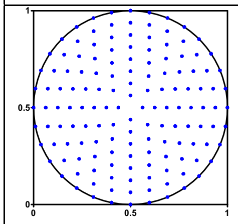
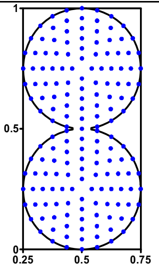
Domain	Case	Forcing function $F(\vec{x},t)$	Forcing function depends on	Issue
Regular domains				
	Case 1	$F(\vec{x},t)=-(6x+6y+2)/12$	x,y	* The efficiency of different numerical schemes
	Case 2	$F(\vec{x},t)=2\pi\ \mathrm{Cos}(2\pi\ t)$	t	* The convergence of FDMFS * Performance of the proposed scheme for problem with Neumann B.C.
	Case 3	$F(\vec{x},t)=2\pi(x^2-1)(y^2-1)\mathrm{Cos}\ (2\pi\ t)$ $-2k(x^2+y^2-2)\ \mathrm{Sin}(2\pi\ t)$	x,y,t	* Tests on different diffusion coefficient
Irregular domains				
	Case 4	$F(\vec{x},t)=2\pi\ \mathrm{Cos}(2\pi\ t)$	t	* Tests on the proposed suggestions (M1,M2,M3) for problems in irregular domain
	Case 5	$F(\vec{x},t)=2\pi(x^2-1)(y^2-1)\mathrm{Cos}\ (2\pi\ t)$ $-2k(x^2+y^2-2)\ \mathrm{Sin}(2\pi\ t)$	x,y,t	
	Case 6	$F(\vec{x},t)=2\pi(x^2-1)(y^2-1)\mathrm{Cos}\ (2\pi\ t)$ $-2k(x^2+y^2-2)\ \mathrm{Sin}(2\pi\ t)$	x,y,t	

Table 2: The comparisons for CPU time and required memory for different methods for case 1

Method	Node		Δt		CPU Time(sec)	Memory(KB)
Explicit FDM	N=11x11		$\Delta t=10^{-3}$		0.3	1480
Implicit FDM					1.5	1576
Explicit FDMFS	N_p=11x11	N_h=11x11	$\Delta t_p=10^{-3}$	$\Delta t_h=10^{-2}$	1.7	1756
Implicit FDMFS					2.5	1824
MPSMFS					8.3	2344

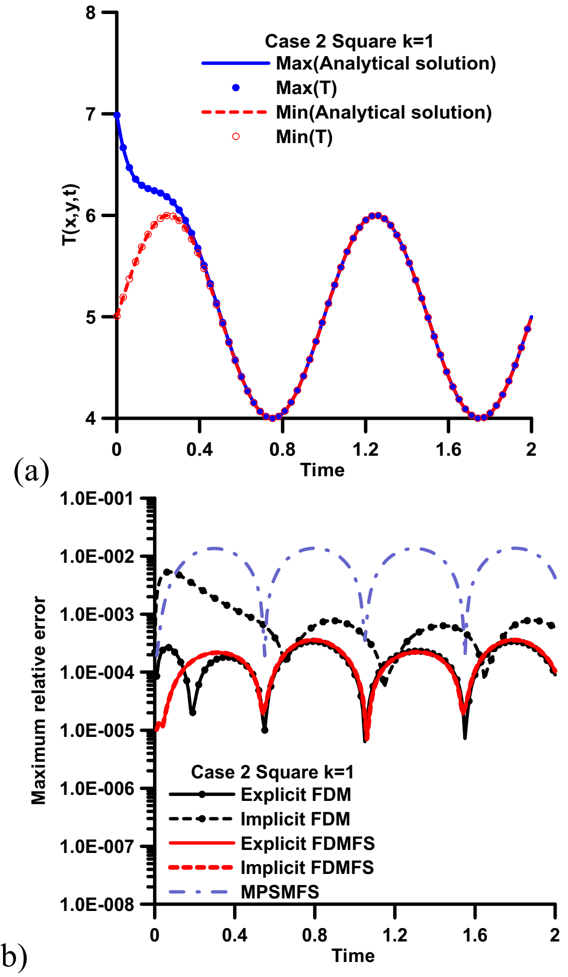


Figure 6: Time history of (a) maximum and minimum values of T and (b) maximum relative error for case 2

4.1.2 *Case 2: forcing function depends on t and Neumann boundary condition is considered*

We next consider the forcing function, which is an oscillation function and dependent on t, as follows:

$$F(\vec{x},t) = 2\pi\cos(2\pi t) \tag{22}$$

The analytical solution is

$$T(\vec{x},t) = (\sin(\pi x)+\sin(\pi y))\,e^{-k\pi^2 t}+\sin(2\pi t)+5 \tag{23}$$

Figure 6 (a) shows the maximum and minimum solution by explicit FDMFS (with 11 × 11 Carte-

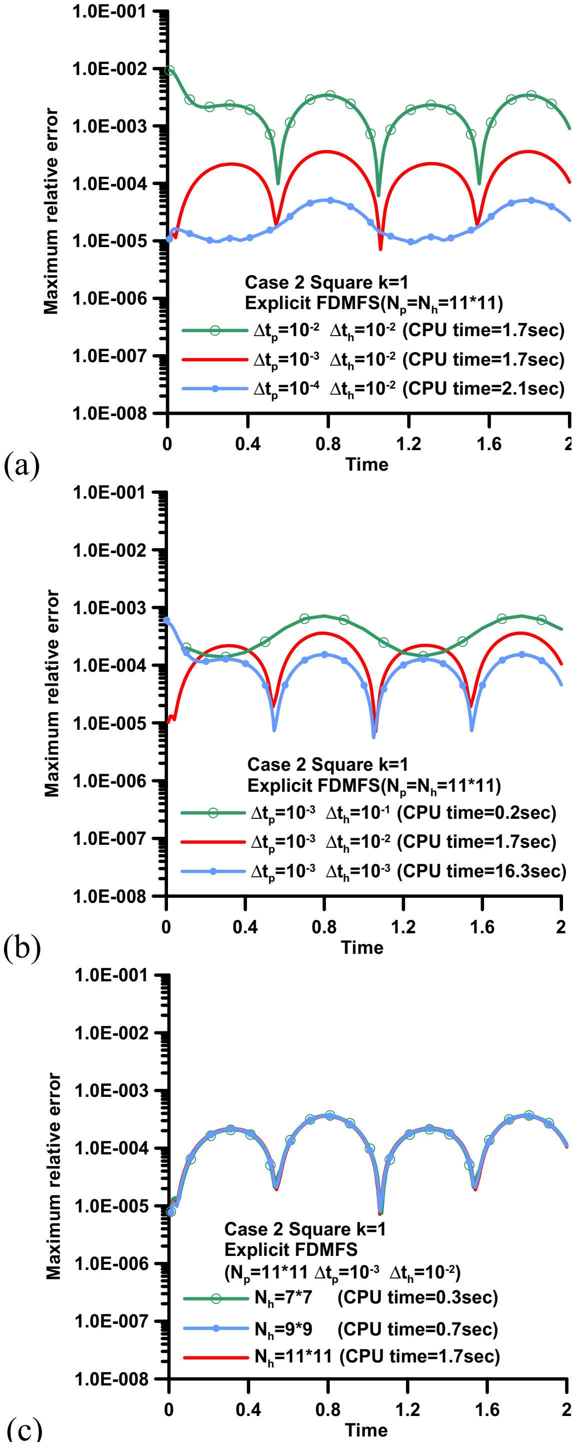


Figure 7: Time history of maximum relative error (a) with different time increments for particular solution (b) with different time increments for homogeneous solution (c) with different number of MFS nodes for case 2

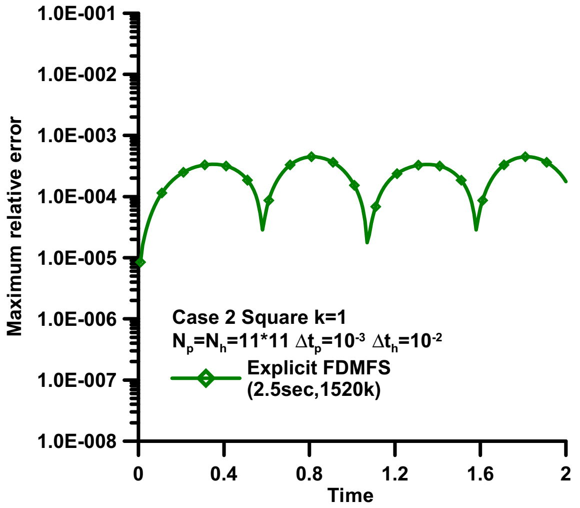
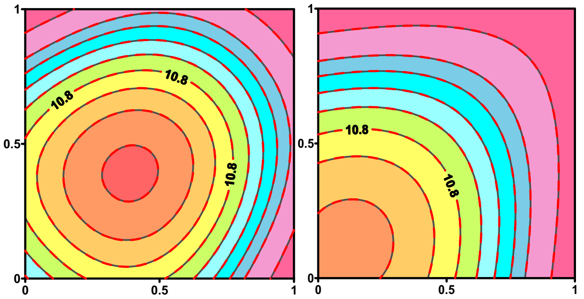


Figure 8: Time history of maximum relative error for problem with Neumann boundary condition at y=1 for case 2



(a) t=0.10 (b) t=0.25

Figure 9: Contour maps of temperature for case 3 for k=1

sian grid for particular solution, 121 MFS nodes, $\Delta t_p = 10^{-3}$ and $\Delta t_h = 10^{-2}$). The numerical solutions are in good agreement with analytical solutions. In Fig. 6 (b), the maximum relative errors of T by five different numerical schemes are plotted. The MPSMFS is applied to this problem by assuming the quasi-steady particular solutions. Therefore the numerical solutions of MPSMFS have a greater error shown in Fig. 6 (b). Yet the results using the FDMFS are quite good, with less than 0.03% relative error in both explicit and fully implicit schemes. In Tab. 3, we list the CPU time and memory cost of these simulations. As the table shown, conventional FDM has high efficiency in regular domain problem. The advantage of the FDMFS can be demonstrated in irregular domain problem, because it does not need the coordinate transformation which the conventional FDM needs.

The consistency and stability analysis of FDMFS is also included in this study. Figures 7 (a)-(c) depict the error histogram for different time increments and different numbers of points, in which smaller time increments and more points will give better results as expected. In Fig. 7 (a) we use fixed time increments for homogeneous solutions and change the time increments for particular solutions. As the figure shows, the smaller time increments induce better numerical solutions. We can obtain the same conclusions in Fig. 7 (b) with a changed time increments for homogeneous solutions and fixed the time increments for particular solutions. Lastly the Fig. 7 (c) shows that the number of MFS nodes will not influence the accuracy of the proposed scheme in the studying range.

A problem with Neumann boundary condition is also considered to test the proposed scheme. Let the boundary condition at y=1 is a Neumann boundary condition:

$$\frac{\partial T(x,t)}{\partial n} = -\pi e^{-k\pi^2 t}. \tag{24}$$

The relative error is shown in Fig. 8 in which the error is less than 0.05%.

4.1.3 Case 3: forcing function depends on x, y, t

In case 3, we consider the forcing function is dependent on x, y, and t. The analytical solution is shown as follows:

$$\begin{aligned} T(\vec{x},t) =& 10 + (\sin(\pi x) + \sin(\pi y))e^{-k\pi^2 t} \\ &+ (1+x)(1-x)(1+y)(1-y)\sin(2\pi t) \end{aligned} \tag{25}$$

and the forcing function is

$$\begin{aligned} F(\vec{x},t) =& 2\pi\left(x^2-1\right)\left(y^2-1\right)\cos(2\pi t) \\ &- 2k\left(x^2+y^2-2\right)\sin(2\pi t) \end{aligned} \tag{26}$$

The numerical results (black solid line) at t=0.1 and t= 0.25 shown in Fig. 9 agree well with the analytical solutions (red dashed line).

Table 3: The comparisons for CPU time and required memory for different methods for case 2

Method	Node		Δt		CPU Time(sec)	Memory(KB)
Explicit FDM	N=11x11		$\Delta t=10^{-3}$		0.6	1480
Implicit FDM					2	1576
Explicit FDMFS	N_p=11x11	N_h=11x11	$\Delta t_p=10^{-3}$	$\Delta t_h=10^{-2}$	1.7	1756
Implicit FDMFS					2.4	1824
MPSMFS					8.3	2344

We use this case to test the proposed meshless scheme for problems with different diffusion coefficient. Figure 10 (a) shows the time history of maximum and minimum temperature by explicit FDMFS for k=10, 1, and 0.01. As we can see, the results generally are in good agreement with analytical solutions. When k=10, the phenomena of diffusion process change quickly, as a result a smaller time increment is needed. The maximum relative error shown in Fig. 10 (b) is less than 0.05%. This case shows the present scheme is easy to handle problems with different diffusion coefficients.

4.2 Irregular domains

To illustrate the advantages of the proposed meshless scheme, there are three cases tested with irregular domains. Cases 4 and 5 demonstrate the application of FDMFS for circular domains. A twin circle domain problem is adopted in case 6. Each case has comparisons of required CPU time between the FDM and the FDMFS. Because the conventional FDM can not directly solve irregular domain problem, we adopt the boundary-fitted coordinate (BFC) transformation method [Lee and Leap (1994)] for the coordinate transformation. Brief descriptions of BFC transformation that we used are given in Appendix.

4.2.1 Case 4: forcing function depends on t

In this case, we consider the problem of a circular domain and the same analytical solution as case 2. Figure 11 (a) plots the time history of numerical results at (x,y)=(0.5,0.5) by M1, M2 and M3 and the numerical results show good agreement with analytical solutions. The maximum relative errors of the numerical results by explicit and implicit FDMFS schemes with different methods for handling of irregular domain are shown in Fig. 11 (b). As we expect, the accuracy of these results is almost in the same order and M1 has the best numerical results due to the spatial-independent forcing function is considered. Table 4 lists the number of nodes and time increment in the test and shows the comparisons of CPU time and memory between FDM and FDMFS. As the table shows, the coordinate transformation process needs 30 seconds of CPU time and 16,120 KB for memory storage. On the contrary, the FDMFS scheme obtains the numerical results in a very short time and saves a lot of memory, especially for explicit FDMFS scheme.

4.2.2 Case 5: forcing function depends on x, y, t

The analytical solution of case 5 is the same as case 3. The forcing function is dependent on x, y and t. Figure 12 (a) plots the time history of numerical results at (x,y)=(0.5,0.5) by M1, M2 and M3 and shows good agreement with analytical solutions. Fig. 12 (b) shows that the maximum relative error of M1, M2 and M3 are all less than 0.1%. The contour maps are shown in Fig. 13 for k=1 at t=0.1 ant t=0.25. These three methods for handling the irregular domain problem can produce reasonable results for arbitrary domain problems. Moreover, the comparisons of CPU time and memory between FDM and FDMFS are also listed on Tab. 5. In this case, the FDMFS needs finer mesh for particular solutions, because using the Cartesian grid for particular solutions needs more nodes to fit the unsteady and space-dependent forcing function. On the other hand, since the governing equation has been transformed and the extra coefficients of each term have been stored, the FDM scheme consumes a lot memory not only in mapping process but also in computing process. In contrast, the FDMFS is implemented without mapping pro-

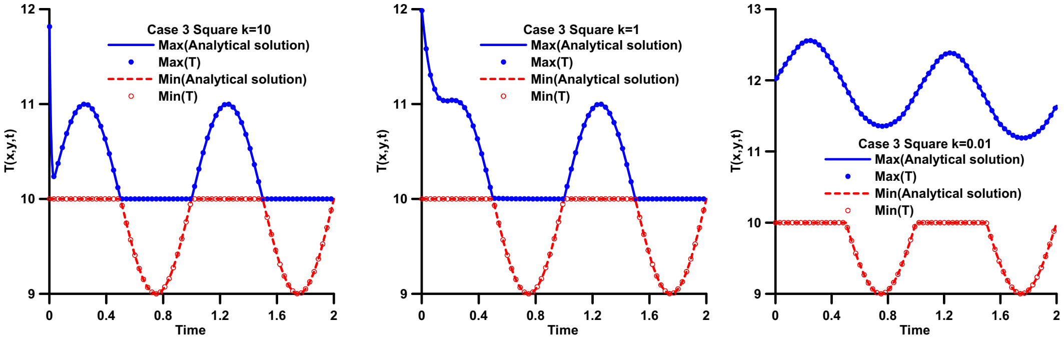


Figure 10(a): Time history of (a) maximum and minimum temperatures and (b) maximum relative error for case 3 when k=10, k=1 and k=0.01

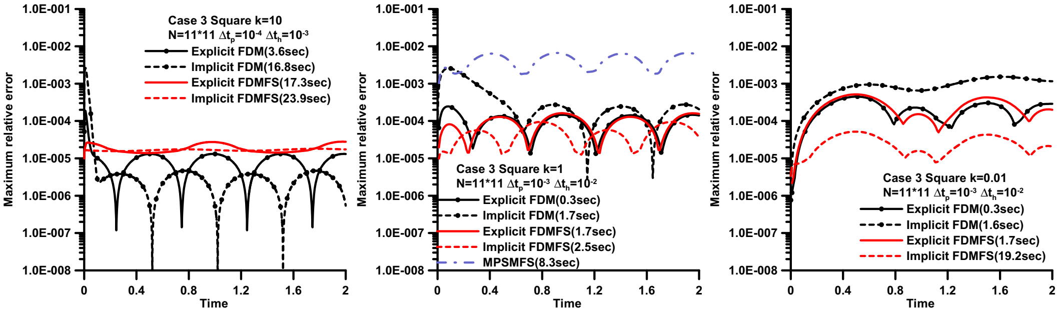
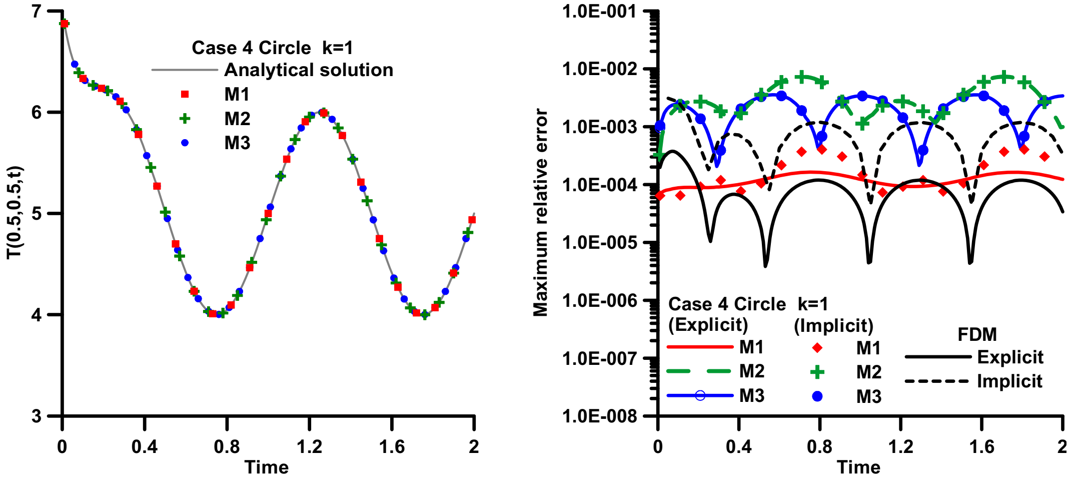
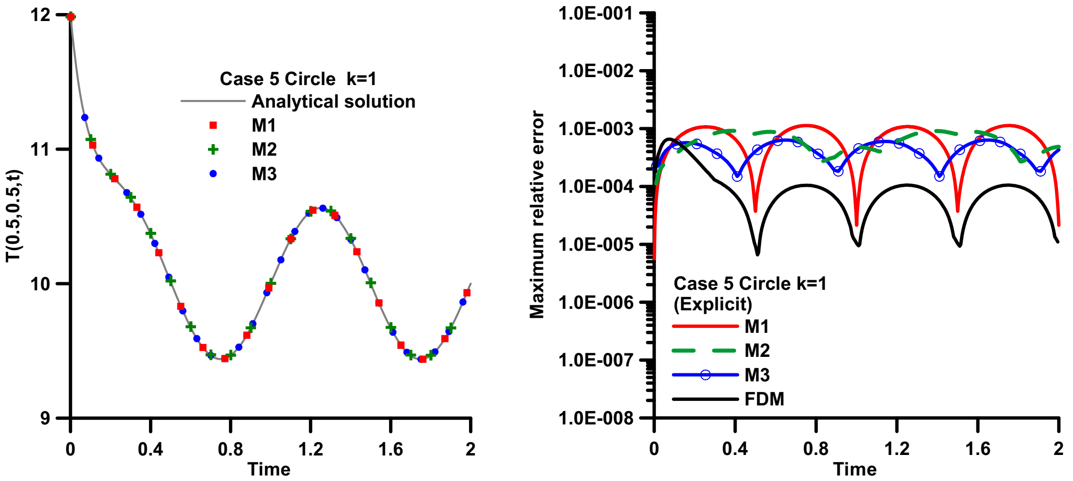


Figure 10(b): Time history of (a) maximum and minimum temperatures and (b) maximum relative error for case 3 when k=10, k=1 and k=0.01



(a) (b)

Figure 11: Time history of (a) temperature at (0.5, 0.5) and (b) maximum relative error for case 4 for k=1



(a) (b)

Figure 12: Time history of (a) temperature at (0.5, 0.5) and (b) maximum relative error for case 5 for k=1

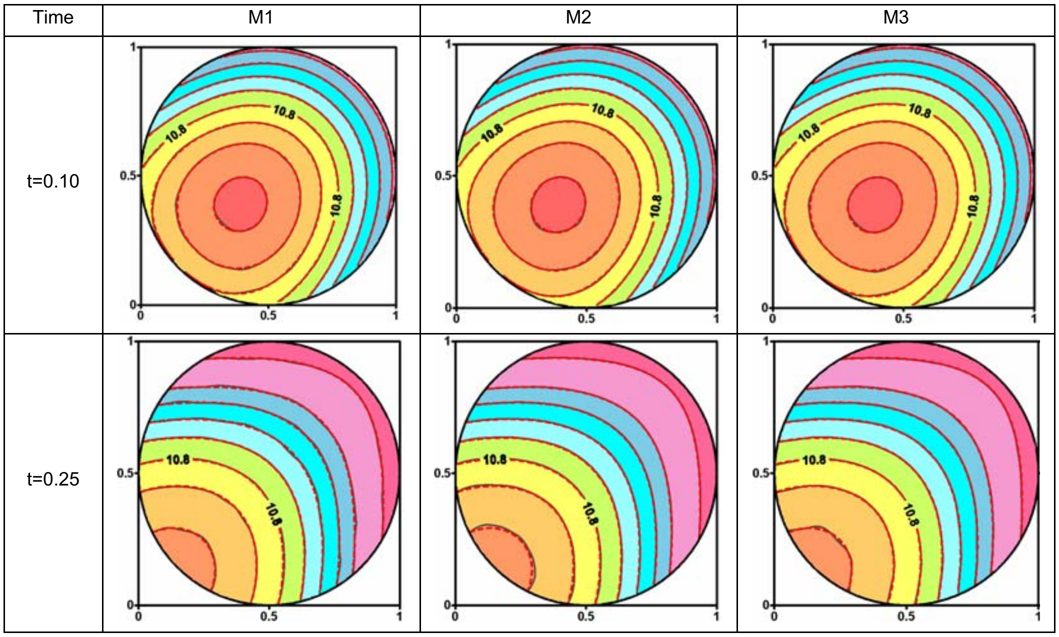


Figure 13: Contour maps of temperature for case 5

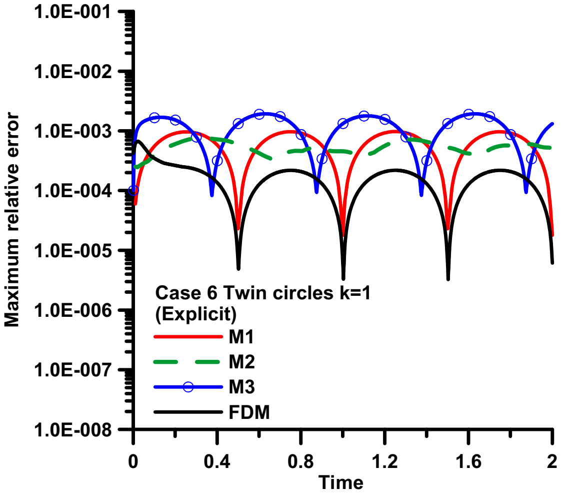


Figure 14: Time history of maximum relative error for case 6

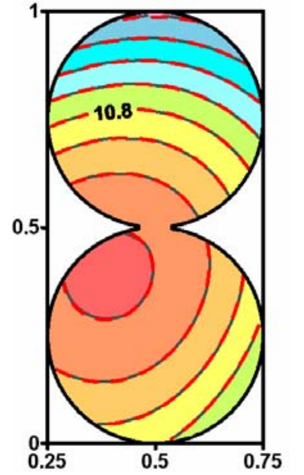
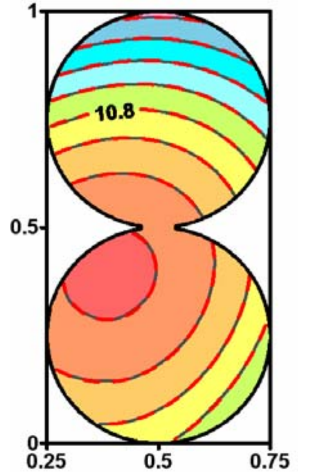
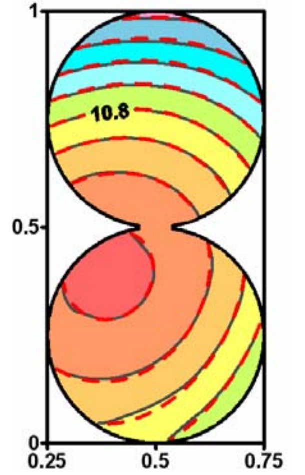
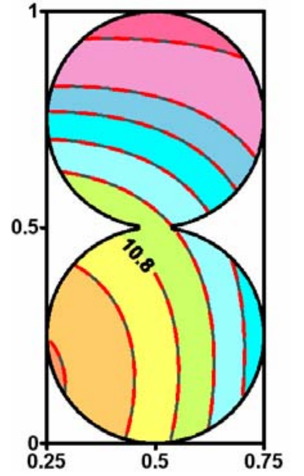
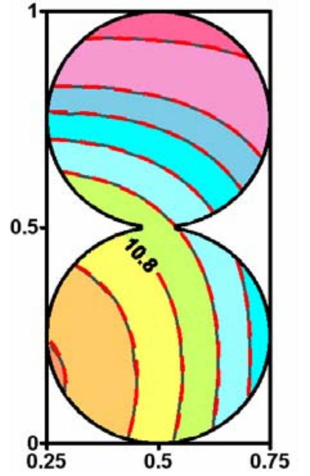
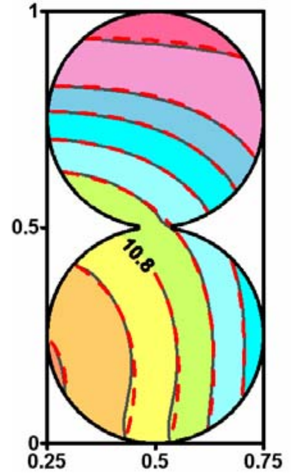
Time	M1	M2	M3
t=0.10			
t=0.25			

Figure 15: Contour maps of temperature for case 6 using different methods for particular solution

Table 4: The comparisons for CPU time and required memory for different methods for case 4

Method			Node		Δt		CPU Time(sec) total time (after mapping process)	Memory(KB) total cost (after mapping process)
Explicit FDM			N=31x31		$\Delta t=10^{-4}$		45.0 (14.3)	16120 (8952)
Implicit FDM					$\Delta t=10^{-3}$		52.3 (21.6)	16120 (9120)
FDMFS	Explicit	M1	N_p=41x41	N_h=144	$\Delta t_p=10^{-4}$	$\Delta t_h=10^{-2}$	7.7	1924
		M2					15.7	1924
		M3					7.3	1924
	Implicit	M1			$\Delta t_p=10^{-3}$		58.1	2148
		M2					73.5	2160
		M3					24.6	2132

Table 5: The comparisons for CPU time and required memory for different methods for case 5

Method			Node		Δt		CPU Time(sec) total time (after mapping process)	Memory(KB) total cost (after mapping process)
Explicit FDM			N=31x31		$\Delta t=10^{-5}$		173.1(142.33)	16148(9140)
FDMFS	Explicit	M1	N_p=81x81	N_h=144	$\Delta t_p=10^{-5}$	$\Delta t_h=10^{-2}$	370.5	2064
		M2					881.9	2064
		M3					274.6	2064

Table 6: The comparisons for CPU time and required memory for different methods for case 6

Method			Node		Δt		CPU Time(sec) total time (after mapping process)	Memory(KB) total cost (after mapping process)
Explicit FDM			N=21x41		$\Delta t=10^{-6}$		1307.3 (1278.6)	13308 (7548)
FDMFS	Explicit	M1	N_p=41x81	N_h=167	$\Delta t_p=10^{-5}$	$\Delta t_h=10^{-2}$	176.8	1936
		M2	Np=81x141				1531.5	2280
		M3	Np=21x41				68.6	2160

cess, so it can save memory significantly.

4.2.3 *Case 6: forcing function depends on x, y, t*

The last case is to simulate a twin circle domain problem. The analytical solution is chosen the same as case 3 and k=1. The maximum relative error is depicted in Fig. 14 and all of them are less than 0.2%. Moreover, Fig. 15 shows the contour maps via M1, M2 and M3 when t=0.10 and t=0.25. From this figure, we find the M3 has some error due to the discontinuity of the forcing function near the boundary. Furthermore, the comparisons of efficiency between FDM and FDMFS are provided in Tab. 6. In this case, the FDM based on BFC transformation uses a coarse mesh and a very small time step ($\Delta t = 10^{-6}$) such that the simulation costs more CPU time. In addition, the memory storage for FDM scheme is still large. On the other hand, the FDMFS based on M1 or M3 can obtain the numerical results in a very short time and require less memory. The FDMFS based on M2 is more time-consuming and it requires a finer Cartesian grid for particular solutions.

5 Conclusions

The nonhomogeneous diffusion equation with unsteady forcing function is analyzed by the proposed FDMFS which is the combination of the conventional FDM using simple grid and the meshless MFS. The solutions are assumed as the combination of particular solutions and homogeneous solutions. The FDM is applied to solve the particular solutions by a simple Cartesian grid which covers the whole physical domain. On the other hand, the homogeneous solution which is governed by the linear diffusion equation is solved by time-dependent MFS. Finally, the numerical solutions are obtained by summing the particular solutions and homogeneous solutions. The present numerical scheme can solve the nonhomogeneous diffusion equation with a time-dependent forcing function successfully. In addition, the boundary conditions for particular solution are not required and the initial conditions for particular solution can be assumed as an arbitrary function in the present scheme.

The proposed numerical scheme, FDMFS, is mainly aimed at diffusion problems with unsteady forcing function and in irregular domain. Since the forcing function is unknown outside the real domain, we addressed three approaches to deal with irregular domain problem for particular solution. From the numerical tests, it shows that M1 has the highest accuracy, M2 is the most reasonable method for the practical problems and M3 can get rough numerical solutions instantly. If the unsteady forcing function is spatial-independent, we recommend M1 for the particular solution. Otherwise, M2 is another good numerical scheme and commended to be applied.

In the present scheme, the particular solution over the whole domain is satisfied by the nonhomogeneous equation. We only need to construct a Cartesian grid covered the whole problem domain and the Cartesian grid for particular solution is very simple to generate. Besides, the MFS is free from mesh generation and numerical quadrature. Therefore the present scheme is very suitable and easy to analyze the diffusion problem with irregular domains. The reason we suggest the FDM to solve the particular solutions is that the scheme is very simple comparing to other discretized methods found in the literature. In addition, if we adopt the explicit FDM scheme, the matrix solver is not needed and numerical scheme can be speeded up. The numerical results are compared well with the analytical solutions. For future works, the FEM, differential quadrature (DQ) method also can be considered to analyze the particular solution in the proposed numerical scheme. The combination of the proposed FDMFS and Euler-Lagrangian method (ELM) to solve the nonhomogeneous advection-diffusion equation is expected and will be carried out in the near future.

Acknowledgement: The National Science Council of Taiwan is gratefully acknowledged for providing financial supports to carry out the present work under the grant No. NSC 95-2622-E-002-016-CC3 and No. NSC 95-2221-E-002-406. It is greatly appreciated. A partial financial support from CoreTech System is also acknowledged.

References

Alves, C.J.S.; Chen, C.S. (2005): A new method of fundamental solutions applied to nonhomogeneous elliptic problems. *Adv Comput Math*, vol. 23, pp. 125-142.

Balakrishnan, K.; Ramachandran, P.A. (2001): Osculatory interpolation in the method of fundamental solution for nonlinear Poisson problems. *J Comput Physics*, vol. 172, pp. 1-18.

Burgess, G.; Mahaherin, E. (1987): The fundamental collocation method applied to the nonlinear Poisson equation in 2 dimensions. *Comput Struct*, vol. 27, pp. 763-767.

Chantasiriwan, S. (2004): Cartesian grid methods using radial basis functions for solving Poisson, Helmholtz, and diffusion-convection equations. *Eng Anal Bound Elem*, vol. 28, pp. 1417-1425.

Chen, C.S. (1995): The method of fundamental solutions and the quasi-Monte Carlo method for Poisson's equation. In: H. Niederreiter and P. Shuie (ed) *Lecture Notes in Statistics106*, Springer, New York, pp. 158-167.

Chen, C.S.; Golberg, M.A.; Hon, Y.C. (1998a): The method of fundamental solutions and quasi-Monte Carlo method for diffusion equations. *Int J Numer Methods in Eng*, vol. 43, pp. 1421-1436.

Chen, C.S.; Golberg, M.A.; Hon, Y.C. (1998b): Numerical justification of the method of fundamental solutions and the quasi-Monte Carlo method for Poisson-type equations. *Eng Anal Bound Elem*, vol. 22, pp. 61-69.

Chen, C.W.; Fan, C.M.; Young, D.L.; Murugesan, K.; Tsai, C.C. (2005): Eigenanalysis for membranes with stringers using the methods of fundamental solutions and domain decomposition. *CMES: Computer Modeling in Engineering & Sciences*, vol. 8, pp. 29-44.

Chen, K.H.; Chen, J.T.; Kao, J.H. (2006): Regularized meshless method for solving acoustic eigenproblem with multiply-connected domain. *CMES: Computer Modeling in Engineering & Sciences*, vol. 16, pp. 27-39.

Cho, H.A.; Golberg, M.A.; Muleshkov, A.S.; Li, X. (2004): Trefftz methods for time dependent partial differential equations. *CMC: Computers Materials & Continua*, vol. 1, pp. 1-37.

Golberg, M.A. (1995): The method of fundamental solutions for Poisson's equation. *Eng Anal Bound Elem*, vol. 16, pp. 205-213.

Golberg, M.A.; Chen, C.S. (1998): The method of fundamental solutions for Potential Helmholtz and Diffusion Problems. *Boundary integral methods*, Computational Mechanics Publications, Boston, U.S.A., pp. 103-176.

Golberg, M.A.; Muleshkov, A.S.; Chen, C.S.; Cheng, A.H.D. (2003): Polynomial particular solutions for certain partial differential operators. *Numer Meth Part Differ Equ*, vol. 19, pp. 112-133.

Hon, Y.C.; Wei, T. (2005): The method of fundamental solutions for solving multidimensional inverse heat conduction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 119-132.

Hu, S.P.; Fan, C.M.; Chen, C.W.; Young, D.L. (2005): Method of fundamental solutions for Stokes' first and second problems. *J Mech*, vol. 21, pp. 25-31.

Kansa, E.J. (1990): Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics. 2. solutions to parabolic, hyperbolic and elliptic partial-differential equations. *Comput Math Appl*, vol. 19, pp. 147-161.

Kupradze, V.D.; Aleksidze, M.A. (1964): The method of functional equations for the approximate solution of certain boundary value problem. *USSR Comput Math Math Phys*, vol. 4, pp. 82-126.

Kythe P.K. (1996): Fundamental Solutions for Differential Operators and Applications, Birkhäuser, Boston.

Lee, K.K.; Leap, D.I. (1994): Application of boundary-fitted coordinate transformations to groundwater-flow modeling. *Transp Porous Media*, vol. 17, pp. 145-169.

Lin, H.; Atluri, S.N. (2000): Meshless local Petrov-Galerkin (MLPG) method for convection-diffusion problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 1, pp. 45-60.

Liu, Y.J.; Nishimura, N.; Yao, Z.H. (2005): A fast multipole accelerated method of fundamental solutions for potential problems. *Eng Anal Bound Elem*, vol. 29, pp. 1016-1024.

Mai-Cao, L.; Tran-Cong, T. (2005): A meshless IRBFN-based method for transient problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 149-171.

Sladek, V.; Sladek, J.; Tanaka, M. (2005): Local integral equations and two meshless polynomial interpolations with application to potential problems in non-homogeneous media. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 69-83.

Tsai, C.C.; Young, D.L.; Cheng, A.H.D. (2002): Meshless BEM for three-dimensional Stokes flows. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 117-128.

Wang, H.; Qin, Q.H.; Kang, Y.L. (2005): A new meshless method for steady-state heat conduction problems in anisotropic and inhomogeneous media. *Arch Appl Mech*, vol. 74, pp. 563-579.

Young, D.L.; Tsai, C.C.; Murugesan, K.; Fan, C.M.; Chen, C.W. (2004): Time-dependent fundamental solutions for homogeneous diffusion problems. *Eng Anal Bound Elem*, vol. 28, pp. 1463-1473.

Young, D.L.; Tsai, C.C.; Fan, C.M. (2004): Direct approach to solve nonhomogeneous diffusion problems using fundamental solutions and dual reciprocity methods. *J Chin Inst Eng*, vol. 27, pp. 597-609.

Young, D.L.; Ruan, J.W. (2005): Method of fundamental solutions for scattering problems of electromagnetic waves. *CMES: Computer Mod-*

eling in Engineering & Sciences, vol. 7, pp. 223-232.

Young, D.L.; Chen, C.W.; Fan, C.M.; Tsai, C.C. (2006): The method of fundamental solutions with eigenfunction expansion method for nonhomogeneous diffusion equation. *Numer Methods Partial Differ Equ*, vol. 22, pp. 1173-1196.

Young, D.L.; Chiu, C.L.; Fan, C.M.; Tsai, C.C.; Lin, Y.C. (2006): Method of fundamental solutions for multidimensional Stokes equations by the dual-potential formulation. *Eur J Mech B-Fluids*, vol. 25, pp. 877-893.

Young, D.L.; Fan, C.M.; Hu, S.P.; Atluri, S.N. (2007): The Eulerian-Lagrangian method of fundamental solutions for two-dimensional unsteady Burgers' equations. *Eng Anal Bound Elem* (doi:10.1016/j.camwa.2007.05.015).

Young, D.L.; Jane, S.J.; Fan, C.M.; Murugesan, K.; Tsai, C.C. (2006): The method of fundamental solutions for 2D and 3D Stokes problems. *J Comput Physics*, vol. 211, pp. 1-8.

Young, D.L.; Chen, K.H.; Chen, J.T.; Kao, J.H. (2007): A modified method of fundamental solutions with sources on the boundary for solving Laplace equations with circular and arbitrary domains. *CMES: Computer Modeling in Engineering and Sciences*, vol. 19, pp. 197-221.

Appendix FDM based on Boundary-Fitted Coordinate (BFC) Transformation for irregular domain

In this paper, the FDM solutions of irregular physical domain problem are solved based on the BFC transformation. In BFC system, the physical domain point (x,y) is correspondent with (X,Y) and $X=X(x,y), Y=Y(x,y)$. The BFC transformation generates the computational grid (X,Y) by solving the following Poisson equation:

$$X_{xx}+X_{yy}=P(X,Y) \tag{27}$$

$$Y_{xx}+Y_{yy}=Q(X,Y) \tag{28}$$

where P and Q are terms which control the point spacing on the interior of domain and can be assumed as zero. Eqs. (27)–(28) are then transformed to computational space by interchanging the roles of the independent and dependent variables. This yields a system of two elliptic equations of the form

$$\alpha x_{XX}-2\beta x_{XY}+\gamma x_{YY}=-J^2\left(Px_X+Qx_Y\right) \tag{29}$$

$$\alpha y_{XX}-2\beta y_{XY}+\gamma y_{YY}=-J^2\left(Py_X+Qy_Y\right) \tag{30}$$

where

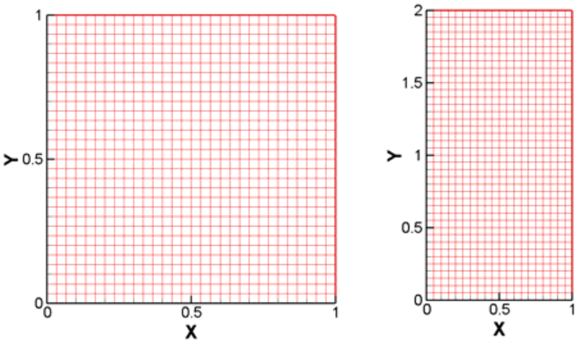
$$\alpha=x_Y^2+y_Y^2 \tag{31}$$

$$\beta=x_Xx_Y+y_Xy_Y \tag{32}$$

$$\gamma=x_X^2+y_X^2 \tag{33}$$

$$J=\frac{\partial(x,y)}{\partial(X,Y)}=x_Xy_Y-x_Yy_X \tag{34}$$

By solving this system equation, the relation between (x,y) in Cartesian coordinate and (X,Y) in BFC system is obtained. The structure grid (X,Y) as Fig. 16 (a)-(b) can be used in computational process instead of the physical mesh (x,y) as Fig. 17 (a)-(b) for numerical experiment case 4 to case 6.



(a) (b)

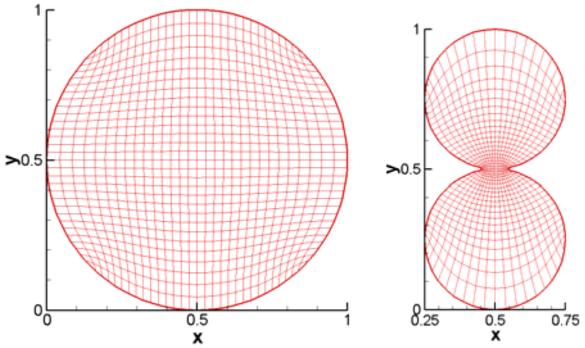
Figure 16: Computational domain and structure grid for (a) case 4 and 5 (b) case 6

Moreover, the governing equation, Eq. (1), is transferred as following:

$$\frac{\partial T}{\partial t}=k\left[\frac{1}{J^2}\left(\alpha T_{XX}-2\beta T_{XY}+\gamma T_{YY}\right)+pT_X+qT_Y\right]+F \tag{35}$$

where α, β, γ, J are defined in Eqs. (31)-(34), and

$$\begin{aligned} p=&-\frac{1}{J^3}y_Y\left(\alpha x_{XX}-2bx_{XY}+cx_{YY}\right)\\ &+\frac{1}{J^3}x_Y\left(\alpha y_{XX}-2by_{XY}+cy_{YY}\right) \end{aligned} \tag{36}$$



(a) (b)

Figure 17: Physical domain and generated grid for (a) case 4 and 5 (b) case 6

$$
\begin{aligned}
q =& \frac{1}{J^3} y_X \left(\alpha x_{XX} - 2b x_{XY} + c x_{YY}\right) \\
& - \frac{1}{J^3} x_X \left(\alpha y_{XX} - 2b y_{XY} + c y_{YY}\right)
\end{aligned} \tag{37}
$$

By discretizing the Eq. (35), the diffusion problem in irregular domain can be solved by FDM scheme.