Research Report

วิธีเมชเลสสำหรับการแก้ปัญหาสมการเบอร์เกอร์ แบบเชิงคู่โดยใช้วิธีมูฟวิงคริกกิง

Meshless Method for Solving Coupled Burgers Equation Based on Moving Kriging Interpolation Method

By
Kanittha Yimnak
Dhurakij Pundit University

This research project was granted by
Dhurakij Pundit University
Year 2016

Research Title : Meshless Method for Solving Coupled Burgers Equation Based on Moving Kriging Interpolation Method

Researcher : Miss Kanittha Yimnak **Department :** Mathematics and Statistics

Faculty : Applied Sciences **Year of Publication :** 2016

No. of Pages : 25 pages **Copyright :** Dhurakij Pundit University

ABSTRACT

In this research, The Meshless Local Petrov Galerkin formulation has been developed based on the moving Kriging interpolation method for solving coupled Burgers' equations in two dimensional spaces subjected to Dirichlet boundary conditions on a square domain with different values of Reynolds number (Re). The Crank-Nicloson method is chosen for the temporal discretization and the Kronecker delta function is used for the test function. Numerical results are compared with those of exact solutions and other available results for different values of Reynolds number. The results show that the developed formulation works well for this problem and has the accuracy of the estimation.

Keywords : Coupled Burgers' Equations, Moving Kriging Interpolation Method, The Meshless Local Petrov Galerkin

ชื่องานวิจัย : วิธีเมชเลสสำหรับการแก้ปัญหาสมการเบอร์เกอร์แบบเชิงคู่โดยใช้วิธีมูฟวิงคริกกิง

ผู้วิจัย : นางสาวกนิษฐา ยิ้มนาค **ภาควิชา** : คณิตศาสตร์และสถิติ

คณะ : วิทยาศาสตร์ประยุกต์ **ปีที่พิมพ์** : 2559

จำนวนหน้า : 25 หน้า

บทคัดย่อ

งานวิจัยครั้งนี้นำเสนอการพัฒนาวิธีเมชเลสโลคัลพีทรอฟ-กาเลอคิน โดยใช้ตัวประมาณแบบมูฟวิ่งคริกกิง เพื่อใช้ในการแก้ปัญหาสมการเบอร์เกอร์แบบคู่ใน 2 มิติภายใต้เงื่อนไขขอบเขตแบบดีรีเคล ในโดเมนที่มีลักษณะเป็นแบบสี่เหลี่ยม โดยจะศึกษาในกรณีที่ค่าเรย์โนลด์นัมเบอร์แตกต่างกันไป ในส่วนของการประมาณค่าแบบไม่ต่อเนื่องเชิงเวลา ในงานวิจัยนี้จะประยุกต์ใช้ วิธีแครงค์นิโคลสัน และใช้ฟังก์ชันทดสอบแบบฟังก์ชันเดลตาโครเนกเกอร์ ผลที่ได้จากการวิเคราะห์เชิงตัวเลขจากวิธีที่พัฒนาขึ้นดังกล่าวนี้ จะนำไปเปรียบเทียบกับผลเฉลยแม่นตรง(Exact Solution) และผลลัพธ์จากวิธีการแก้ปัญหาจากงานวิจัยอื่นๆ โดยจะศึกษาจากกรณีที่ค่าเรย์โนลด์นัมเบอร์ที่แตกต่างกันไป ผลการวิจัยพบว่าวิธีที่พัฒนาขึ้นทำงานได้ดีสำหรับแก้ปัญหาดังกล่าวและมีความถูกต้องของการประมาณค่า

คำสำคัญ : สมการการเบอร์เกอร์แบบคู่, ตัวประมาณมูฟวิ่งคริกกิง, วิธีเมชเลสโลคัลพีทรอฟ-กาเลอคิน,

ACKNOWLEDGEMENTS

This research has been completed because of suggestion from a state-of-the-art technical computing language and numerical algorithms by Asst. Prof. Dr. Anirut Luadsong. Moreover, I would like to thank Dhurakij Pundit University for the scholarship and their financial support.

Kanittha Yimnak

23 April 2016

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER1
INTRODUCTION

1.1 Rational

Burgers'equation, which proposed by Johannes Martinus Burgers (1895-1981), is a fundamental partial differential equation from fluid mechanics. It occurs in various areas of applied mathematics, such as modeling of gas dynamics and traffic flow. The developing of the numerical methods for solving this problem has been an interesting task for mathematicians. Generally, the system of nonlinear PDEs are solved by finite element (FE) or finite difference(FD) methods. However, the FEM or FDM have some limitations for example, the form for solving problem could be in strong form, and it fits nodes are arranged.

Meshless, or meshfree methods are proposed for solving this problem. Meshless, or meshfree methods, which overcome many of the limitations of the finite element method, have achieved significant progress in numerical computations of a wide range of engineering problems. A comprehensive introduction to meshless methods, meshless methods and Their Numerical Properties gives complete mathematical formulations for the most important and classical methods.

In this research, the meshless local Pretov-Galerkin (MLPG) method with the test function in view of the Kronecker delta function based on the moving Kriging approximation(MKA) method is proposed for solving the two-dimensional coupled nonlinear Burgers' equations[Srivastava, 2011] of the form,

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \end{aligned} \tag{1.1}$$

where $u(x, y, t)$ and $v(x, y, t)$ are velocity components to be determine and Re is the Reynolds number.

Constructing of shape functions is one of the most important issues in the MLPG method. Development of more effective methods for constructing shape functions have been one of the most active areas of research. There are many methods for constructing a shape function such as the moving least square (MLS) and the weighted least square (WLS) method. The most popular method is the moving least square. Although the MLPG method and many other meshless methods have been gradually applied to different fields, There exists an inconvenience because of the difficulty in implementing some essential boundary conditions; the shape function constructed by MLS approximation does not satisfy the Kronecker delta function property. Recently, we have tried to use the moving kriging approximation technique to construct meshless shape functions. The moving kriging approximation procedure originally employed in geostatistics by using known values and a semivariogram to determine unknown values. This mathematical model is name after Krige (Sack, 1989) who introduced the initial version of this spatial prediction process. The moving kriging approximation has two advantage; (1) the Kronecher delta property and (2) the consistency property.

These advantage enhance the accuracy of the estimation.two-dimensional coupled nonlinear Bugers' equations is proposed to be solved by the local integral equation formulation and one-step time discretization method by using the Crank-Nicolson methods. The boundary and domain integrals are calculated using Gauss-Legendre quadrature method. Two numerical examples are considered in order to verify the proposed method with testing its convergence and accuracy.

1.2 Literature Review

There are many researchers who developed the numerical methods for solving the two-dimensional Burgers' nonlinear differential equations.

Biazar (2009) proffered the variation iteration method(VIM) to solve the nonlinear Burgers' equations. This method is a powerful tool for solving a large number of problems. Using variational iteration method, it is possible to find the exact solution or a closed approximate solution of problem. Comparing the results with those of Adomain's decomposition and finite difference methods reveals significant points. To illustrate the ability and reliability of the method, some example are provided.

Zhu (2010) proposed the discrete Adomain decomposition method (ADM) for solving the two-dimensional Burgers' nonlinear differential equations. Two test problems are considered to illustrate the accuracy of the proposed discrete decomposition method. The numerical results are in good agreement with the exact solutions for each problem.

Abdul-Zahra (2012) presented an extension of exponential function method in rational form to find an exact solution of coupled Burgers'equation. This extended exponential function method in rational form allows us to find extra travelling wave solutions of coupled Burgers'equation instead of exponential function method in rational form.

Srivastava (2011) proposed scheme forms a system of nonlinear algebraic difference equations to be solved two-dimensional Burgers' equation at each time step. To linearize the non-linear system of equations, Newton's method is used. The obtained linear system is then solved by Gauss elimination with partial pivoting. The proposed scheme is unconditionally stable and second order accurate in both space and time. Numerical results are compared with those of exact solutions and other available results for different values of Reynolds number. The proposed method canbe easily implemented for solving nonlinear problems evolving in several branches of engineering and science.

1.3 Objective of Research

The purpose of this study is to develop a numerical algorithm using the Meshless Local Petrov-Galerkin (MLPG) method for solving the two-dimensional Burgers' nonlinear differential equations to increase accuracy.

1.4 Scope of Research

1. The equations are needed to be studied as the two-dimensional Burgers' nonlinear differential equations.

2. A procedure for the numerical solution is developed by the local integral equation to solve coupled nonlinear reaction-diffusion equations by using moving Kriging approximation.

1.5 Advantage of Research

The meshless method, which is developed, is a numerical procedure for solving coupled nonlinear reaction-diffusion equations and correspond to the boundary conditions problem.

CHAPTER2
THEORICAL AND BLACKGROUND

The theories, that related to this research, have been divided into 5 sections such as governing equation, moving kriging interpolation method, Gauss-Legendre quadrature Method , temporal discretization by the Crank-Nicolson method, and finally MLPG Upwinding Scheme I.

2.1 The Governing Equation

The two-dimensional coupled Burgers'equation are as following

$$\begin{aligned} &\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=0,\\ &\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}-\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)=0, \end{aligned} \tag{2.1}$$

subject to the initial conditions

$$u(x,y,0)=\psi_1(x,y);(x,y)\in\Omega,$$

$$v(x,y,0)=\psi_2(x,y);(x,y)\in\Omega,$$

and boundary conditions

$$u(x,y,t)=\xi_1(x,y);(x,y)\in\partial\Omega,\ t>0,$$

$$v(x,y,t)=\xi_2(x,y);(x,y)\in\partial\Omega, t>0,$$

where $\Omega=\{(x,y): a\le x\le b, c\le y\le d\}$ and $\partial\Omega$ is its boundary ; $u(x,y,t)$ and $v(x,y,t)$ are the velocity components to be determined, ψ_1, ψ_2, ξ_1 and ξ_2 are known functions and Re is the Reynolds number.

2.2 Moving Kriging Interpolation Method

The kriging interpolation is a well-known geostatic technique for spatial interpolation in geology and mining (Lei, 2003).The formulation of the construction of meshless shape function by moving kriging approximation (MKA) is introduced briefly in the following.Similar to the MLS approximation, Consider the function $u(\boldsymbol{x})$ defined in the domain Ω discretized by a set of properly scattered nodes $\boldsymbol{x}_i$, $(i=1,2,\dots,n)$, where n is the total number of nodes in the whole domain. It is assumed that only N nodes surrounding point $\boldsymbol{x}$ have the effect on $u(\boldsymbol{x})$.The sub-domain $\Omega_{\boldsymbol{x}}$ that encompasses these surrounding nodes is called the interpolation domain of point $\boldsymbol{x}$. The MKA $u^h(\boldsymbol{x})$ at point $\boldsymbol{x}$ is defined as presented in (Chen, 2011). Therefore the formulation of the meshless shape function using MKA is given by

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{N} \phi_I(\boldsymbol{x}) u_I = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{u} \qquad , \boldsymbol{x} \in \Omega_x \tag{2.2}$$

where $\boldsymbol{u} = [u(\boldsymbol{x}_1)\ \ u(\boldsymbol{x}_2) \cdots u(\boldsymbol{x}_N)]^T$ is a vector value of the function in the domain Ω. $\boldsymbol{\Phi}(\boldsymbol{x})$ is a $1 \times N$ vector of shape functions, expressed as:

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{A} + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{B}, \tag{2.3}$$

where matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as:

$$\begin{gathered} \boldsymbol{A} = (\boldsymbol{P}^T\boldsymbol{R}^{-1}\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{R}^{-1}, \\ \boldsymbol{B} = \boldsymbol{R}^{-1}(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{A}). \end{gathered} \tag{2.4}$$

In which $\boldsymbol{I}$ is a unit matrix of size $N{\times}N$, and vector $\boldsymbol{p}(\boldsymbol{x})$ is:

$$\boldsymbol{p}^T(x) = [p_1(\boldsymbol{x}_1) \quad \cdots \quad p_m(\boldsymbol{x}_N)]. \tag{2.5}$$

In general, a linear basis in two-dimensional space is:

$$\boldsymbol{p}^T(x) = (1, x, y), \ \ m = 3 \ ,$$

a quadratic basis is given as

$$\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2), \ \ m = 6,$$

and a cubic basis is

$$\boldsymbol{p}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3), \ \ m = 10.$$

For matrix $\boldsymbol{P}$ with the size $N{\times}m$, values of the polynomial basis functions (2.5) at the given set of nodes are collected:

$$\boldsymbol{P} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ \cdots & \cdots & \cdots \\ p_1(\boldsymbol{x}_N) & \cdots & p_m(\boldsymbol{x}_N) \end{bmatrix}. \tag{2.6}$$

Matrices $\boldsymbol{R}$ and vector $\boldsymbol{r}(\boldsymbol{x})$ are defined by the following equations:

$$\boldsymbol{R} = \begin{bmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \gamma(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \cdots & \cdots & \cdots \\ \gamma(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & \gamma(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix},$$

$$\boldsymbol{r}^T(\boldsymbol{x}) = [\gamma(\boldsymbol{x},\boldsymbol{x}_1) \ \cdots \ \gamma(\boldsymbol{x},\boldsymbol{x}_N)],$$

where $\gamma(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is the correlation function between any pair of nodes located at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, representing the covariance of the field value $u(\boldsymbol{x})$, i.e.

$$\gamma(\boldsymbol{x}_i,\boldsymbol{x}_j) = E[u(\boldsymbol{x}_i)\, u(\boldsymbol{x}_j)], \tag{2.7}$$

Similarly, the covariance $E[u(\boldsymbol{x}_i)\, u(\boldsymbol{x}_j)]$ can be replaced by $\gamma(\boldsymbol{x},\boldsymbol{x}_j)$. It can be seen from the foregoing formulations that the values of matrices $\boldsymbol{R}$ and $\boldsymbol{r}$ play important roles in the computation. A simple and frequently- used correlation function is a Gaussian function:

$$\gamma(\boldsymbol{x}_i,\boldsymbol{x}_j) = e^{-\theta r_{ij}^2}, \tag{2.8}$$

where $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and $\theta > 0$ are the correlation parameters used to fit the model and are assumed to be given.

The first-order partial derivatives of the shape function $\boldsymbol{\Phi}(\boldsymbol{x})$ against the coordinates $\boldsymbol{x}_i, i = 1{,}2$ can be easily obtained from Eq. (2.3)

$$\boldsymbol{\Phi}_{,i}(\boldsymbol{x}) = \boldsymbol{p}_{,i}^T(\boldsymbol{x})\boldsymbol{A} + \boldsymbol{r}_{,i}^T(\boldsymbol{x})\boldsymbol{B}, \tag{2.9}$$

where $(\cdot)_{,i}$ denotes $\partial(\cdot)/\partial x^i$.

The shape function obtained from the moving kriging approximation possesses the following delta function property:

$$\Phi_I(\boldsymbol{x}_J) = \delta_{IJ} = \begin{cases} 1, (I = J, I = 1,2,\ldots,N) \\ 0, (I \neq J, I, J = 1,2,\ldots,N) \end{cases} \tag{2.10}$$

The moving kriging approach is an exact interpolator, and its shape functions can exactly reproduce any function included in the basis. In particular, if all constants and linear terms are included, it reproduces a general linear polynomial exactly, that is,

$$\left.\begin{aligned} \sum_{I=1}^{N} \Phi_I(\boldsymbol{x}) &= 1 \\ \sum_{I=1}^{N} \Phi_I(\boldsymbol{x})\, x_I &= x \\ \sum_{I=1}^{N} \Phi_I(\boldsymbol{x})\, y_I &= y \end{aligned}\right\}. \tag{2.11}$$

2.3 Gauss-Legendre Quadrature Method

For 1-D, Let x_i be nodes and w_i be weights. The quadrature techniques formulation is following as (Abbott, 2005):

$$I_1 = \int_a^b f(x)dx, \tag{2.12}$$

where $f(x)$ be a polynomial of order 2n-1. Let [a,b] be [-1,1] can be accomplished by scaling.

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} g(\xi)d\xi \approx \sum_{i=1}^{n} w_i\, g_i \tag{2.13}$$

where ξ be transformation of variable x and $g(\xi)$ be transformation of variable x.

For 2-D, Let x_i and y_i, $i = 1,2, \dots, n$ are nodes and w_i and w_j are weights.

The quadrature techniques formulation is following as :

$$I_2 = \int_a^b f(x,y)d\Omega = \int_{-1}^{1}\int_{-1}^{1} g(\xi,\eta)d\,\xi d\eta \approx \int_{-1}^{1} I_1\, d\eta$$

$$= \int_{-1}^{1}\sum_{i=1}^{n_\xi} w_i\, g_i d\eta \approx \sum_{j=1}^{n_\eta} w_j \left(\sum_{i=1}^{n_\xi} w_i\, g_{ij}\right) = \sum_{j=1}^{n_\eta}\sum_{i=1}^{n_\xi} w_i w_j\, g_{ij},$$

where ξ_i, η_j are transformation of variables and g_{ij} is transformation function.

<u>**Calculating for weight**</u>

If we change the boundary condition form [-1,1]. We defined

$$x = C_0 + C_1\lambda,$$

where C_0, C_1 are unknown constants.

$$a = C_0 + C_1(-1),$$

$$b = C_0 + C_1(-1),$$

$$C_0 = \frac{\begin{vmatrix} a & -1 \\ b & 1 \end{vmatrix}}{\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}} = \frac{a+b}{1+1} = \frac{a+b}{2},$$

$$C_1 = \frac{\begin{vmatrix} 1 & a \\ 1 & b \end{vmatrix}}{\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}} = \frac{b-a}{1+1} = \frac{b-a}{2},$$

Hence

$$x = \left(\frac{a+b}{2}\right) + \left(\frac{b-a}{2}\right)\lambda\,, dx = \left(\frac{b-a}{2}\right) d\lambda$$

$$I = \int_a^b f(x)dx = \int_{-1}^{1} f(\lambda)\left(\frac{b-a}{2}\right) d\lambda \approx \left(\frac{b-a}{2}\right)\sum_{i=1}^{n} w_i\, f(\lambda_i)$$

For 2-D, we used Gauss-Legendre polynomial $(1, \lambda, \lambda^2, \lambda^3)$ then

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} 1 d\lambda = 2,$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda d\lambda = 0,$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda^2 d\lambda = \frac{2}{3},$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = \int_{-1}^{1} \lambda^3 d\lambda = 0.$$

Hence

$$w_1 f(1) + w_2 f(1) = 2,$$

$$w_1 f(\lambda_1) + w_2 f(\lambda_2) = 0,$$

$$w_1 f(\lambda_1^2) + w_2 f(\lambda_2^2) = \frac{2}{3},$$

$$w_1 f(\lambda_1^3) + w_2 f(\lambda_2^3) = 0.$$

The answers of equation (2.44) are $w_1 = w_2 = 1$, $\lambda_1 = -\frac{\sqrt{3}}{2}$ and $\lambda_2 = \frac{\sqrt{3}}{2}$. Some low-order rules for solving the integration problem are listed table1.

Table 2.1 abscissas and weights for Gaussian quadrature.

Number of points, n	Points, x_i	Weights, w_i
1	0	2
2	$\pm\frac{\sqrt{3}}{2}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\pm\sqrt{\frac{(3-2\sqrt{6/5})}{7}}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\frac{(3+2\sqrt{6/5})}{7}}$	$\frac{18-\sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

2.4 Temporal Discretization by The Crank–Nicolson method

The Crank–Nicolson method is based on the trapezoidal rule, giving second-order convergence in time. For example, in one dimension, if the partial differential equation is

$$\frac{\partial u}{\partial t} = F\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right), \tag{2.14}$$

then, letting $u(i\Delta x, n\Delta t) = u_i^n$, the equation for Crank–Nicolson method is a combination of the forward Euler method at n and the backward Euler method at $n + 1$ (note, however, that the method itself is not simply the average of those two methods, as the equation has an implicit dependence on the solution):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^n\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right),$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = F_i^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right),$$

Hence (Eq.(2.45)+Eq.(2.46))/2 reveals in Eq.(2.47),

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}\left[F_i^{n+1}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) + F_i^{n}\left(u, x, t, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right)\right], \tag{2.15}$$

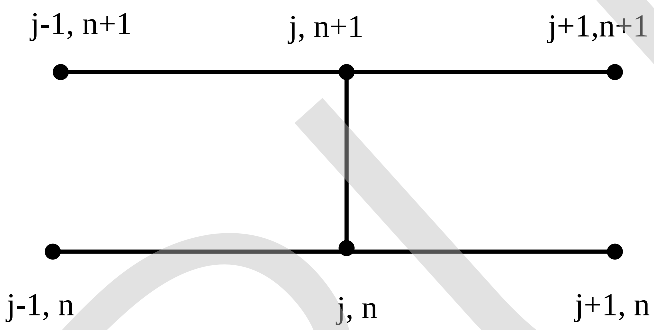


Figure2.1 The Crank–Nicolson stencil for a 1-D problem

2.5 MLPG Upwinding Scheme I

The MLPG method is based on the Petrov-Galerkin weighted residual procedures. Different spaces for the test and trial functions can be used, as shown in Figure 2.2 (Atluri, 2004)

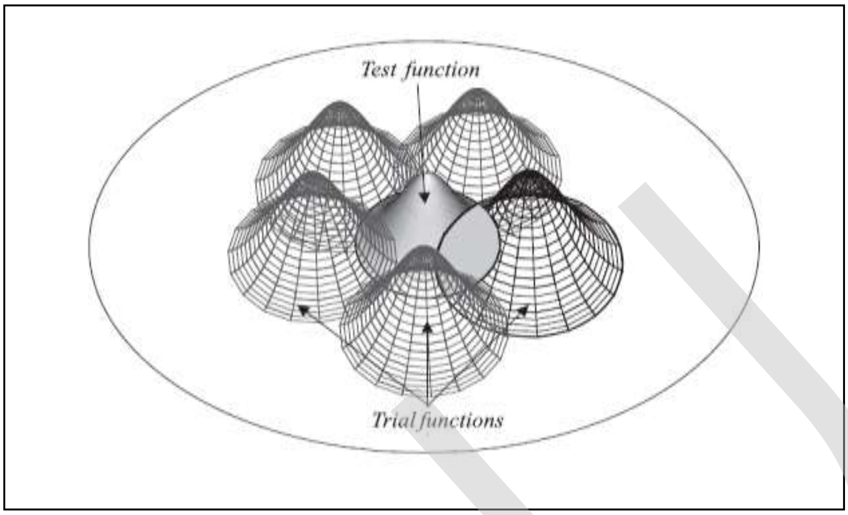


Figure 2.2 : The MLPG Method without Upwinding.

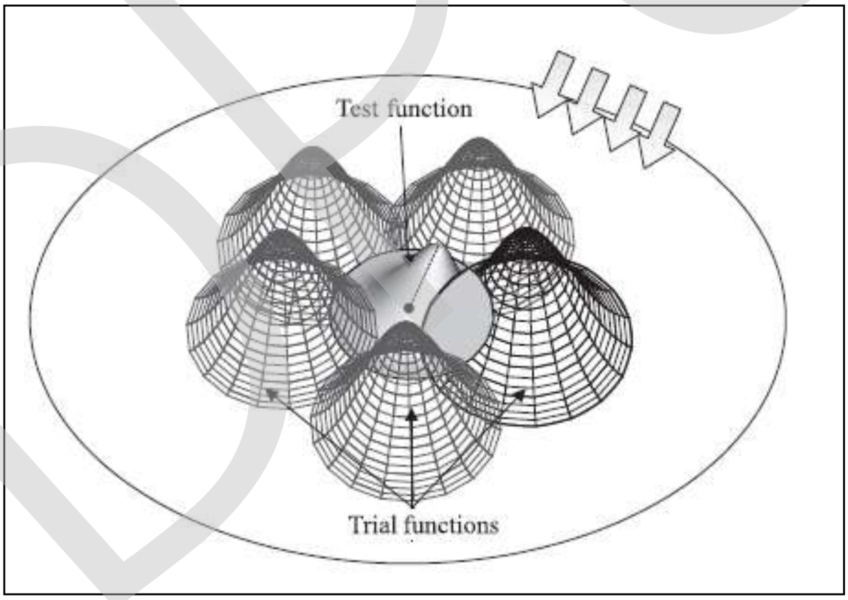


Figure 2.3 : MLPG Upwinding Scheme I (US-I).

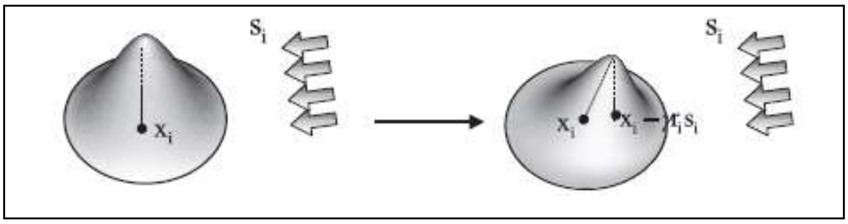


Figure 2.4 : MLPG Upwinding Scheme I (US-I): Specification.

Therefore one of the very natural ways to construct upwinding schemes is to choose different trial and test functions. This can be done by a lot of ways. For example, in order to apply upwinding in the streamline direction, we can skew the test function opposite to the streamline direction as shown in Figure 2.3. For convenience, we denote this as upwinding scheme I (US I). As an illustration, we choose a skewed weight function as the test function. The skewed weight function is given as follows: using the same form of weight function, we shift the position of the maximum of $w_i(\boldsymbol{x})$ from $\boldsymbol{x}_i$ to $\boldsymbol{x}_i$- $\gamma r_i s_i$, as shown in Figure 2.4, where, s_i is the unit vector of the streamline direction at $\boldsymbol{x}_i$, r_i is the size of the support for the test functions at $\boldsymbol{x}_i$, and γ is given by

$$\gamma = \frac{1}{2} coth\left(\frac{Pe}{2}\right) - \frac{1}{Pe} \tag{2.16}$$

in which Pe is a local Peclet number defined as:

$$Pe = \frac{ur_i}{k} \quad or \quad \frac{\text{advective transport rate}}{\text{diffusive transport rate}} \tag{2.17}$$

The size of the support for the trial functions also equal to r_i at $\boldsymbol{x}_i$, and the local sub-domain at $\boldsymbol{x}_i$ is coincided with the support for the test functions at $\boldsymbol{x}_i$.

CHAPTER3
METHODOLOGY

3.1 Space Discretization by MLPG Method with a Kronecher Delta Function (MLPG2)

The local integral formulation of Eq.(2.1) are as following

$$
\begin{aligned}
\int_{\Omega_s^i} \frac{\partial u}{\partial t} w_i dt &= \frac{1}{Re}\int_{\Omega_s^i} \nabla^2 u\, w_i dt - \int_{\Omega_s^i} u\frac{\partial u}{\partial x} w_i dt - \int_{\Omega_s^i} v\frac{\partial u}{\partial y} w_i dt,\\
\int_{\Omega_s^i} \frac{\partial v}{\partial t} w_i dt &= \frac{1}{Re}\int_{\Omega_s^i} \nabla^2 v\, w_i dt - \int_{\Omega_s^i} u\frac{\partial v}{\partial x} w_i dt - \int_{\Omega_s^i} v\frac{\partial v}{\partial y} w_i dt.
\end{aligned} \tag{3.1}
$$

Let $\tilde{u}(\boldsymbol{x},t)$ and $\tilde{v}(\boldsymbol{x},t)$, which substitute $u(\boldsymbol{x},t)$ and $v(\boldsymbol{x},t)$ respectively, be the trial solutions.

$$
\tilde{u}(\boldsymbol{x},t) = \sum_{j=1}^{N} \phi_j\,(x)\hat{u}_j(t),\quad \tilde{v}_j(\boldsymbol{x},t) = \sum_{j=1}^{N} \phi_j\,(\boldsymbol{x})\hat{v}_j(t),
$$

For internal nodes, from local integral equations (3.1), we have the following nonlinear equations:

$$
\begin{aligned}
\sum_{j=1}^{N}\int_{\Omega_s^i} \phi_j(\boldsymbol{x}) w d\Omega \frac{\partial \hat{u}_j}{\partial t} = &\sum_{j=1}^{N}\int_{\Omega_s^i} \frac{1}{Re}\Big(\phi_{j,xx}(\boldsymbol{x}) + \phi_{j,yy}(\boldsymbol{x})\Big) w d\Omega \hat{u}_j\\
&-\sum_{j=1}^{N}\int_{\Omega_s^i} \phi_{j,x}(\boldsymbol{x}) u(\boldsymbol{x})\, w d\Omega \hat{u}_j\\
&-\sum_{j=1}^{N}\int_{\Omega_s^i} \phi_{j,y}(\boldsymbol{x}) v(\boldsymbol{x})\, w d\Omega \hat{u}_{j.}
\end{aligned} \tag{3.2}
$$

Similarly, we have

$$
\begin{aligned}
\sum_{j=1}^{N}\int_{\Omega_s^i} \phi_j(\boldsymbol{x}) w d\Omega \frac{\partial \hat{v}_j}{\partial t} = &\sum_{j=1}^{N}\int_{\Omega_s^i} \frac{1}{Re}\Big(\phi_{j,xx}(\boldsymbol{x}) + \phi_{j,yy}(\boldsymbol{x})\Big) w d\Omega \hat{v}_j\\
&-\sum_{j=1}^{N}\int_{\Omega_s^i} \phi_{j,x}(\boldsymbol{x}) u(\boldsymbol{x})\, w d\Omega \hat{v}_j
\end{aligned} \tag{3.3}
$$

$$-\sum_{j=1}^{N}\int_{\Omega_s^i}\phi_{j,y}(\boldsymbol{x})v(\boldsymbol{x})\,wd\Omega\hat{v}_j,$$

where w_i is a Kronecher delta function used as the test function:

$$w_i(\boldsymbol{x})=\begin{cases}1\,, & \boldsymbol{x}=\boldsymbol{x_i}\\ 0\,, & \boldsymbol{x}\neq\boldsymbol{x_i},\end{cases}\tag{3.4}$$

Eqs.(3.2) and (3.3) can be written as:

$$\sum_{j=1}^{N}\phi_j(\boldsymbol{x_i})\frac{\partial\hat{u}_j}{\partial t}=\sum_{j=1}^{N}\frac{1}{Re}\left(\phi_{j,xx}(\boldsymbol{x_i})+\phi_{j,yy}(\boldsymbol{x_i})\right)\hat{u}_j-\sum_{j=1}^{N}\phi_{j,x}(\boldsymbol{x_i})u(\boldsymbol{x_i})\hat{u}_j$$
$$-\sum_{j=1}^{N}\phi_{j,y}(\boldsymbol{x_i})v(\boldsymbol{x_i})\hat{u}_j,\tag{3.5}$$

$$\sum_{j=1}^{N}\phi_j(\boldsymbol{x_i})\frac{\partial\hat{v}_j}{\partial t}=\sum_{j=1}^{N}\frac{1}{Re}\left(\phi_{j,xx}(\boldsymbol{x_i})+\phi_{j,yy}(\boldsymbol{x_i})\right)\hat{v}_j-\sum_{j=1}^{N}\phi_{j,x}(\boldsymbol{x_i})u(\boldsymbol{x_i})\hat{v}_j$$
$$-\sum_{j=1}^{N}\phi_{j,y}(\boldsymbol{x_i})v(\boldsymbol{x_i})\hat{v}_j,\tag{3.6}$$

Because of $u(\boldsymbol{x_i})=\hat{u}_i$ and $v(\boldsymbol{x_i})=\hat{v}_i$, Eqs.(3.5) and (3.6) are as the following

$$\sum_{j=1}^{N}\phi_j(\boldsymbol{x_i})\frac{\partial\hat{u}_j}{\partial t}=\sum_{j=1}^{N}\frac{1}{Re}\left(\phi_{j,xx}(\boldsymbol{x_i})+\phi_{j,yy}(\boldsymbol{x_i})\right)\hat{u}_j-\sum_{j=1}^{N}\phi_{j,x}(\boldsymbol{x_i})\hat{u}_i\hat{u}_j$$
$$-\sum_{j=1}^{N}\phi_{j,y}(\boldsymbol{x_i})\hat{v}_i\,\hat{u}_j,\tag{3.7}$$

$$\sum_{j=1}^{N}\phi_j(\boldsymbol{x_i})\frac{\partial\hat{v}_j}{\partial t}=\sum_{j=1}^{N}\frac{1}{Re}\left(\phi_{j,xx}(\boldsymbol{x_i})+\phi_{j,yy}(\boldsymbol{x_i})\right)\hat{v}_j-\sum_{j=1}^{N}\phi_{j,x}(\boldsymbol{x_i})\hat{u}_i\hat{v}_j$$
$$-\sum_{j=1}^{N}\phi_{j,y}(\boldsymbol{x_i})\hat{v}_i\,\hat{v}_j,\tag{3.8}$$

Eqs.(3.7) and (3.8) can be written as:

$$\sum_{j=1}^{N}\phi_j(\boldsymbol{x_i})\frac{\partial\hat{u}_j}{\partial t}=\sum_{j=1}^{N}\Big[(1/Re)\left(\phi_{j,xx}(\boldsymbol{x_i})+\phi_{j,yy}(\boldsymbol{x_i})\right)$$
$$-\phi_{j,x}(\boldsymbol{x_i})\hat{u}_i-\phi_{j,y}(\boldsymbol{x_i})\hat{v}_i\Big]\hat{u}_j,\tag{3.9}$$

$$\sum_{j=1}^{N} \phi_j(\boldsymbol{x_i}) \frac{\partial \hat{v}_j}{\partial t} = \sum_{j=1}^{N} \Big[(1/Re) \left(\phi_{j,xx}(\boldsymbol{x_i}) + \phi_{j,yy}(\boldsymbol{x_i}) \right) - \phi_{j,x}(\boldsymbol{x_i}) \hat{u}_i - \phi_{j,y}(\boldsymbol{x_i}) \hat{v}_i \Big] \hat{v}_j. \tag{3.10}$$

The following abbreviations have been used for the integral term:

$A_{ij} = \phi_j(\boldsymbol{x_i})$,

$B_{ij} = (1/Re) \left(\phi_{j,xx}(\boldsymbol{x_i}) + \phi_{j,yy}(\boldsymbol{x_i}) \right) - \phi_{j,x}(\boldsymbol{x_i}) \hat{u}_i - \phi_{j,y}(\boldsymbol{x_i}) \hat{v}_i$.

Eqs. (3.9) and (3.10) can be transformed into matrices form as:

$$\boldsymbol{A} \frac{\partial \boldsymbol{U}}{\partial t} = \boldsymbol{B}(\boldsymbol{U}, \boldsymbol{V}) \boldsymbol{U}, \tag{3.11}$$

$$\boldsymbol{A} \frac{\partial \boldsymbol{V}}{\partial t} = \boldsymbol{B}(\boldsymbol{U}, \boldsymbol{V}) \boldsymbol{V}, \tag{3.12}$$

where $\boldsymbol{A} = [A_{ij}]_{N \times N}$, $\boldsymbol{B} = [B_{ij}]_{N \times N}$, $\widehat{\boldsymbol{U}} = [\hat{u}_1 \quad \hat{u}_2 \quad \cdots \quad \hat{u}_N]'$ and $\widehat{\boldsymbol{V}} = [\hat{v}_1 \quad \hat{v}_2 \quad \cdots \quad \hat{v}_N]'$.

The finite-difference approximation of time derivatives of Eqs.(3.11) and (3.12) in the θ method is given as follows:

$$\frac{U^{k+1} - U^K}{\Delta t} = \theta B(U^{k+1}, V^{k+1}) U^{k+1} + (1-\theta) B(U^k, V^k) U^k, \tag{3.13}$$

$$\frac{V^{k+1} - V^K}{\Delta t} = \theta B(U^{k+1}, V^{k+1}) V^{k+1} + (1-\theta) B(U^k, V^k) V^k, \tag{3.14}$$

Eqs. (3.13) and (3.14) can be written as:

$$[I - \Delta t \theta B(U^{k+1}, V^{k+1})] U^{k+1} = [I + \Delta t (1-\theta) B(U^k, V^k)] U^k \tag{3.15}$$

$$[I - \Delta t \theta B(U^{k+1}, V^{k+1})] V^{k+1} = [I + \Delta t (1-\theta) B(U^k, V^k)] V^k \tag{3.16}$$

Because of $\boldsymbol{B}$ are nonlinear functions of U and V, we solve them iteratively in each time step with replacing $\boldsymbol{B}^{k+1}$ by $\boldsymbol{B}^k$, respectively, at zeroth iteration ($U^{k+1,0} = U^k, V^{k+1,0} = V^k$). Eqs. (3.15) and (3.16) are converted into a set of nonlinear algebraic equation for unknowns $\widehat{\boldsymbol{U}}^{k+1}$ and $\widehat{\boldsymbol{V}}^{k+1}$.

$$U^{k+1,l+1} = [I - \Delta t \theta B(U^{k+1,l}, V^{k+1,l})]^{-1} [I + \Delta t (1-\theta) B(U^k, V^k)] U^k \tag{3.17}$$

$$V^{k+1,l+1} = [I - \Delta t \theta B(U^{k+1,l}, V^{k+1,l})]^{-1} [I + \Delta t (1-\theta) B(U^k, V^k)] V^k \tag{3.18}$$

CHAPTER 4
NUMERICAL EXPERIMENTS

In this chapter, some numerical results are presented to verify this approach which compares to an exact solution and the previous research.

4.1 Example 1

In the following example, we consider the 2D Burgers' equations, with the initial conditions $u(x,y,0) = x + y$, $v(x,y,0) = x - y$ and the exact solutions are as follows (Bahadir, 2003):

$$u(x,y,t) = \frac{x + y - 2xt}{1 - 2t^2},$$
$$v(x,y,t) = \frac{x - y - 2yt}{1 - 2t^2}.$$

The computational domain has been taken as $\Omega = \{(x,y): 0 \leq x \leq 0.5, 0 \leq y \leq 0.5\}$. The results have been obtained using $Re = 1, N = 441$, $\Delta t = 0.0001$ at time instant $t = 0.1$ and $t = 0.4$, respectively. It is clear that the results from the present study are in good agreement with the exact solution (see in table 4.1-4.4). Perspective views of u and v are given in Fig.4.2. The exact and numerical solutions of u and v coincide (see in Figure 4.1-4.2). In addition, the proposed method achieves similar results given by (Zhu,2010).

4.2 Example 2

For the first example consider the following system of nonlinear PDEs in the region $\Omega = [0,1] \times [0,1]$. A solution of Bugers equation was given by Fletcher (Srivastava, 2011) using the Hopf-Cole as follows:

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4\left[1 + exp\left(\frac{(-4x + 4y - t)R}{32}\right)\right]},$$
$$v(x,y,t) = \frac{3}{4} + \frac{1}{4\left[1 + exp\left((-4x + 4y - t)R/32\right)\right]}.$$

The results have been obtain using $Re = 10, 80$ and $\Delta t = 0.0001$ at time instant t =0.05 and t =0.2, respectively. Table5 and Fig.4.4 show the results for $Re = 10$.
The numerical solutions are similar to the exact solution. For tables 4.6-4.7, the results for $Re = 80$ demonstrate that the developed method achieves similar results given by (Zhu, 2010). Perspective views of u and v for $Re = 80$ at $\Delta t = 0.0001$ are given in Figure 4.5. In case of $Re = 500$, $\Delta t = 0.01$ at time instant $t = 0.5$ with applying upwinding scheme I, the numerical solutions are similar to the exact solution and the previous research (Srivastava, 2011) (see in table 4.8 and Figure 4.6). However, the absolute errors of u and v in this research seem to be less than the previous research (see in Figure 4.3).

Table 4.1 Comparison among the exact solution, the previous research and numerical solution for u at $t = 0.1$.

(x, y)	u(Exact)	u(Zhu,2010)	Present work
(0.1,0.1)	0.18367	0.18368	0.18367
(0.3,0.1)	0.34694	0.34694	0.34694
(0.2,0.2)	0.36735	0.36735	0.36735
(0.4,0.2)	0.53061	0.53062	0.53061
(0.1,0.3)	0.38776	0.38776	0.38776
(0.3,0.3)	0.55102	0.55103	0.55102
(0.2,0.4)	0.57143	0.57144	0.57143
(0.3,0.4)	0.65306	0.65307	0.65306
(0.5,0.5)	0.91837	0.91838	0.91837

Table 4.2 Comparison among the exact solution, the previous research and numerical solutions for v at $t = 0.1$.

(x, y)	v(Exact)	v(Zhu,2010)	Present work
(0.1,0.1)	-0.02041	-0.02041	-0.02041
(0.3,0.1)	0.18367	0.18368	0.18367
(0.2,0.2)	-0.04082	-0.04082	-0.04082
(0.4,0.2)	0.16327	0.16327	0.16327
(0.1,0.3)	-0.26531	-0.26531	-0.26531
(0.3,0.3)	-0.06122	-0.06123	-0.06122
(0.2,0.4)	-0.28571	-0.28572	-0.28571
(0.3,0.4)	-0.18367	-0.18368	-0.18367
(0.5,0.5)	-0.10204	-0.10205	-0.10204

Table 4.3 Comparison among the exact solution, the previous research and numerical solutions for u at $t = 0.4$.

(x, y)	u(Exact)	u(Zhu,2010)	Present work
(0.1,0.1)	0.17647	0.17657	0.17647
(0.3,0.1)	0.23529	0.23585	0.23529
(0.2,0.2)	0.35294	0.35314	0.35294
(0.4,0.2)	0.41176	0.41242	0.41176
(0.1,0.3)	0.47059	0.47044	0.47059
(0.3,0.3)	0.52941	0.52972	0.52941
(0.2,0.4)	0.64706	0.64701	0.64706
(0.3,0.4)	0.67647	0.67665	0.67647
(0.5,0.5)	0.88235	0.88286	0.88235

Table 4.4 Comparison among the exact solution, the previous research and numerical solutions for v at $t = 0.4$.

(x, y)	v(Exact)	v(Zhu,2010)	Present work
(0.1,0.1)	-0.11765	-0.11729	-0.11765
(0.3,0.1)	0.17647	0.17657	0.17647
(0.2,0.2)	-0.23529	-0.23458	-0.23529
(0.4,0.2)	0.05882	0.05928	0.05882
(0.1,0.3)	-0.64706	-0.64574	-0.64706
(0.3,0.3)	-0.35294	-0.35188	-0.35294
(0.2,0.4)	-0.76471	-0.76303	-0.76471
(0.3,0.4)	-0.61765	-0.61610	-0.61765
(0.5,0.5)	-0.58824	-0.58646	-0.58824

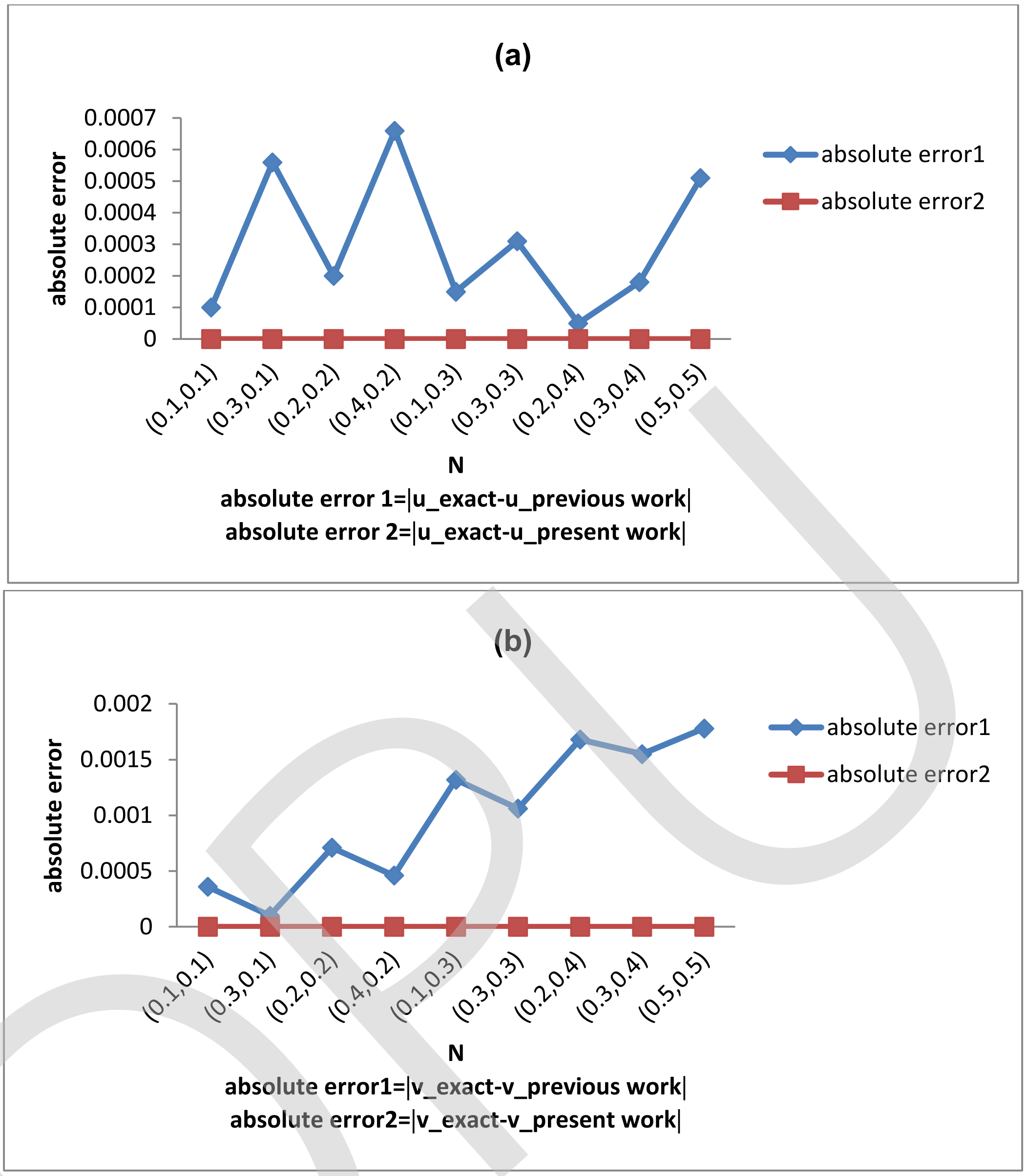
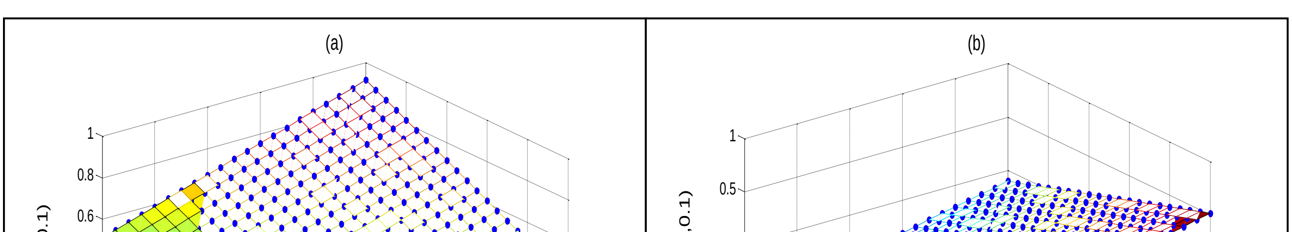


Figure 4.1: Absolute error of u and v, $\Delta t = 0.0001$ at time instant $t = 0.4$: (a) Absolute error of u ; (b) Absolute error of v.



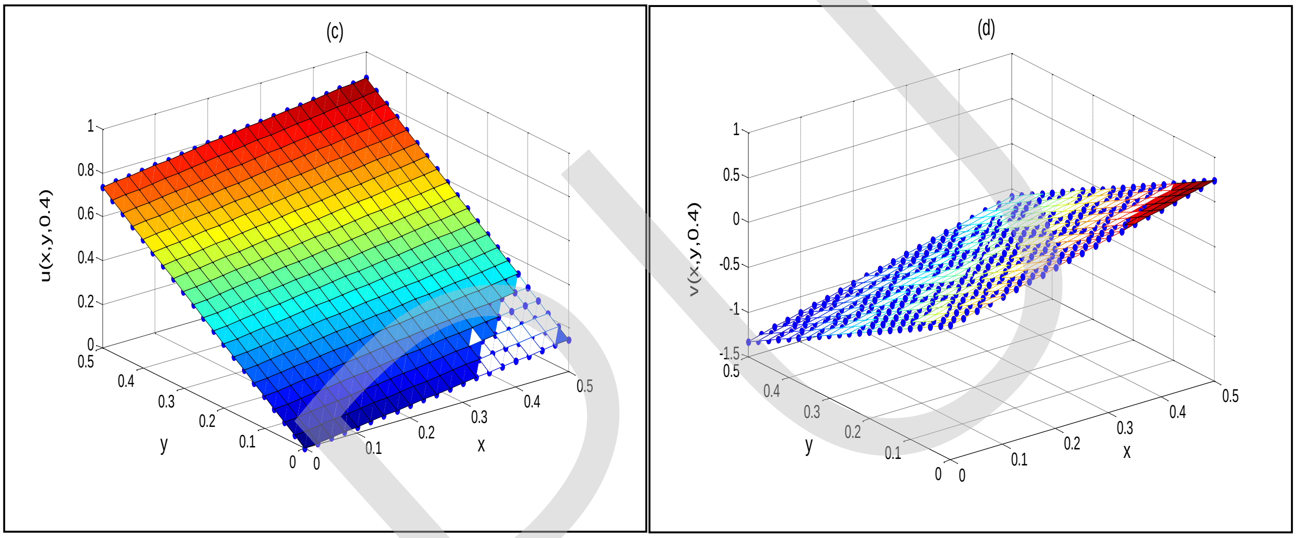


Figure 4.2 : A numerical illustration of approximation solutions of example2 by the developed method at $\Delta t = 0.0001$: (a) $u\ (x, y, 0.1)$; (b) $v\ (x, y, 0.1)$; (c) $u\ (x, y, 0.4)$; and (d)$v\ (x, y, 0.4)$.

Table 4.5 Comparison among the exact solution, the previous research and numerical solutions for u and v at $Re = 10$, $N = 441$, and $\Delta t = 0.0001$.

(x, y)	t = 0.05				t = 0.2			
	u (Exact)	u Present work	v (Exact)	v Present work	u (Exact)	u Present work	v (Exact)	v Present work
(0.1, 0.1)	0.61525	0.61525	0.88475	0.88475	0.58716	0.58716	0.91284	0.91284
(0.5, 0.1)	0.58540	0.58540	0.91460	0.91460	0.56127	0.56129	0.93873	0.93871
(0.9, 0.1)	0.55983	0.55984	0.94016	0.94016	0.54113	0.54111	0.95887	0.95889
(0.3, 0.3)	0.61525	0.61525	0.88475	0.88475	0.58716	0.58717	0.91284	0.91283
(0.7, 0.3)	0.58540	0.58540	0.91460	0.91460	0.56127	0.56128	0.93873	0.93872
(0.1, 0.5)	0.64628	0.64628	0.85372	0.85372	0.61720	0.61721	0.88280	0.88279
(0.5, 0.5)	0.61525	0.61525	0.88475	0.88476	0.58716	0.58717	0.91284	0.91283
(0.9, 0.5)	0.58540	0.58540	0.91460	0.91460	0.56127	0.56128	0.93873	0.93872
(0.3, 0.7)	0.64628	0.64628	0.85372	0.85372	0.61729	0.61720	0.88280	0.88280
(0.7, 0.7)	0.61525	0.61525	0.88475	0.88475	0.58716	0.58717	0.91284	0.91283
(0.1, 0.9)	0.67481	0.67481	0.82519	0.82519	0.64817	0.64816	0.85183	0.85184
(0.5, 0.9)	0.64628	0.64628	0.85372	0.85372	0.61720	0.61720	0.88280	0.88280
(0.9, 0.9)	0.61525	0.61525	0.88475	0.88475	0.58716	0.58717	0.91284	0.91283

Table 4.6 Comparison among the exact solution, the previous research and numerical solutions for u using $Re = 80$ and $\Delta t = 0.0001$.

(x, y)	t=0.05			t=0.2		
	Exact	u(Zhu,2010)	Present work	Exact	u(Zhu,2010)	Present work
(0.1, 0.1)	0.61720	0.61733	0.61709	0.59438	0.59465	0.59401
(0.9, 0.2)	0.50020	0.50020	0.50021	0.50014	0.50013	0.49987
(0.8, 0.3)	0.50148	0.50147	0.50151	0.50102	0.50098	0.50129
(0.7, 0.4)	0.51052	0.51046	0.51054	0.50733	0.50714	0.50736
(0.9, 0.5)	0.50398	0.50395	0.50411	0.50275	0.50266	0.50292
(0.1, 0.6)	0.74811	0.74810	0.74799	0.74725	0.74723	0.74467
(0.8, 0.6)	0.52667	0.52658	0.52673	0.51896	0.51867	0.51905
(0.3, 0.7)	0.74492	0.74491	0.74488	0.74267	0.74264	0.74254
(0.4, 0.7)	0.73665	0.73663	0.73659	0.73103	0.73102	0.73097
(0.2, 0.8)	0.74930	0.74930	0.74929	0.74898	0.74897	0.74815
(0.6, 0.8)	0.71676	0.71677	0.71666	0.70439	0.70457	0.70431
(0.1, 0.9)	0.74990	0.74990	0.75012	0.74986	0.74986	0.74978
(0.9, 0.9)	0.61720	0.61733	0.61727	0.59438	0.59465	0.59452

Table 4. 7 Comparison among the exact solution, the previous research and numerical solutions for v using $Re = 80$ and $\Delta t = 0.0001$.

(x, y)	t=0.05			t=0.2		
	Exact	u(Zhu,2010)	Present work	Exact	u(Zhu,2010)	Present work
(0.1, 0.1)	0.88280	0.88267	0.88291	0.90561	0.90534	0.90598
(0.9, 0.2)	0.99980	0.99980	0.99979	0.99986	0.99987	1.00013
(0.8, 0.3)	0.99852	0.99853	0.99849	0.99898	0.99902	0.99871
(0.7, 0.4)	0.98948	0.98954	0.98946	0.99267	0.99286	0.99264
(0.9, 0.5)	0.99602	0.99605	0.99589	0.99725	0.99734	0.99708
(0.1, 0.6)	0.75189	0.75190	0.75201	0.75275	0.75277	0.75533
(0.8, 0.6)	0.97333	0.97342	0.97327	0.98103	0.98133	0.98095
(0.3, 0.7)	0.75508	0.75509	0.75512	0.75733	0.75736	0.75746
(0.4, 0.7)	0.76335	0.76336	0.76341	0.76896	0.76898	0.76903
(0.2, 0.8)	0.75070	0.75070	0.75071	0.75102	0.75103	0.75185
(0.6, 0.8)	0.78324	0.78323	0.78334	0.79561	0.79543	0.79569
(0.1, 0.9)	0.75009	0.75009	0.74988	0.75014	0.75014	0.75022
(0.9, 0.9)	0.88280	0.88267	0.88273	0.90561	0.90534	0.90548

Table 4.8 Comparison among the exact solution, the previous research and numerical solutions for u and v using $Re = 500$ and $\Delta t = 0.01$ at t=0.5

(x, y)	t=0.5			t=0.5		
	Exact	u(Srivastava,2011)	Present work	Exact	v(Srivastava,2011)	Present work
(0.1, 0.1)	0.50010	0.48714	0.50238	0.99990	1.01286	0.99762
(0.5, 0.1)	0.50000	0.50002	0.50575	1.00000	0.99999	0.99425
(0.9, 0.1)	0.50000	0.50000	0.49726	1.00000	1.00000	1.00274
(0.3, 0.3)	0.50010	0.49519	0.50547	0.99990	1.00481	0.99453
(0.7, 0.3)	0.50000	0.50001	0.50530	1.00000	0.99999	0.99470
(0.1, 0.5)	0.75000	0.74990	0.74916	0.75000	0.75010	0.75000
(0.5, 0.5)	0.50010	0.49429	0.50470	0.99990	1.00571	0.99530
(0.9, 0.5)	0.50000	0.49978	0.50262	1.00000	1.00022	0.99738
(0.3, 0.7)	0.75000	0.75001	0.74782	0.75000	0.74999	0.75218
(0.7, 0.7)	0.50010	0.49325	0.50313	0.99990	1.00676	0.99687
(0.1, 0.9)	0.75000	0.75000	0.748550	0.75000	0.75000	0.75145
(0.5, 0.9)	0.75000	0.75001	0.74990	0.75000	0.74999	0.75011
(0.9, 0.9)	0.50010	0.47275	0.50471	0.99990	1.02725	0.99529

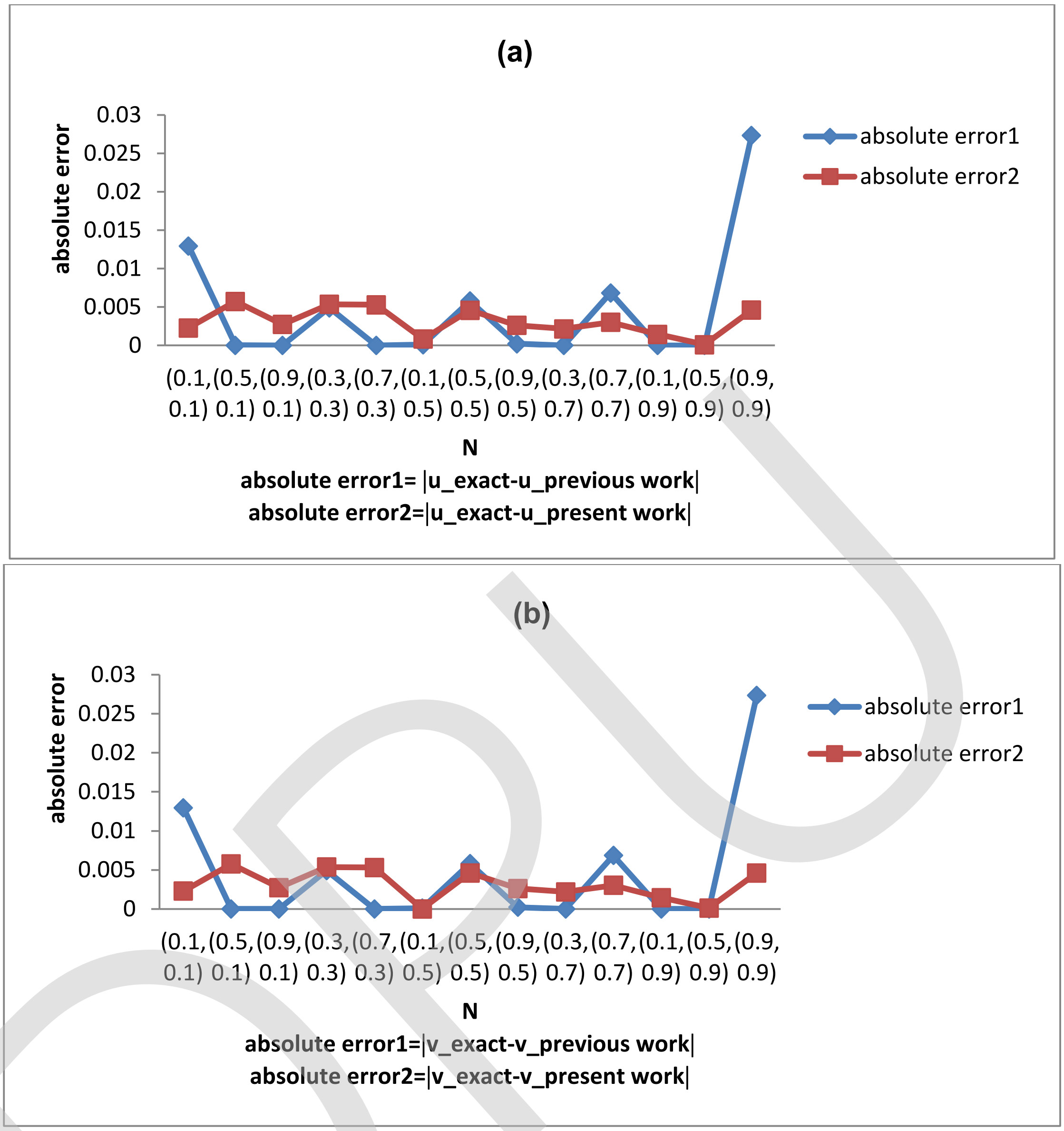


Figure 4.3 : Absolute error of u and v $\Delta t = 0.01$ at time instant $t = 0.5$:
(a) Absolute error of u ; (b) Absolute error of v.

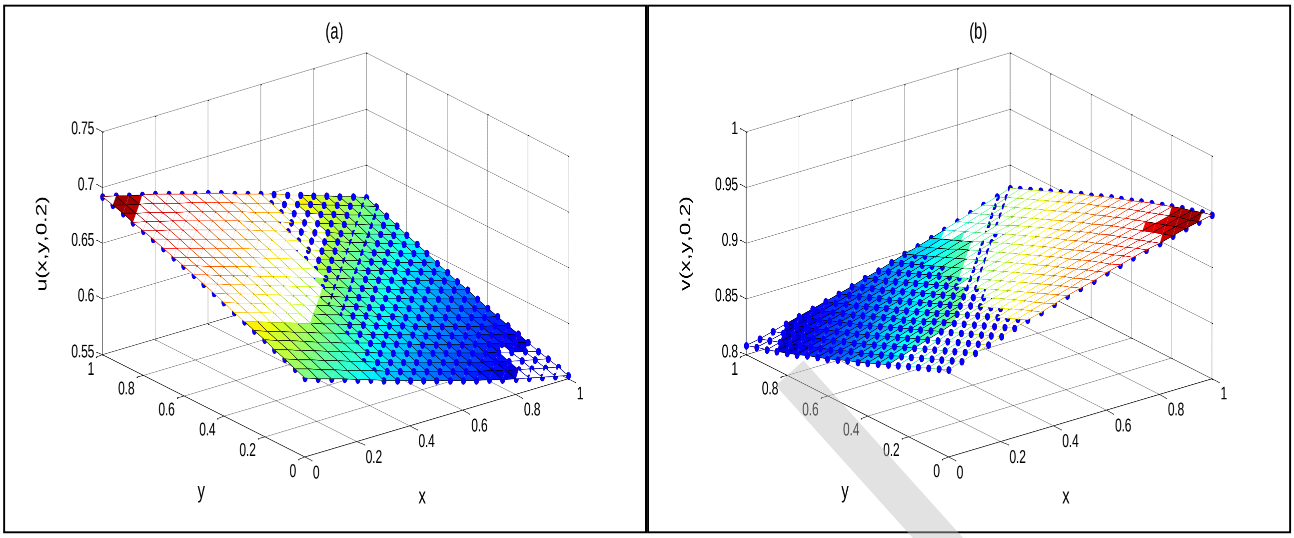


Figure 4.4 : A numerical illustration of approximation solutions of example1 by the developed method at $Re = 10, \Delta t = 0.0001$: (a) $u\ (x, y, 0.2)$; and (b)$v\ (x, y, 0.2)$.

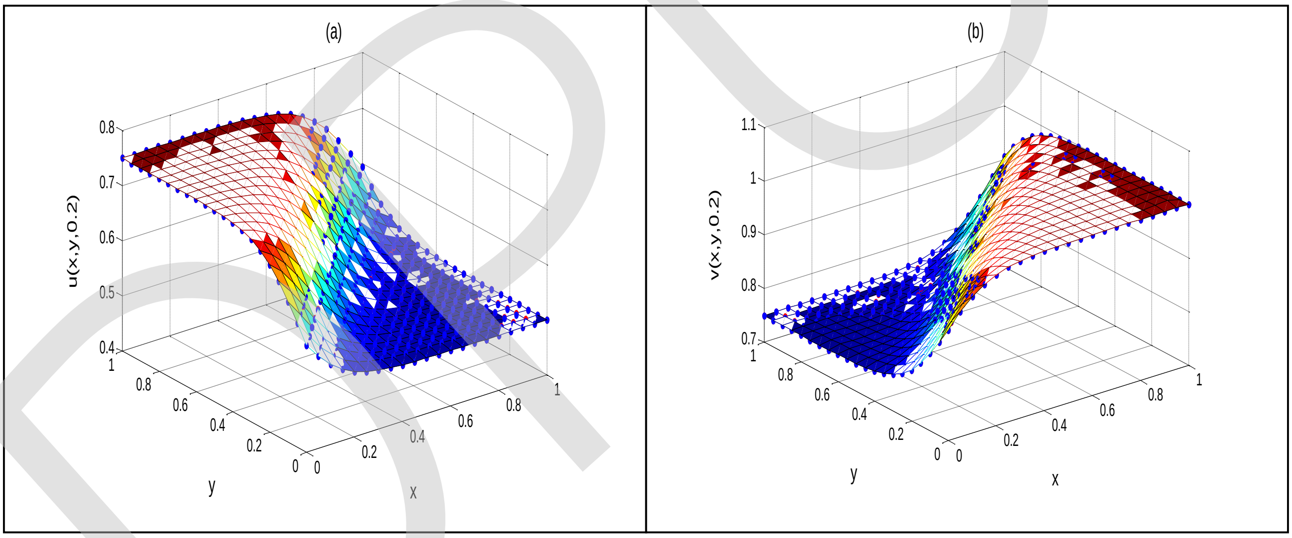


Figure 4.5 : A numerical illustration of approximation solutions of example1 by the developed method at $Re = 80, \Delta t = 0.0001$: (a) $u\ (x, y, 0.2)$; and (b)$v\ (x, y, 0.2)$.

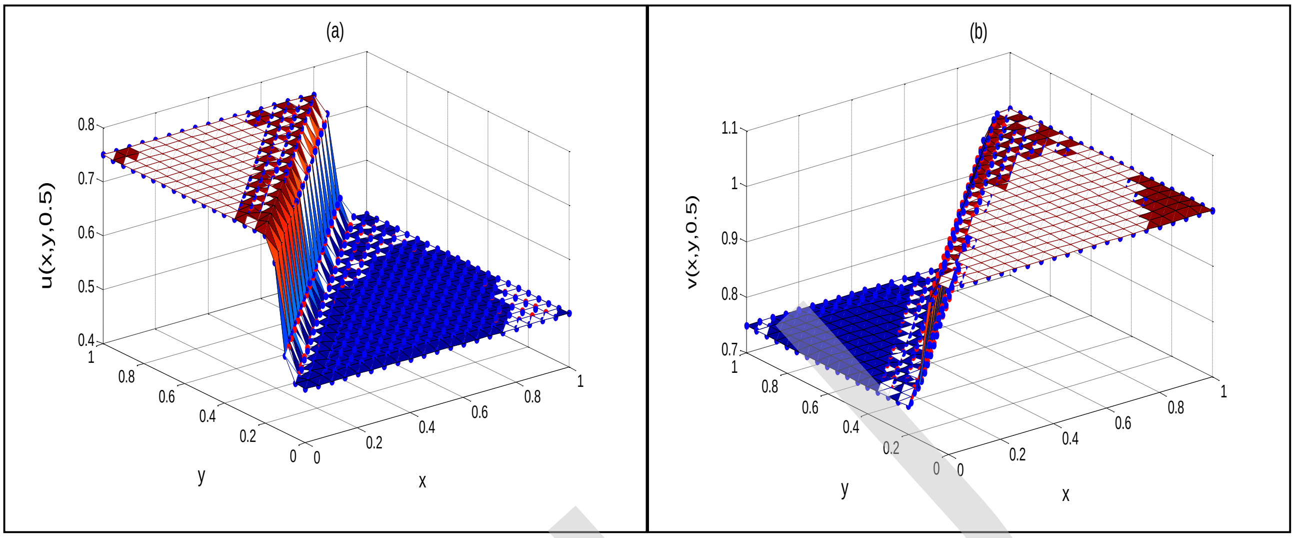


Figure 4.6 : A numerical illustration of approximation solutions of example1 by the developed method at $Re = 500, \Delta t = 0.01$: (a) $u\ (x, y, 0.5)$; (b) $v\ (x, y, 0.5)$.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The developed MLPG method based on MKA method had been successfully applied to solve the solutions of coupled Burger system. The efficiency and accuracy of the developed method was demonstrated by two test problems.

For problem1, it obviously that there are no difference between the exact and the trial solutions. For problem2, in case of $Re = 10$ and 80, the numerical results show that the developed method is reliable to solve the problem. In addition, the values of u and v obtained from the numerical solution close to the exact solution and the previous research.

In case of $Re = 500$, the developed method with upwinding scheme works well. In addition, the results of the developed method are closer to the exact solution and the previous serearch.

The developed MLPG method is more efficient than the previous methods. The reasons for this is the evidence from the errors of u and v by the developed method seem to be less than the previous method (see in Figures 4.1 and 4.3).

5.2 Recommendations

In future works, the MLPG formulations will be developed for solving coupled Burgers' equations in 3-dimensional spaces which satisfy physics and engineering problems.

REFERENCES

Abdul-Zahra, K. A.. "Extended exponential function method in rational form for exact solution of coupled Burgers equation". Journal of Basrah Researches (Sciences). 38, 1. A(2012) : 72-78

Atluri, S.N. TheMeshless Method(MLPG) for Domain and Bie Discretizations. Tech Science Press. (2004).

Bahadir, R. A fully implicit finite-difference scheme for two dimensional Burgers' equations. Journal Applied Mathematics and Computation. 137(2003) : 131-137.

Biazar, J., Aminikhah, H. Exact and Numerical Solutions for Non-Linear Burger's Equation by VIM. Mathematical and Computer Model. 49(2009) : 1394-1400.

Chen, L. and Liew, K.M. A Local Petrov-Galerkin Approach with Moving Kriging Interpolation for Solving Transient Heat Conduction Problems. Computational Mechanics. 47(2011) : 455-467.

Lei, G., Moving Kriging Interpolation and Element-Free Galerkin Method, International Journal for Numerical Methods in Engineering. 56(2003) : 594–602.

Ozis, T., Aslan, Y. The semi-approximate approach for solving Burgers' equation with high Reynolds number. Applied Mathematics and Computation. 163(2005) : 131-145.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn H.P. Design and Analysis of Computer Experiments. Statistical Science. 4(1989): 409-435.

Srivastava, V.K., Tamsir, M., Bhardwaj, U. and Sanyasiraju. Crank-Nicolson Sheme for Numerical Solutions of Two-dimensional Coupled Bugers' Equations. International Journal of Scientific & Engineering Research. 2(2011): 1-6.

Zhu, H., Shu, H., Ding, M. Numerical solutions of two-dimensional Burgers' equations by discrete Adomain decomposition method. Computers and Mathematics with Applications. 60(2010): 840-848.