INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:263–288 (DOI: 10.1002/fld.758)

Least-squares meshfree method for incompressible Navier–Stokes problems

Xiang Kun Zhang[§], Kie-Chan Kwon[¶] and Sung-Kie Youn[*,†,‡]

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Gusung-dong, Yusung-ku, Taejon, 305-701, Korea

SUMMARY

A least-squares meshfree method based on the first-order velocity–pressure–vorticity formulation for two-dimensional incompressible Navier–Stokes problem is presented. The convective term is linearized by successive substitution or Newton's method. The discretization of all governing equations is implemented by the least-squares method. Equal-order moving least-squares approximation is employed with Gauss quadrature in the background cells. The boundary conditions are enforced by the penalty method. The matrix-free element-by-element Jacobi preconditioned conjugate method is applied to solve the discretized linear systems. Cavity flow for steady Navier–Stokes problem and the flow over a square obstacle for time-dependent Navier–Stokes problem are investigated for the presented least-squares meshfree method. The effects of inaccurate integration on the accuracy of the solution are investigated. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: least-squares; meshfree method; Navier–Stokes equation; LSMFM

1. INTRODUCTION

In computational fluid dynamics (CFD), various finite elements methods, finite difference methods or finite volume methods for incompressible fluid flow have been developed. Typically four approaches are commonly taken to implement the discretization process in finite element methods based on the velocity–pressure formulation. They are the classical Galerkin mixed method [1] (including the projection methods [2]), penalty method [3], streamline upwind Petrov–Galerkin (SUPG) method [4] and least-squares method [5], respectively. The classical Galerkin mixed method is restricted by Ladyzhenskaya–Babuška–Brezzi (LBB) condition. The resulting algebraic matrix is non-symmetric, and some oscillations on the result

[*] Correspondence to: Sung-Kien Youn, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Gusung-dong, Yusung-ku, Taejon, 305-701, Korea.
[†] E-mail: skyoun@sorak.kaist.ac.kr
[‡] Professor.
[§] Post-Doctoral Fellow.
[¶] Ph.D. Student.

Published online 14 July 2004
Received 24 February 2003
Revised 22 January 2004

of pressure are observed. In the penalty method, the penalty parameter affects the accuracy and convergence of the solution. In the SUPG method, resulting algebraic system is non-symmetric. Thus the leading three methods are not always satisfactory methods for large-scale problems in CFD. Compared with the former three methods, least-squares method is robust, which is based on the minimization of the squared residuals. The least-squares method can overcome the above difficulties. It can reduce oscillations and instability of the solutions from the methods based on Galerkin formulation, and its resulting system matrix is symmetric and positive definite; it is easier to use equal-order approximations on all variables which are computed in the fully coupled manner and can be efficiently solved by iterative methods for the large-scale computation; no special treatments, such as upwinding or adjustable parameters are required.

In attempts to reduce the meshing-related difficulties, many meshfree methods have been developed. Among them are the smooth particle hydrodynamics (SPH) [6, 7], generalized finite difference method (GFDM) [8–10], element-free Galerkin method (EFGM) [11, 12], reproducing kernel particle method (RKPM) [13, 14], partition of unity finite element method (PUM) [15], hp-cloud method [16, 17], meshless local Petrov–Galerkin approach (MLPG) [18], diffuse element method (DEM) [19], etc. Meshfree method does not involve remeshing process and easy to realize adaptivity strategy. For fluid dynamics problems, the meshfree methods have recently been employed. Liu *et al.* [20] used the reproducing kernel particle method (RKPM) with SUPG formulation to solve 2D advection–diffusion equation. Sadat and Couturier [21] employed the diffuse element method (DEM) with the project method to study the laminar natural convection problem. Yagawa and Shirazaki [22] applied free mesh method (FMM) with the weighed residual-Galerkin method to unsteady two-dimensional incompressible viscous flow. Cheng and Liu [23] adopted the finite point method (FPM) with the discretization defined by the positions of points to analyse two-dimensional driven cavity flow. Kim and Kim [24] presented some analyses of fluids by meshfree point collocation method (MPCM). The least-squares meshfree method (LSMFM) was proposed by Park and Youn [25]. The convergence under inaccurate integration, a posteriori error estimates and an adaptive scheme of LSMFM has been studied [26, 27] and the convergence behavior and approximation accuracy on Stokes problem by LSMFM have been presented [28].

Continuing the previous work in reference [28], the LSMFM will be further investigated in the application to the two-dimensional incompressible Navier–Stokes problem based on the first-order velocity–pressure–vorticity formulation. The numerical results will be discussed.

2. VELOCITY–PRESSURE–VORTICITY FOMULATION

2.1. Steady incompressible Navier–Stokes problem

The steady incompressible Navier–Stokes equations in first-order velocity–pressure–vorticity formulation for two-dimensional problem can be described as the following form:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$
\begin{aligned}
u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+\frac{\partial p}{\partial x}+\frac{1}{R_e}\frac{\partial \omega}{\partial y}&=f_x\\
u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+\frac{\partial p}{\partial y}-\frac{1}{R_e}\frac{\partial \omega}{\partial x}&=f_y \quad \text{in } \Omega\\
\omega+\frac{\partial u}{\partial y}-\frac{\partial v}{\partial x}&=0
\end{aligned}
\tag{1}
$$

Here all variables are non-dimensionalized, ω is the z-component of vorticity, (f_x, f_y) the body force vector, and the Reynolds number R_e is defined as

$$R_e = UD/v$$

where U is the characteristic velocity, D the characteristic length of the domain Ω and v the kinematic viscosity.

The convective term can be linearized by the successive substitution or Newton's method. Here both of linearization processes are given in the following.

Steady incompressible Navier–Stokes equations with successive substitution can be expressed as follows:

$$
\begin{aligned}
\frac{\partial u_{k+1}}{\partial x}+\frac{\partial v_{k+1}}{\partial y}&=0\\
u_k\frac{\partial u_{k+1}}{\partial x}+v_k\frac{\partial u_{k+1}}{\partial y}+\frac{\partial p_{k+1}}{\partial x}+\frac{1}{R_e}\frac{\partial \omega_{k+1}}{\partial y}&=f_x\\
u_k\frac{\partial v_{k+1}}{\partial x}+v_k\frac{\partial v_{k+1}}{\partial y}+\frac{\partial p_{k+1}}{\partial y}-\frac{1}{R_e}\frac{\partial \omega_{k+1}}{\partial x}&=f_y\\
\omega_{k+1}+\frac{\partial u_{k+1}}{\partial y}-\frac{\partial v_{k+1}}{\partial x}&=0
\end{aligned}
\tag{2}
$$

where the subscripts k and $k+1$ denote the kth and $(k+1)$th linearization step, respectively.

The above governing equation can be written in a general form of a first-order system:

$$\mathbf{Lu}=\mathbf{f} \tag{3}$$

where $\mathbf{L}$ is the first-order differential operator and can be written as

$$\mathbf{Lu}=\mathbf{A}_1\frac{\partial \mathbf{u}}{\partial x}+\mathbf{A}_2\frac{\partial \mathbf{u}}{\partial y}+\mathbf{A}_0\mathbf{u} \tag{4}$$

The coefficient matrix $\mathbf{A}_i$, the force vector $\mathbf{f}$ and $\mathbf{u}$ (including velocity, vorticity and pressure) are given as follows:

$$
\mathbf{A}_1=\begin{bmatrix}1&0&0&0\\u_k&0&1&0\\0&u_k&0&-1/R_e\\0&-1&0&0\end{bmatrix},\quad
\mathbf{A}_2=\begin{bmatrix}0&1&0&0\\v_k&0&0&1/R_e\\0&v_k&1&0\\1&0&0&0\end{bmatrix}
$$

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{f} = \begin{Bmatrix} 0 \\ f_x \\ f_y \\ 0 \end{Bmatrix}, \quad \mathbf{u} = \begin{Bmatrix} u \\ v \\ p \\ \omega \end{Bmatrix}_{k+1}$$

If linearization process is completed by Newton's method, Equation (1) can be written as follows:

$$\begin{aligned} \frac{\partial u_{k+1}}{\partial x} + \frac{\partial v_{k+1}}{\partial y} &= 0 \\ u_k \frac{\partial u_{k+1}}{\partial x} + v_k \frac{\partial u_{k+1}}{\partial y} + u_{k+1} \frac{\partial u_k}{\partial x} + v_{k+1} \frac{\partial u_k}{\partial y} + \frac{\partial p_{k+1}}{\partial x} + \frac{1}{R_e} \frac{\partial \omega_{k+1}}{\partial y} &= f_x + u_k \frac{\partial u_k}{\partial x} + v_k \frac{\partial u_k}{\partial y} \\ u_k \frac{\partial v_{k+1}}{\partial x} + v_k \frac{\partial v_{k+1}}{\partial y} + u_{k+1} \frac{\partial v_k}{\partial x} + v_{k+1} \frac{\partial v_k}{\partial y} + \frac{\partial p_{k+1}}{\partial y} - \frac{1}{R_e} \frac{\partial \omega_{k+1}}{\partial x} &= f_y + u_k \frac{\partial v_k}{\partial x} + v_k \frac{\partial v_k}{\partial y} \\ \omega_{k+1} + \frac{\partial u_{k+1}}{\partial y} - \frac{\partial v_{k+1}}{\partial x} &= 0 \end{aligned} \tag{5}$$

It can be also written as the standard form of Equation (3), the coefficient matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are same as those from successive substitution, but matrix $\mathbf{A}_0$ and force vector $\mathbf{f}$ are different and given as follows:

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{\partial u_k}{\partial x} & \dfrac{\partial u_k}{\partial y} & 0 & 0 \\ \dfrac{\partial v_k}{\partial x} & \dfrac{\partial v_k}{\partial y} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{f} = \begin{Bmatrix} 0 \\ f_x + u_k \dfrac{\partial u_k}{\partial x} + v_k \dfrac{\partial u_k}{\partial y} \\ f_y + u_k \dfrac{\partial v_k}{\partial x} + v_k \dfrac{\partial v_k}{\partial y} \\ 0 \end{Bmatrix} \tag{6}$$

2.2. Time-dependent incompressible Navier–Stokes problem

The time-dependent incompressible Navier–Stokes equations in first-order velocity–pressure–vorticity formulation for two-dimensional problems can be described by the following dimensionless form:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} + \frac{1}{R_e}\frac{\partial \omega}{\partial y} = f_x$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{R_e}\frac{\partial \omega}{\partial x} = f_y \quad \text{in } \Omega$$

$$\omega + \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \tag{7}$$

For above equations, time discretization can be completed by θ-method for the time step $\Delta t = t^{n+1} - t^n$. The value of θ can be selected among the following schemes.

$$\theta = \begin{cases} 0 & \text{Forward-Euler scheme} \quad O(\Delta t) \\ 1/2 & \text{Crank-Nicolson scheme} \quad O(\Delta t^2) \\ 1 & \text{Backward-Euler scheme} \quad O(\Delta t) \end{cases} \tag{8}$$

Backward–Euler scheme and Crank-Nicolson scheme are unconditionally stable and there is no limitation on the size of the time step. After temporal discretization, Equation (7) can be written as follows at $(n+1)$th time step:

$$\frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} = 0 \tag{9a}$$

$$\begin{aligned} &\frac{u^{n+1} - u^n}{\Delta t} + \theta\left(u^{n+1}\frac{\partial u^{n+1}}{\partial x} + v^{n+1}\frac{\partial u^{n+1}}{\partial y} + \frac{\partial p^{n+1}}{\partial x} + \frac{1}{R_e}\frac{\partial \omega^{n+1}}{\partial y}\right) \\ &\quad +(1-\theta)\left(u^n\frac{\partial u^n}{\partial x} + v^n\frac{\partial u^n}{\partial y} + \frac{\partial p^n}{\partial x} + \frac{1}{R_e}\frac{\partial \omega^n}{\partial y}\right) = f_x^{n+1} \end{aligned} \tag{9b}$$

$$\begin{aligned} &\frac{v^{n+1} - v^n}{\Delta t} + \theta\left(u^{n+1}\frac{\partial v^{n+1}}{\partial x} + v^{n+1}\frac{\partial v^{n+1}}{\partial y} + \frac{\partial p^{n+1}}{\partial y} - \frac{1}{R_e}\frac{\partial \omega^{n+1}}{\partial x}\right) \\ &\quad +(1-\theta)\left(u^n\frac{\partial v^n}{\partial x} + v^n\frac{\partial v^n}{\partial y} + \frac{\partial p^n}{\partial y} - \frac{1}{R_e}\frac{\partial \omega^n}{\partial x}\right) = f_y^{n+1} \end{aligned} \tag{9c}$$

$$\omega^{n+1} + \frac{\partial u^{n+1}}{\partial y} - \frac{\partial v^{n+1}}{\partial x} = 0 \tag{9d}$$

where the superscripts n and $n+1$ represent the nth and $(n+1)$th time step, respectively.

As described before, the convective term in governing equations can be linearized by using the successive substitution method or Newton's method. Here Newton's linearization is presented. Using Newton's method the linearized convective term at $(k+1)$th linearization step is given as

$$u_{k+1}^{n+1}\frac{\partial v_{k+1}^{n+1}}{\partial x} = u_k^{n+1}\frac{\partial v_{k+1}^{n+1}}{\partial x} + u_{k+1}^{n+1}\frac{\partial v_k^{n+1}}{\partial x} - u_k^{n+1}\frac{\partial v_k^{n+1}}{\partial x} \tag{10}$$

Thus we have the following form of the governing equations:

$$\frac{\partial u_{k+1}^{n+1}}{\partial x} + \frac{\partial v_{k+1}^{n+1}}{\partial y} = 0 \tag{11a}$$

$$\begin{aligned}&\frac{u_{k+1}^{n+1}-u^n}{\Delta t}+\theta\left(u_k^{n+1}\frac{\partial u_{k+1}^{n+1}}{\partial x}+u_{k+1}^{n+1}\frac{\partial u_k^{n+1}}{\partial x}-u_k^{n+1}\frac{\partial u_k^{n+1}}{\partial x}\right.\\&\quad\left.+v_k^{n+1}\frac{\partial u_{k+1}^{n+1}}{\partial y}+v_{k+1}^{n+1}\frac{\partial u_k^{n+1}}{\partial y}-v_k^{n+1}\frac{\partial u_k^{n+1}}{\partial y}+\frac{\partial p_{k+1}^{n+1}}{\partial x}+\frac{1}{R_e}\frac{\partial \omega_{k+1}^{n+1}}{\partial y}\right)\\&\quad+(1-\theta)\left(u^n\frac{\partial u^n}{\partial x}+v^n\frac{\partial u^n}{\partial y}+\frac{\partial p^n}{\partial x}+\frac{1}{R_e}\frac{\partial \omega^n}{\partial y}\right)=f_x^{n+1}\end{aligned}\tag{11b}$$

$$\begin{aligned}&\frac{v_{k+1}^{n+1}-v^n}{\Delta t}+\theta\left(u_k^{n+1}\frac{\partial v_{k+1}^{n+1}}{\partial x}+u_{k+1}^{n+1}\frac{\partial v_k^{n+1}}{\partial x}-u_k^{n+1}\frac{\partial v_k^{n+1}}{\partial x}\right.\\&\quad\left.+v_k^{n+1}\frac{\partial v_{k+1}^{n+1}}{\partial y}+v_{k+1}^{n+1}\frac{\partial v_k^{n+1}}{\partial y}-v_k^{n+1}\frac{\partial v_k^{n+1}}{\partial y}+\frac{\partial p_{k+1}^{n+1}}{\partial y}-\frac{1}{R_e}\frac{\partial \omega_{k+1}^{n+1}}{\partial x}\right)\\&\quad+(1-\theta)\left(u^n\frac{\partial v^n}{\partial x}+v^n\frac{\partial v^n}{\partial y}+\frac{\partial p^n}{\partial y}-\frac{1}{R_e}\frac{\partial \omega^n}{\partial x}\right)=f_y^{n+1}\end{aligned}\tag{11c}$$

$$\omega_{k+1}^{n+1}+\frac{\partial u_{k+1}^{n+1}}{\partial y}-\frac{\partial v_{k+1}^{n+1}}{\partial x}=0\tag{11d}$$

It should be noted that $u_0^{n+1}=u^n$ in above equations.

The above equations can be also written as the standard form of Equation (3) and corresponding $\mathbf{u}$, coefficient matrix $\mathbf{A}_i$, and the force vector $\mathbf{f}$ are given as follows:

$$\mathbf{u}_{k+1}^{n+1}=\left\{\begin{matrix}u_{k+1}^{n+1}\\ v_{k+1}^{n+1}\\ p_{k+1}^{n+1}\\ \omega_{k+1}^{n+1}\end{matrix}\right\},\quad \mathbf{A}_0=\begin{bmatrix}0 & 0 & 0 & 0\\ \dfrac{1}{\Delta t}+\theta\dfrac{\partial u_k^{n+1}}{\partial x} & \theta\dfrac{\partial u_k^{n+1}}{\partial y} & 0 & 0\\ \theta\dfrac{\partial v_k^{n+1}}{\partial x} & \dfrac{1}{\Delta t}+\theta\dfrac{\partial v_k^{n+1}}{\partial y} & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{A}_1=\begin{bmatrix}1 & 0 & 0 & 0\\ \theta u_k^{n+1} & 0 & \theta & 0\\ 0 & \theta u_k^{n+1} & 0 & -\theta\dfrac{1}{R_e}\\ 0 & -1 & 0 & 0\end{bmatrix},\quad \mathbf{A}_2=\begin{bmatrix}0 & 1 & 0 & 0\\ \theta v_k^{n+1} & 0 & 0 & \theta\dfrac{1}{R_e}\\ 0 & \theta v_k^{n+1} & \theta & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\tag{12}$$

$$\mathbf{f}=\left\{\begin{matrix}0\\ f_x^{n+1}+\dfrac{u^n}{\Delta t}-(1-\theta)\left(u^n\dfrac{\partial u^n}{\partial x}+v^n\dfrac{\partial u^n}{\partial y}+\dfrac{\partial p^n}{\partial x}+\dfrac{1}{R_e}\dfrac{\partial \omega^n}{\partial y}\right)+\theta\left(u_k^{n+1}\dfrac{\partial u_k^{n+1}}{\partial x}+v_k^{n+1}\dfrac{\partial u_k^{n+1}}{\partial y}\right)\\ f_y^{n+1}+\dfrac{v^n}{\Delta t}-(1-\theta)\left(u^n\dfrac{\partial v^n}{\partial x}+v^n\dfrac{\partial v^n}{\partial y}+\dfrac{\partial p^n}{\partial y}-\dfrac{1}{R_e}\dfrac{\partial \omega^n}{\partial x}\right)+\theta\left(u_k^{n+1}\dfrac{\partial v_k^{n+1}}{\partial x}+v_k^{n+1}\dfrac{\partial v_k^{n+1}}{\partial y}\right)\\ 0\end{matrix}\right\}$$

3. MOVING LEAST-SQUARES APPROXIMATION

The moving least-squares (MLS) approximation scheme is widely used in meshfree methods and is also employed in the present work. Here only calculation scheme for shape functions and derivatives in terms of linear consistency condition is reviewed. The details can be found in References [29, 30].

The approximation $u^h(\mathbf{x})$ can be expressed as

$$u^h(\mathbf{x}) = \mathbf{N}(\mathbf{x})\hat{\mathbf{u}} \tag{13a}$$

$$\hat{\mathbf{u}}^{\mathrm{T}} = [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n] \tag{13b}$$

where $\mathbf{N}(\mathbf{x})$ is the shape function, $\hat{u}_I$ approximations to the values of continuous function $\mathbf{u}(\mathbf{x})$ at point $\mathbf{x}_I$, and n the number of points in the neighborhood of $\mathbf{x}$ for which the weight function is greater than zero.

The shape functions can be described in the form

$$N_I(\mathbf{x}) = \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{p}(\mathbf{x}_I)w_I(\mathbf{x}) \tag{14}$$

where $\boldsymbol{\alpha}$ is the coefficient vector, which is the functions of space coordinates $\mathbf{x} = [x, y, z]^{\mathrm{T}}$, $\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I) = [p_1(\mathbf{x}_I), p_2(\mathbf{x}_I), \ldots, p_m(\mathbf{x}_I)]$ a complete monomial basis of m dimensions at point $\mathbf{x}_I$, and $w_I(\mathbf{x})$ the weight function associated with node I which should be compactly supported.

The linear consistency equation can be expressed as follows:

$$\sum_I N_I(\mathbf{x})\mathbf{p}(\mathbf{x}_I) = \mathbf{p}(\mathbf{x}) = [1 \;\; x \;\; y]^{\mathrm{T}} \tag{15}$$

Substituting Equation (14) into Equation (15) results in

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{p}(\mathbf{x}) \tag{16}$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{n} w_I(\mathbf{x})\mathbf{p}(\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)^{\mathbf{T}} \tag{17}$$

The derivatives of $\boldsymbol{\alpha}$ can be obtained by differentiation of Equation (16)

$$\mathbf{A}\boldsymbol{\alpha}_{,i} = \mathbf{p}_{,i}(\mathbf{x}) - \mathbf{A}_{,i}\boldsymbol{\alpha} \tag{18}$$

The derivatives of the shape function are then given by

$$N_{I,i} = \mathbf{p}(\mathbf{x}_I)^{\mathrm{T}}\boldsymbol{\alpha}_{,i}(\mathbf{x})w_I(\mathbf{x}) + \mathbf{p}(\mathbf{x}_I)^{\mathrm{T}}\boldsymbol{\alpha}(\mathbf{x})w_{I,i}(\mathbf{x}) \tag{19}$$

In this paper, linear complete basis in two-dimensional problem is used in numerical examples, and the following weight functions in Reference [16] are employed:

$$w_I(\mathbf{x}) = \begin{cases} \sqrt{4/\pi}\left(1 - \dfrac{\|\mathbf{x} - \mathbf{x}_I\|^2}{r_I^2}\right)^4 & \text{if } \|\mathbf{x} - \mathbf{x}_I\| < r_I \\ 0 & \text{if } \|\mathbf{x} - \mathbf{x}_I\| \geqslant r_I \end{cases} \tag{20}$$

where r_I is the size of nodal support selected by $r_I = \alpha d$ with d denoting the nodal spacing and constant α a chosen proportional coefficient.

4. THE LEAST-SQUARES FORMULATION

4.1. The least-squares formulation

The least-squares method is based on the minimization of the squared residuals. For the first-order system as the standard form given in Equation (3), the least-squares quadratic functional for the system is given as follows:

$$I(\mathbf{u}) = \int_{\Omega} (\mathbf{Lu} - \mathbf{f})^2 \, \mathrm{d}\Omega = \int_{\Omega} (\mathbf{Lu} - \mathbf{f})^{\mathrm{T}} (\mathbf{Lu} - \mathbf{f}) \, \mathrm{d}\Omega \tag{21}$$

In meshfree methods, several approaches are used to evaluate the integrals in the above equation, such as nodal integration, background cell integration with Gauss quadrature, background finite element mesh, etc. In the present paper, background cells are used with Gauss quadrature. Thus, Equation (21) can be written in the discrete form:

$$I(\mathbf{u}) = \sum_{i\text{cell}=1}^{N_{\text{cell}}} \sum_{j\text{gauss}=1}^{N_{\text{gauss}}} w_{j\text{gauss}} (\mathbf{Lu} - \mathbf{f})^{\mathrm{T}} (\mathbf{Lu} - \mathbf{f}) \tag{22}$$

where $w_{j\text{gauss}}$ is the weight factor at the integration point (including area information), N_{gauss} the number of Gauss integration points in each background cell, and N_{cell} the total number of background cells.

For all unknown variables, equal-order local approximation scheme could be employed. It should be noted that the following necessary condition for the existence of the solution must be satisfied:

$$\sum_{i\text{cell}=1}^{n\text{cell}} (N_{\text{gauss}} \times N_{\text{eq}}) \geqslant N_{\text{point}} \times N_{\text{dof}} - N_{\text{bc}} \tag{23}$$

where N_{eq} is the number of governing equations in the first-order system, N_{point} the total number of nodes, N_{dof} the number of degrees of freedom at each node, and N_{bc} the total number of specified nodal degrees of freedom at boundaries.

The approximation of $\mathbf{u}$ at each integration point can be written as in Equation (13) with equal-order approximation scheme. Minimizing Equation (22), the system matrix and vector for jth Gaussian integration point are obtained:

$$\mathbf{k}^{j\text{gauss}} = w_{j\text{gauss}} (\mathbf{LN})^{\mathrm{T}} (\mathbf{LN}) \tag{24a}$$

$$\mathbf{f}^{j\text{gauss}} = w_{j\text{gauss}} (\mathbf{LN})^{\mathrm{T}} \mathbf{f} \tag{24b}$$

For Equation (13), if there are N_{dof} variables at each node, the form of the shape function $\mathbf{N}$ is described as follows:

$$\mathbf{N} = [N_1 \mathbf{I} \; N_2 \mathbf{I} \; \cdots \; N_n \mathbf{I}] \tag{25}$$

where $\mathbf{I}$ is the $N_{\text{dof}} \times N_{\text{dof}}$ identity matrix.

For Equation (2), the component of $\mathbf{LN}$, $\mathbf{L}N_I$ in Equations (24a) and (24b) can be denoted as

$$LN_I = \begin{bmatrix} N_{I,x} & N_{I,y} & 0 & 0 \\ u_k N_{I,x} + v_k N_{I,y} & 0 & N_{I,x} & N_{I,y}/R_e \\ 0 & u_k N_{I,x} + v_k N_{I,y} & N_{I,y} & -N_{I,x}/R_e \\ N_{I,y} & -N_{I,x} & 0 & N_I \end{bmatrix} \tag{26}$$

For the system linearized by Newton's method as Equation (5), $\mathbf{L}N_I$ can be expressed as follows:

$$LN_I = \begin{bmatrix} N_{I,x} & N_{I,y} & 0 & 0 \\ u_k N_{I,x} + v_k N_{I,y} + u_{k,x} N_I & u_{k,y} N_I & N_{I,x} & N_{I,y}/R_e \\ v_{k,x} N_I & u_k N_{I,x} + v_k N_{I,y} + v_{k,y} N_I & N_{I,y} & -N_{I,x}/R_e \\ N_{I,y} & -N_{I,x} & 0 & N_I \end{bmatrix} \tag{27}$$

For the system linearized by Newton's method for time-dependent Navier–Stokes problem as Equation (12), $\mathbf{L}N_I$ can be written as

$$\mathbf{L}N_I = \begin{bmatrix} N_{I,x} & N_{I,y} & 0 & 0 \\ \left\{ \begin{array}{c} \theta u_k^{n+1} N_{I,x} + \theta v_k^{n+1} N_{I,y} \\ + \left(\dfrac{1}{\Delta t} + \theta \dfrac{\partial u_k^{n+1}}{\partial x} \right) N_I \end{array} \right\} & \theta \dfrac{\partial u_k^{n+1}}{\partial y} N_I & \theta N_{I,x} & \theta \dfrac{N_{I,y}}{R_e} \\ \theta \dfrac{\partial v_k^{n+1}}{\partial x} N_I & \left\{ \begin{array}{c} \theta u_k^{n+1} N_{I,x} + \theta v_k^{n+1} N_{I,y} \\ + \left(\dfrac{1}{\Delta t} + \theta \dfrac{\partial v_k^{n+1}}{\partial y} \right) N_I \end{array} \right\} & N_{I,y} & -\theta \dfrac{N_{I,x}}{R_e} \\ N_{I,y} & -N_{I,x} & 0 & N_I \end{bmatrix} \tag{28}$$

Assembling the system matrix and vector by the Gauss integration points in all background cells, the linear algebraic equation is obtained:

$$\mathbf{K}_\mathrm{G} \hat{\mathbf{U}} = \mathbf{F}_\mathrm{G} \tag{29}$$

where $\mathbf{K}_\mathrm{G}$ and $\mathbf{F}_\mathrm{G}$ are the global system matrix and vector, respectively, and $\hat{\mathbf{U}}$ the approximation to the global vector of nodal unknowns.

$$\mathbf{K}_\mathrm{G} = \sum_{i\mathrm{cell}=1}^{N_\mathrm{cell}} \sum_{j\mathrm{gauss}=1}^{N_\mathrm{gauss}} \mathbf{k}^{j\mathrm{gauss}} \tag{30a}$$

$$\mathbf{F}_\mathrm{G} = \sum_{i\mathrm{cell}=1}^{N_\mathrm{cell}} \sum_{j\mathrm{gauss}=1}^{N_\mathrm{gauss}} \mathbf{f}^{j\mathrm{gauss}} \tag{30b}$$

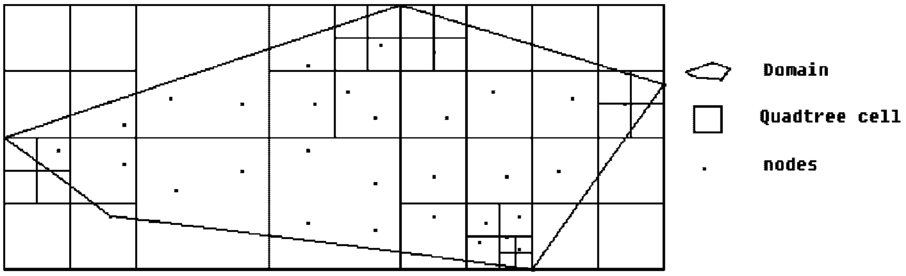

Figure 1. Quadtree algorithm.

It should be noted that the global system matrix $\mathbf{K}_G$ is symmetric and positive definite. Therefore, the resulting linear system of Equation (29) can be solved by the matrix-free element-by-element Jacobi preconditioned conjugate gradient (MFEBEJCG) method. It is simple to get the Jacobi preconditioned matrix. It needs not to form the global stiffness matrix at the evaluation points, so that computation cost can be reduced greatly [31]. It should be mentioned that the element-by-element technique is used in FEM because assemblage process is performed on the element level. In meshfree methods, assemblage process is carried out at all evaluation points. Here, the same nomenclature is used in the present paper.

4.2. Generation of evaluation points by quadtree algorithm

In Equation (22), the evaluation points and the weight factors should be determined in background cells. In meshfree methods, there are many simple integration schemes used to generate evaluation points, such as quadtree/octree algorithm, Delaunay triangulation, Voronoi cell and so on [32, 33]. In the present work, the quadtree algorithm is used in the two-dimensional geometry model. Quadtree algorithm is a well-known algorithm in computational geometry, which is easy to construct evaluation points for the complex model with Gaussian quadrature rule. The quadtree algorithm used in the present work is as follows (Figure 1):

(i) Define the maximum number of nodes $N_{\max}$ in each background cell. In this work, different $N_{\max}$ is used. $N_{\max}=4$ is used if the cell does not contain boundary nodes, or else $N_{\max}=1$.

(ii) Give the rectangular background domain to cover the computational domain.

(iii) Initial square background cells are defined according to ratio between width and height of computational domain, here called as mother cells.

(iv) If a mother cell contains more than $N_{\max}$ nodes, then divide it into four squares with equal size, called as daughter cells. Loop this step until the number of nodes in the each cell is less than or equal to $N_{\max}$.

(v) Generate the evaluation points in background cells in terms of the given Gaussian quadrature rule in Table I.

(vi) The evaluation points outside computational domains are not considered in calculation.

Table I. Gauss quadrature rule for quadtree algorithm.

No. of nodes in cell	Quadrature order (rule 1)	Quadrature order (rule 2)
0	2×2	1×1
1	3×3	1×1
2	3×3	2×2
3	4×4	2×2
4	4×4	2×2

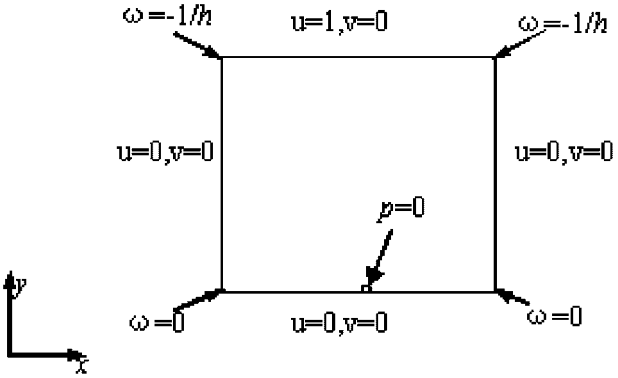

Figure 2. The boundary conditions for the cavity flow.

5. NUMERICAL RESULTS

5.1. Cavity flow for steady Navier–Stokes problem

The driven-cavity flow in unit square domain is considered to test the LSMFM for the steady Navier–Stokes problem.

The boundary conditions are shown in Figure 2. On the top side $u=1$ and $v=0$ are given, and no slip boundary conditions, i.e. $u=0$, $v=0$, are prescribed on the left, right and bottom sides. At the mid-point of the bottom side, the pressure condition $p=0$ is specified. We have $\omega = -\partial u/\partial y + \partial v/\partial x = -1/h + 0 = -1/h = -50$ at the two upper corners, where h is the closest distance from the nodes on the left side or the right side to the corresponding node at the upper corner, $\omega=0$ are given at the two bottom corners. As an initial trial solution for $R_e = 100$, $u=v=p=\omega=0$ are taken.

In meshfree methods, the nodal distribution will affect the computational results. Thus randomly generated nodes will be considered first. The four irregular distributions of nodes are shown in Figure 3. In the present study, the size of nodal support can be changed by the proportional coefficient α, here $\alpha=1.5$ is used. The nodal spacing d is defined as the maximum distance from the closest nodes to the given node in four quadrants. The quadtree algorithm is employed to generate evaluation points in background (quadtree) cells with Gaussian quadrature rule 1. The background domain is coincident with the computational domain.

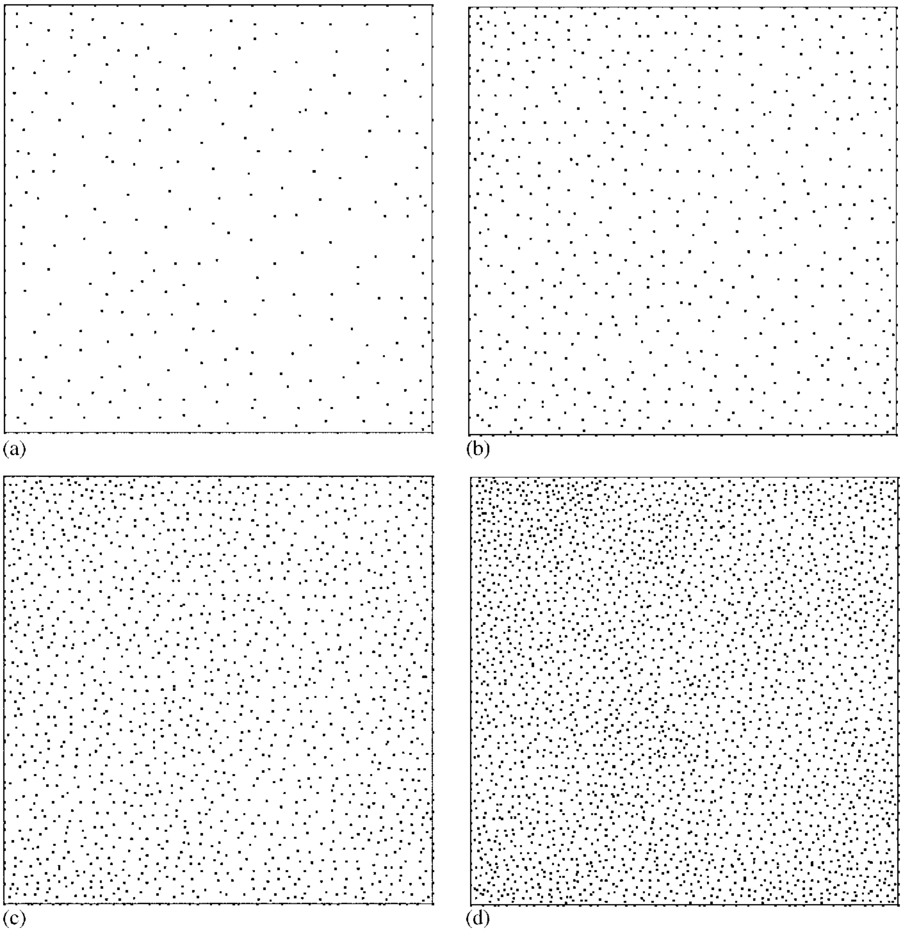

Figure 3. Irregular distributions of nodes for the cavity flow.

The linear basis is used. Equal-order MLS approximation is adopted. The matrix-free element-by-element Jacobi preconditioned conjugate gradient (MFEBEJCG) method is used to solve the resulting linear algebraic equations. Newton' s method is used to linearize the convective term. Penalty method is used to enforce the boundary conditions in LSMFM. The residual of the linear system at each variable is used as the stopping criterion of MFEBEJCG iteration.

$$\max_{1 \leqslant i \leqslant N_{\text{point}} \times N_{\text{dof}}} |\{r\}_i| < 10^{-6} \tag{31}$$

where i denotes the ith degree of freedom.

The stopping criterion for Newton's linearization is given as

$$\frac{\|\mathbf{U}^{k+1} - \mathbf{U}^k\|_2}{\|\mathbf{U}^k\|_2} < 10^{-3} \tag{32}$$

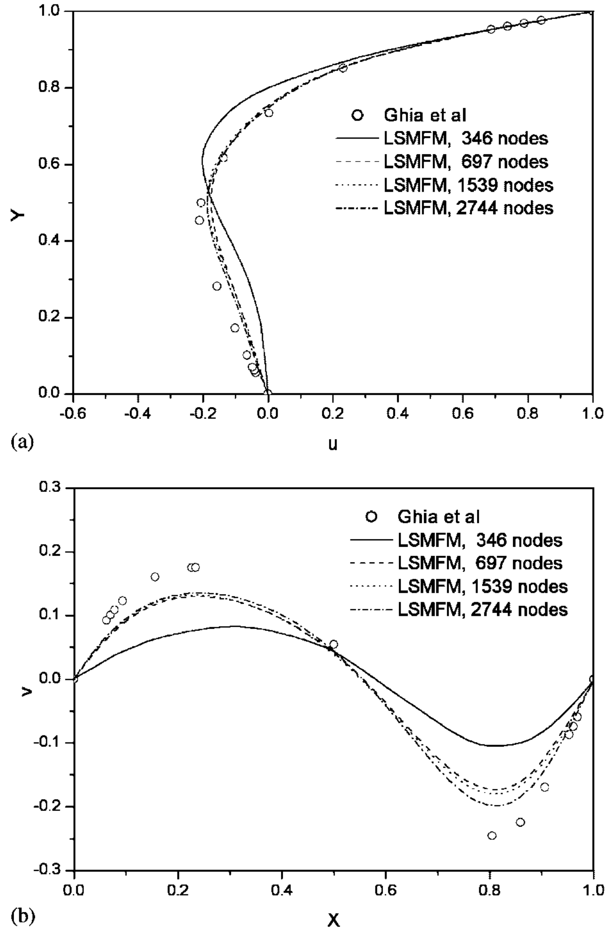

Figure 4. Profiles of u-velocity along $x=0.5$ and v-velocity along $y=0.5$ for irregular nodal distributions.

where $\|\mathbf{U}^k\|_2=\sqrt{\sum_{i=1}^{N_{\text{point}}\times N_{\text{dof}}}(\mathbf{U}_i^k)^2}$ is the L^2-norm of all unknown variables, the subscripts k and $k+1$ denote the kth and the $(k+1)$th linearization step, respectively.

Figure 4 shows the results of the horizontal velocity profiles along $x=0.5$ and vertical velocity profiles along $y=0.5$ for different irregular distributions of nodes.

Since background cells might not coincide with the complex computational domain, background cells would cut the boundaries of domain arbitrarily such that the integration becomes more inaccurate in the vicinity of boundary. Besides this, different Gauss quadrature rule in background cells will influence the accuracy of solution. The following four schemes will be used to investigate the solution accuracy:

Scheme 1: Irregular nodal distribution (the fourth nodal distribution in Figure 3) with quadtree cells. The background domain is coincident with the computational domain. Gauss quadrature rule 1 in Table I is used.

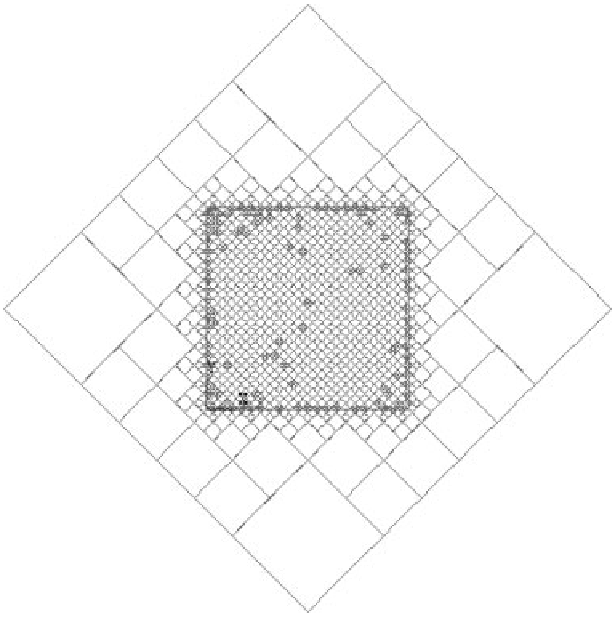

Figure 5. Inclined background cells which are not coincident with the computational domain.

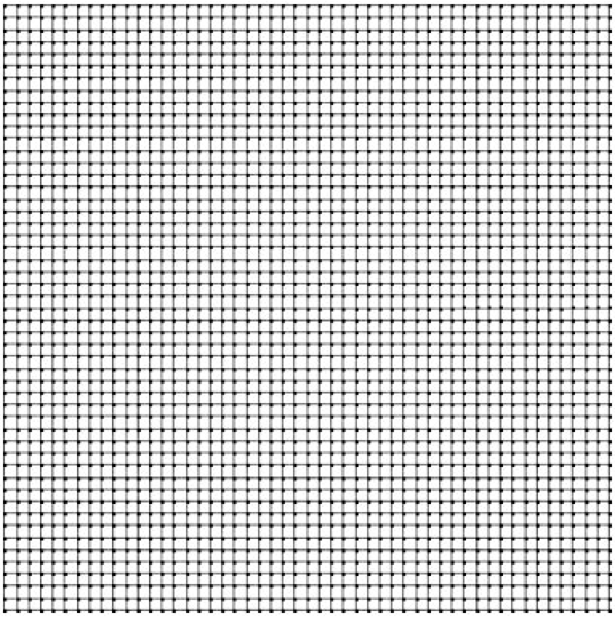

Figure 6. Regular nodal distribution where nodes are at the vertices of background cells.

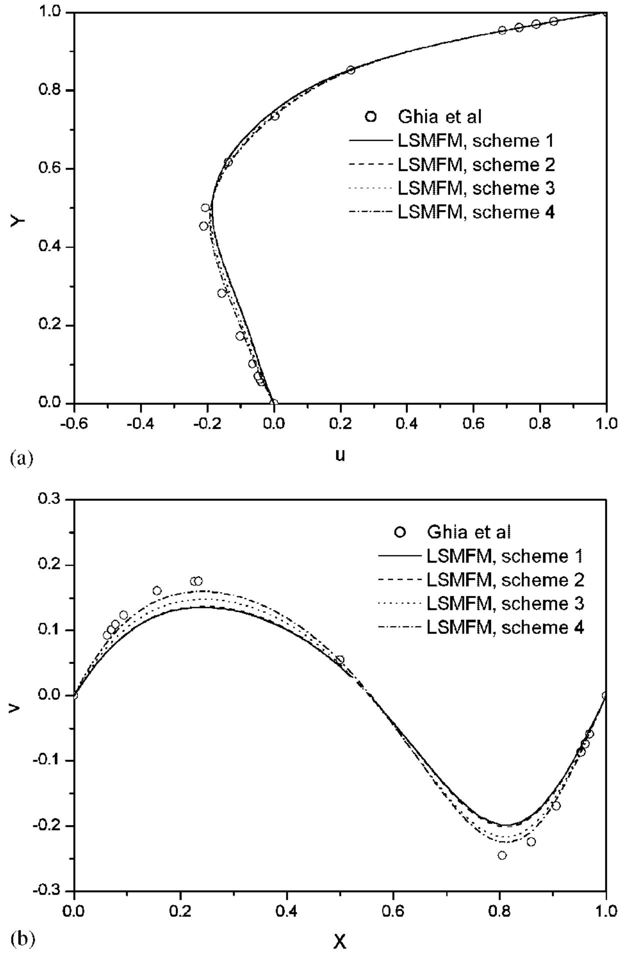

Figure 7. Profiles of u-velocity along $x = 0.5$ and v-velocity along $y = 0.5$ for different schemes.

Scheme 2: Irregular nodal distribution (the fourth nodal distribution in Figure 3) with quadtree cells. The background domain is inclined in Figure 5, which is not coincident with the computational domain. Gauss quadrature rule 1 in Table I is used.

Scheme 3: Irregular nodal distribution (the fourth nodal distribution in Figure 3) with quadtree cells. The background domain is coincident with the computational domain. Gauss quadrature rule 2 in Table I is used.

Scheme 4: Regular nodal distribution (in Figure 6). Quadtree algorithm is not used. The background domain is coincident with the computational domain. The computational domain is divided into uniform 50×50 square background cells, where the vertices of the cells coincide with the nodes. 2×2 Gauss quadrature is used in each background cells.

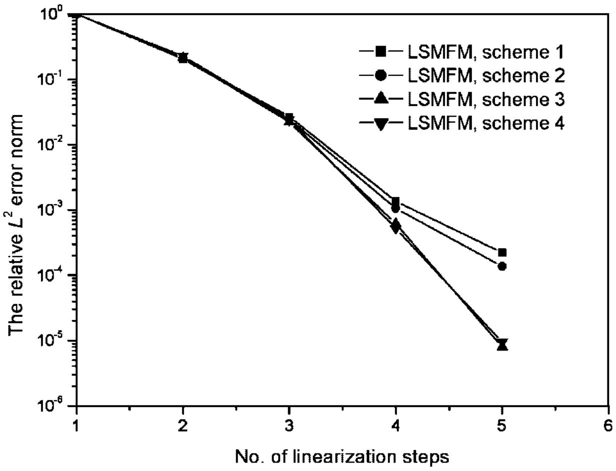

Figure 8. Convergence history for different schemes.

Here let us define the following relative error of all variables in L^2-norm between two schemes, i.e. between schemes 1 and 2.

$$\text{Relative error} = \frac{\|\mathbf{U}^{\text{scheme1}} - \mathbf{U}^{\text{scheme2}}\|_2}{\|\mathbf{U}^{\text{scheme1}}\|_2}$$

where

$$\|\mathbf{U}^{\text{scheme1}}\|_2 = \sqrt{\sum_{i=1}^{N_{\text{point}} \times N_{\text{dof}}} (\mathbf{U}_i^{\text{scheme1}})^2}$$

In the above four schemes, the proportional coefficient $\alpha = 1.5$ is used.

It can be seen from Figure 7 that the results from schemes 1 and 2 are almost coincident. The relative error of all variables between scheme 1 and scheme 2 in L^2-norm is about 2.6%. Thus the effects of inaccurate integration on the accuracy of the solution are quite small. In comparison with schemes 1 and 3, it can be found that lower-order Gaussian quadrature will have better accuracy of the solution. For above four schemes, the result in scheme 4 leads to the best accuracy of the solution. Figure 8 shows the convergence history of the relative L^2-error norm of linearization steps for different schemes. The required numbers of linearization steps are $5, 5, 4$ and 4, respectively.

From above comparisons on the accuracy of the solution, regular nodal distribution and background cells in scheme 4 will be employed to discuss the effect of the size of nodal support and the order of Gauss quadrature rule on the approximation accuracy in the following computations. According to the number of the background cells, nodes and boundary conditions, the reduced integration, i.e. one-point integration, can be also used.

Figure 9 presents the results of the horizontal velocity profiles along $x = 0.5$ and vertical velocity profiles along $y = 0.5$ for the different size of nodal support and Gauss quadrature rules. It can be seen that the results of one-point Gauss integration have better agreement with those from Ghia *et al.* than those of higher-order Gauss integrations when the proportional

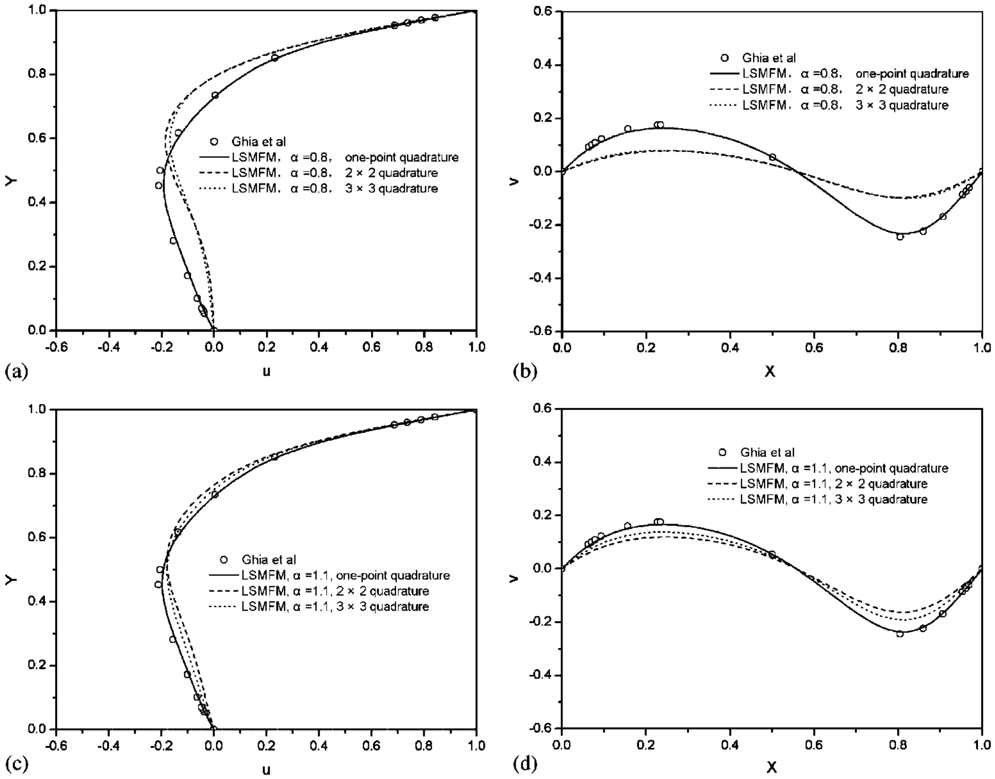

Figure 9. Profiles of u-velocity along $x=0.5$ and v-velocity along $y=0.5$ at different sizes of nodal support and different Gaussian quadrature rules.

size of nodal support α is less than 2.0, but the contrary results are obtained for the large size of nodal support $\alpha=2.0$. From Figure 10, it can be found that the results for the proportional size of nodal support $\alpha=1.5$ with one-point Gauss integration are comparable to those from Ghia *et al.* Too large size of nodal support leads to the bad approximation accuracy because the shape functions are strongly non-polynomial. Moreover, much more computation cost is required. Thus in the following calculations $\alpha=1.5$ and one-point Gauss integration are employed.

Finally, the cavity flows at different Reynolds number are investigated. Here the refined uniform 100×100 square background cells are used in computation. The vertices of the integration cells coincide with the nodes. The vorticity boundary conditions at the two upper corners are given as $\omega=-\partial u/\partial y+\partial v/\partial x=-1/h+0=-1/h=-100$. In calculation, $u=v=p=\omega=0$ are taken as an initial trial solution for $R_e=100$ and then the convergent solution for $R_e=100$ is used as the initial guess for $R_e=400$, and so on.

From Figure 11, it can be seen the comparison between the results of LSMFM and those of Ghia *et al.* [34] in terms of the velocity components along the central axis of the cavity, which are almost identical.

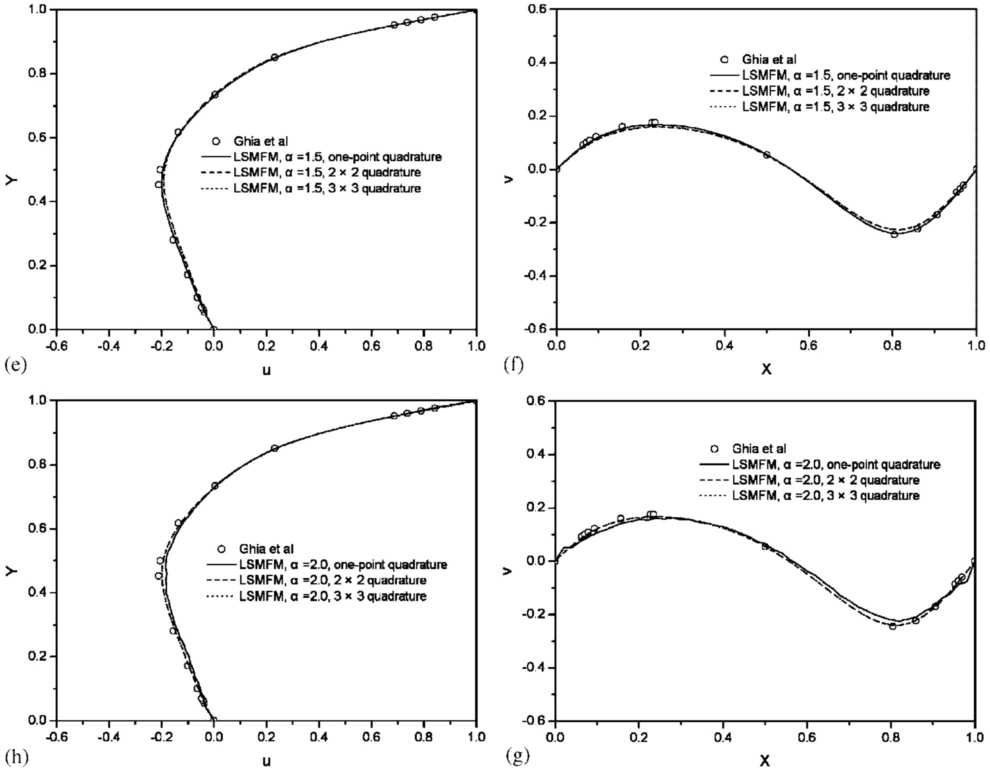

Figure 9. *Continued.*

Figures 12–14 present the computed results of velocity, streamlines, and the contours of vorticity and pressure at different Reynolds numbers. The results are in good agreement with those of Ghia *et al.* The results are remarkable considering that one-point Gauss rule is used in the integration.

5.2. Flow over a square obstacle for time-dependent Navier–Stokes problem

For time-dependent Navier–Stokes problem, the flow over a square obstacle is analysed to investigate the LSMFM. The computational domain is divided into uniform square background cells, where the vertices of the cells coincide with the nodes. The regular nodal distribution will be generated. Figure 15 gives the geometry of the computational domain and background cells (9934 cells and 9641 nodes). The boundary conditions are given as follows [35]: uniform velocity $u=1$, $v=0$ are enforced at the inlet, $p=\omega=0$ are given at the outflow boundary, no-slip conditions $u=v=0$ are imposed along the walls and the square obstacle.

In calculation, $u=v=p=\omega=0$ are taken as an initial trial solution and Reynolds number is 200. Backward-Euler time scheme is employed with time step $\Delta t=1.0$. One Newton's linearization step is used in each time step. As mentioned before, the value of α for the support size is 1.5 and one-point Gauss quadrature can be adopted in uniform background cells when the computation condition is satisfied in Equation (23). Penalty method is used

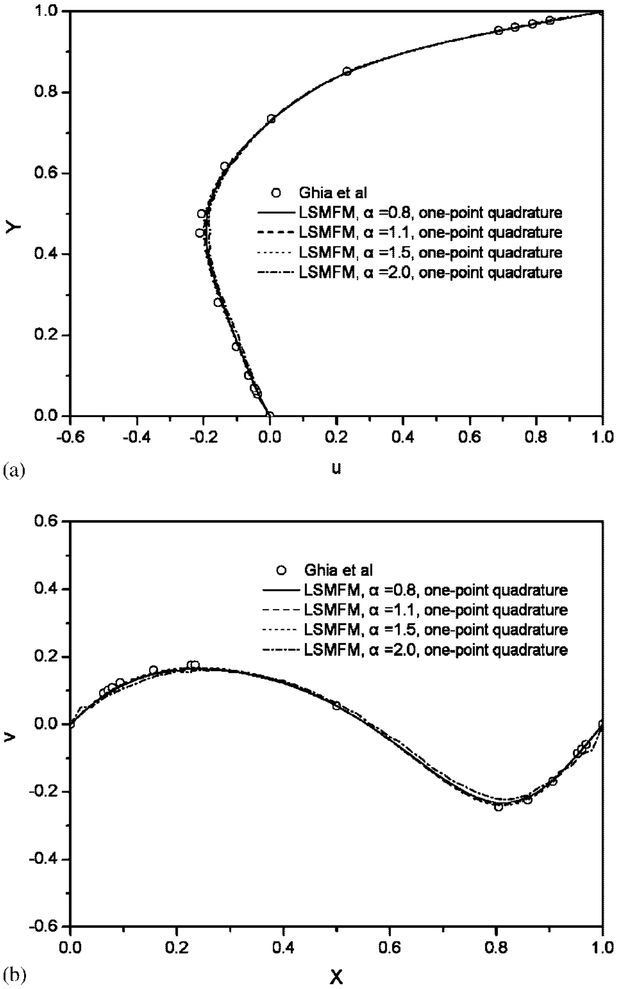

Figure 10. Profiles of u-velocity along $x=0.5$ and v-velocity along $y=0.5$ at different sizes of nodal support with one-point quadrature.

to enforce the boundary condition in LSMFM. Equal-order MLS approximation is employed. The matrix-free element-by-element Jacobi preconditioned conjugate gradient (MFEBEJCG) method is used to solve the resulting linear algebraic equations. The stopping criterion of MFEBEJCG iteration is given as Equation (31). The following criterion is used to judge when the steady-state solution is obtained:

$$\frac{\|\mathbf{U}^{n+1}-\mathbf{U}^{n}\|_2}{\|\mathbf{U}^{n}\|_2}<10^{-5} \tag{33}$$

where the superscripts n and $n+1$ represent the nth and the $(n+1)$th time step, respectively.

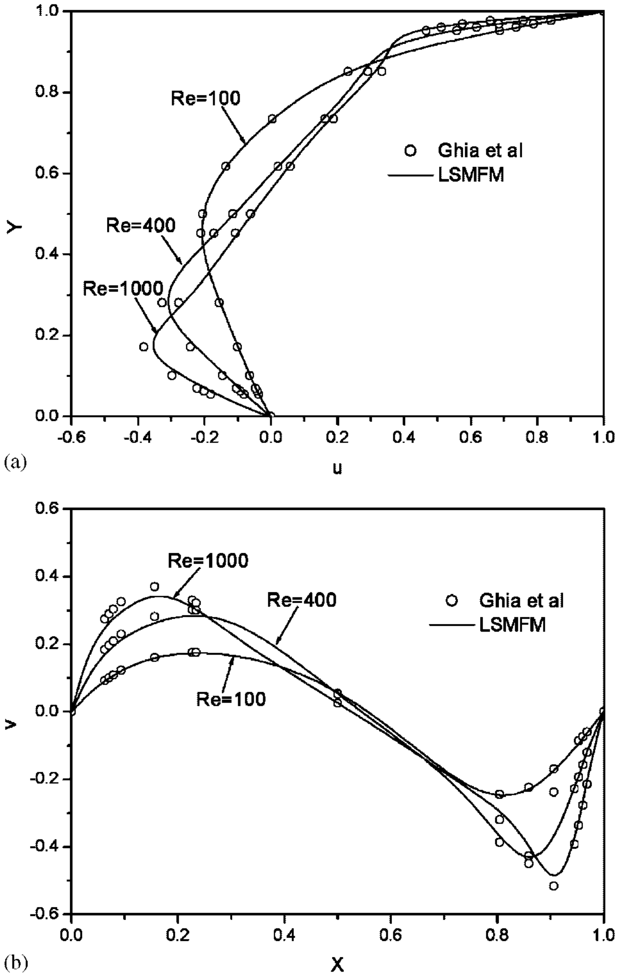

Figure 11. Profiles of u-velocity along $x = 0.5$ and v-velocity along $y = 0.5$ at different Reynolds numbers (the size of nodal support $\alpha = 1.5$; one-point quadrature).

Figure 16 presents the results of streamline at $t = 1, 2, 5, 11, 21$ and 31, respectively. At $t = 31$, the solution gives the convergent results, which satisfy the stopping criterion in Equation (33), and the eddy occurs near $x = 4$, which agrees well with the results from references [35–37]. Figure 17 gives contours of pressure and vorticity at $t = 31$.

6. SUMMARY AND CONCLUSIONS

The least-squares meshfree method (LSMFM) has been presented for the numerical analysis of incompressible Navier–Stokes problem. Velocity–pressure–vorticity formulation for two-

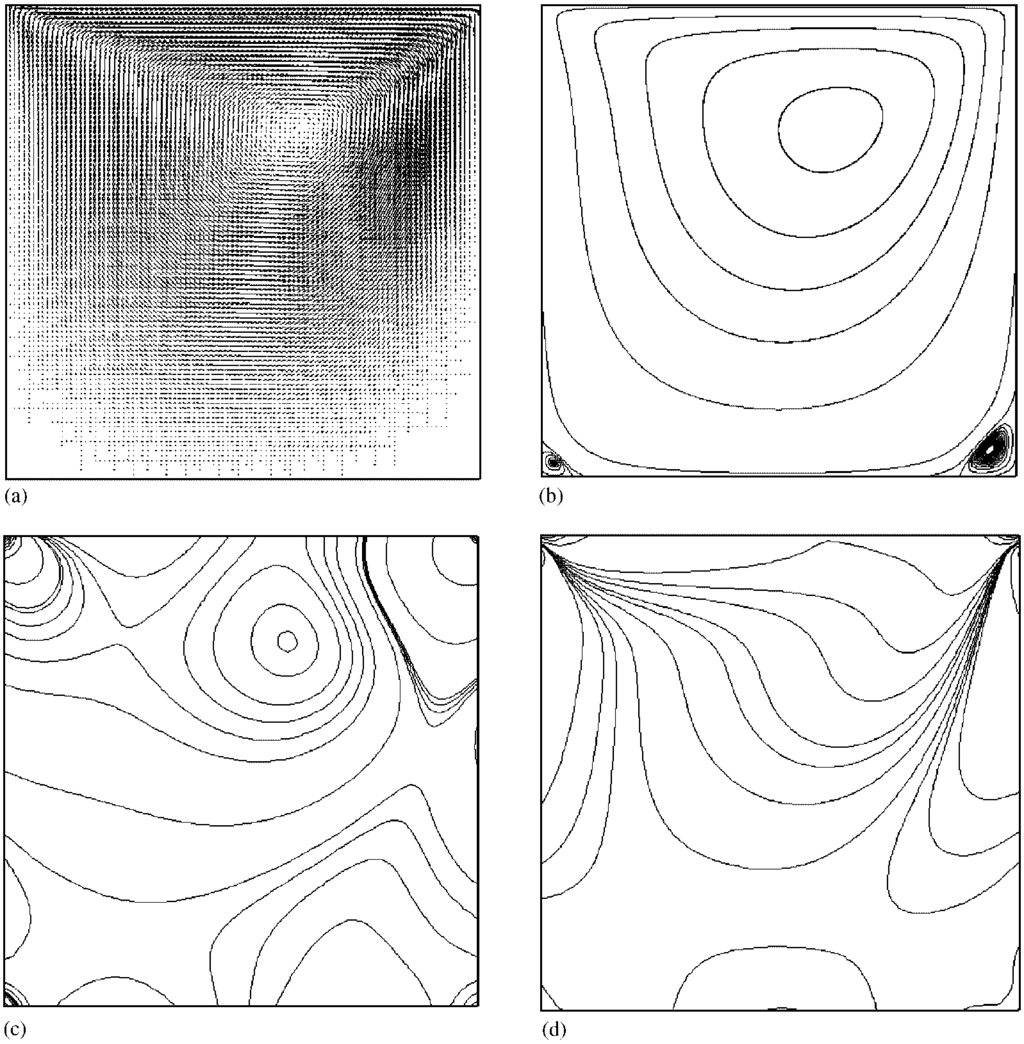

Figure 12. Numerical results for cavity flow at $R_e = 100$: (a) Velocity; (b) Streamline; (c) Pressure; and (d) Vorticity.

dimensional incompressible Navier–Stokes problem has been presented. Equal-order moving Least–Squares (MLS) approximation with Gauss quadrature in background cells was employed. Quadtree algorithm was employed to construct background cells with irregular distributions of nodes. The results of cavity flow for steady Navier–Stokes problem and the flow over a square obstacle for time-dependent Navier–Stokes problem by LSMFM compare well with the benchmarks.

For a second-order system, at least quadratic completeness for MLS shape functions is required to ensure convergence since the second derivative of the variable is involved, so

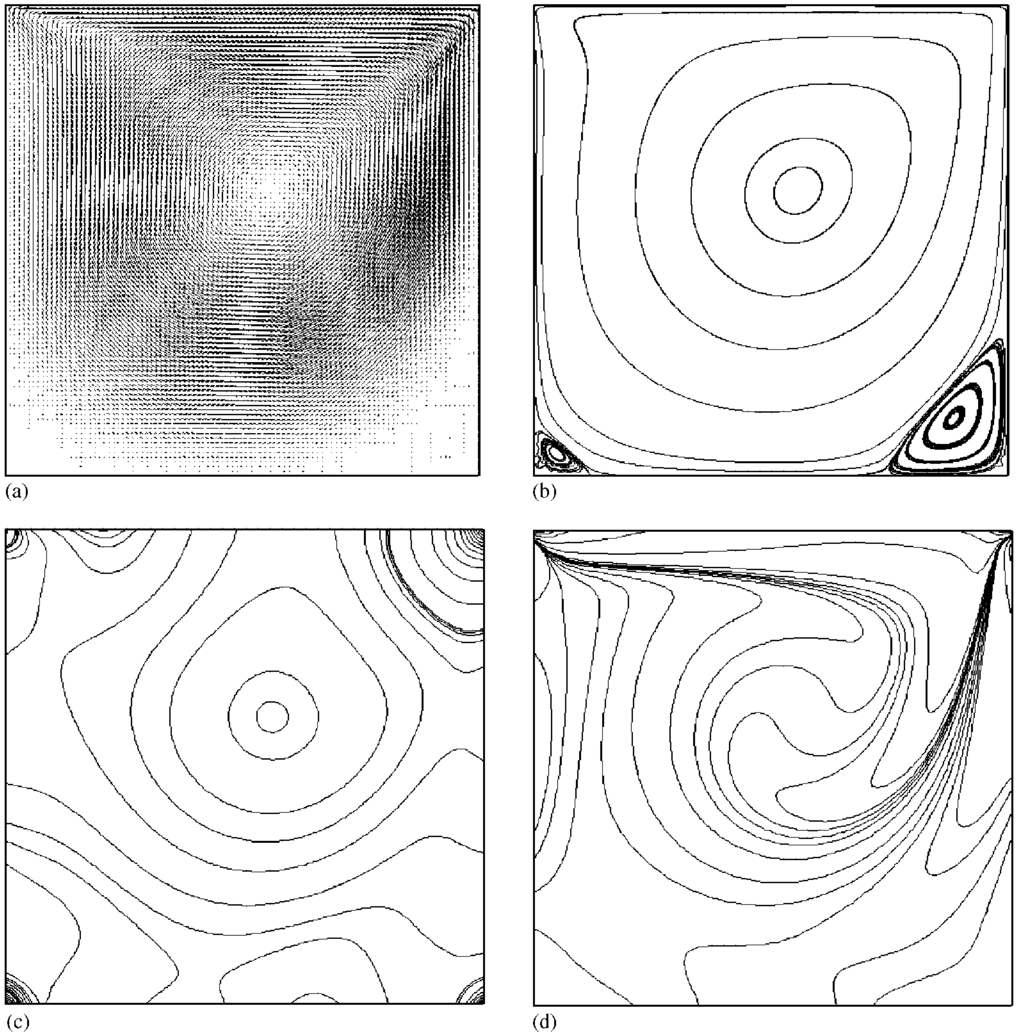

Figure 13. Numerical results for cavity flow at $R_e = 400$: (a) Velocity; (b) Streamline; (c) Pressure; and (d) Vorticity.

the quadratic basis function and large size of nodal supports should be used. But for first-order velocity–pressure–vorticity system, only linear basis function shape functions can be used. Also, costly solutions of Poisson equation are avoided. However, the discretized system must be solved efficiently due to the increase of unknown variables and additional equations. For LSMFM, the resulting system is symmetric and positive definite, and can be efficiently solved by iterative methods for large-scale problems. All examples in the present work are computed by matrix-free element-by-element Jacobi preconditioned conjugate method. For LSMFM, equal-order approximation can be applied to all the variables. All governing

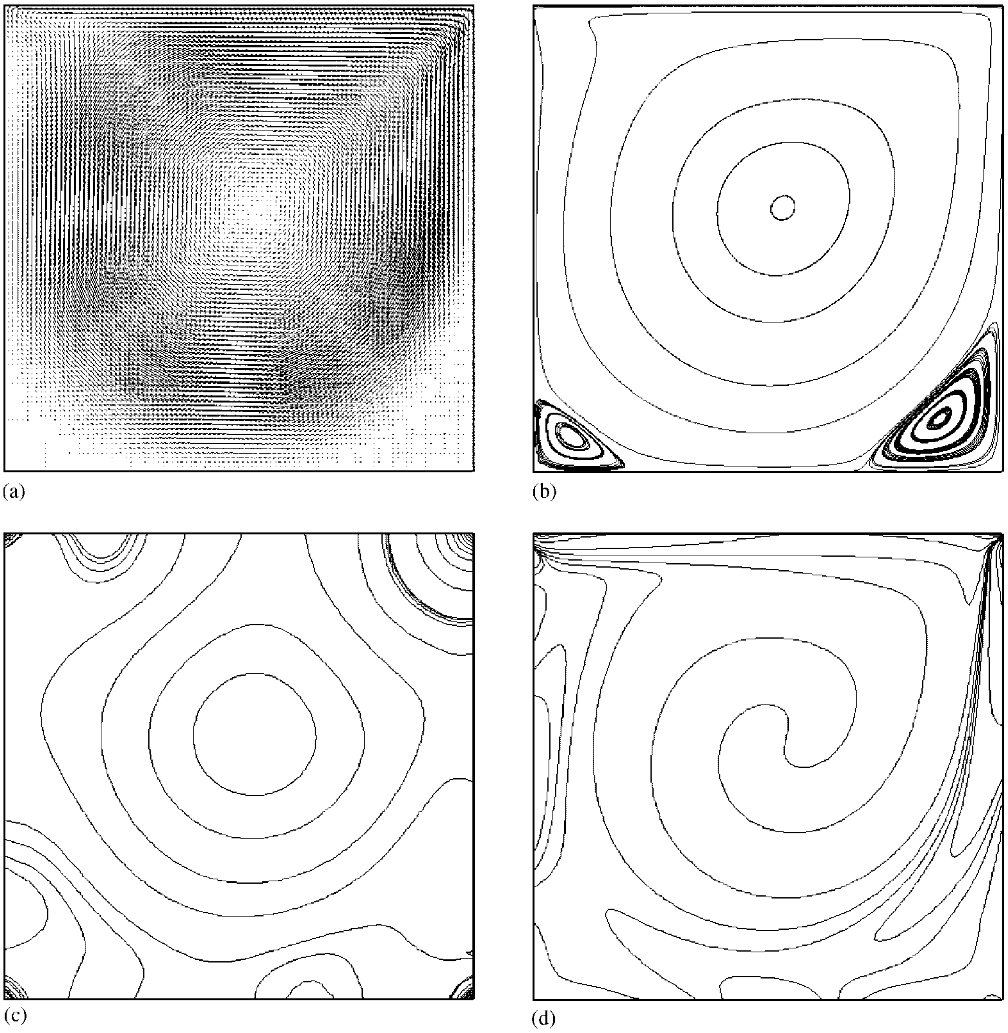

Figure 14. Numerical results for cavity flow at $R_e = 1000$: (a) Velocity; (b) Streamline; (c) Pressure; and (d) Vorticity.

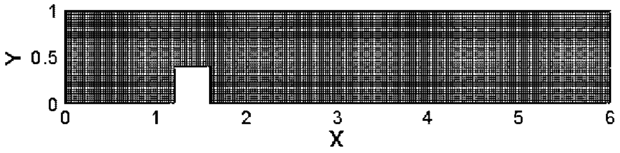

Figure 15. Background cells in time-dependent Navier–Stokes problem.

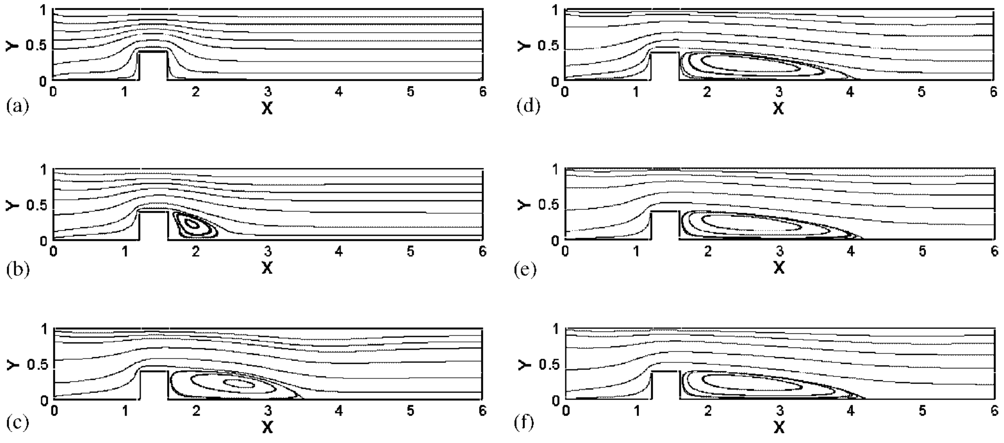

Figure 16. Streamlines at different time steps at $Re=200$ and times: (a) 1.0; (b) 2.0; (c) 5.0; (d) 11.0; (e) 21.0; and (f) 31.0.

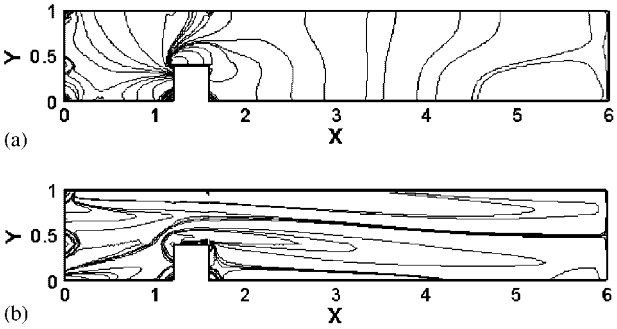

Figure 17. Pressure and voriticty contours at $t=31.0$ for $R_e=200$: (a) Pressure; and (b) Vorticity.

equations are computed in the fully coupled manner. No special treatments, such as upwinding or adjustable parameters are required.

For LSMFM, the numerical results will be affected by the nodal distribution, the size of nodal support and the order of Gauss quadrature in background cell. The effects of inaccurate integration on the accuracy of the solution in LSMFM are quite small. Whether for irregular nodal distribution with quadtree background cells or regular nodal distribution with element-like background cells where the vertices of the cells coincide with the nodes, lower-order Gaussian quadrature has given better accuracy of the solution because it reduces the residual of governing equations much closer to zero. In element-like background cells, when the computation condition for the existence of the solution is satisfied, LSMFM gives excellent results using just one-point Gauss quadrature.

LSMFM preserves the useful meshfree properties. It is easier to generate nodes in preprocessor and implement an adaptive strategy. The adaptive analysis using LSMFM on computational fluid dynamics will be presented in future.

ACKNOWLEDGEMENTS

The postdoctoral fellowship to the first author from Brain Korea 21 Project by the Ministry of Education of Korea is gratefully acknowledged.

REFERENCES

1. Oden JT, Jacquotte OP. Stability of some mixed finite element methods for Stokesian flows. *Computer Methods in Applied Mechanics and Engineering* 1984; **43**(2):231–248.
2. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. Springer-Verlag: Berlin, Heidelberg, 1996.
3. Hughes TJR, Liu WK, Brooks A. Finite element analysis of incompressible viscous flows by the penalty function formulation. *Journal of Computational Physics* 1979; **30**:1–75.
4. Hughes TJR. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1987; **7**:1261–1295.
5. Jiang BN. *The Least-Squares Finite Element Method-Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Springer: Berlin, 1998.
6. Lucy LB. A numerical approach to the testing the fission hypothesis. *Astronautical Journal* 1977; **8**(12): 1013–1024.
7. Gingold RA, Monaghan JJ. Kernal estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics* 1982; **46**:429–453.
8. Liszka T, Orkisz J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures* 1980; **11**:83–95.
9. Kansa EJ. Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics-II Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications* 1990; **19**:147–161.
10. Chung KC. A generalized finite-difference method for heat transfer problems of irregular geometries. *Numerical Heat Transfer* 1981; **4**:345–357.
11. Belyschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
12. Belytschko T, *et al*. Smoothing and accelerated computation in the element free Galerkin method. *Journal of Computational and Applied Mathematic*s 1996; **74**:111–126.
13. Liu WK, Jun S, Li S, Adee J, Belytschko T. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering* 1995; **38**:1655–1679.
14. Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
15. Babuska I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
16. Duarte CA, Oden JT. Hp clouds—a meshless method to solve boundary-value problems. *Technical Report* 95-05, Texas Institute for Computational and Applied Mathematics, Austin, 1995.
17. Duarte CA, Oden JT. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:237–262.
18. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
19. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse element approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
20. Liu WK, Jun S, Sihling DT, Chen YJ, Hao W. Multiresolution reproducing kernel particle method for computational fluid dynamics. *International Journal for Numerical Methods in Fluids* 1997; **24**(12):1391–1415.
21. Sadat H, Couturier S. Performance and accuracy of a meshless method for laminar natural convection. *Numerical Heat Transfer, Part B* 2000; **37**:455–467.
22. Yagawa G, Shirazaki M. Parallel computing for incompressible flow using a nodal-based method. *Computational Mechanics* 1999; **23**:209–217.
23. Cheng M, Liu GR. A novel finite point method for flow simulation. *International Journal for Numerical Methods in Fluids* 2002; **39**:1161–1178.
24. Kim DW, Kim YS. Point collocation methods using the fasting moving least square reproducing kernel approximation. *International Journal for Numerical Methods in Engineering* 2003; **56**(10):1445–1464.
25. Park SH, Youn SK. The least-squares meshfree method. *International Journal for Numerical Methods in Engineering* 2001; **52**:997–1012.
26. Park SH, Kwon KC, Youn SK. A study on the convergence of least-squares meshfree method under inaccurate integration. *International Journal for Numerical Methods in Engineering* 2003; **56**:1397–1419.

27. Park SH, Kwon KC, Youn SK. A posterior error estimates and an adaptive scheme of least-squares meshfree method. *International Journal for Numerical Methods in Engineering* 2003; **58**:1213–1250.
28. Zhang XK, Kwon KC, Youn SK. Least-squares meshfree method for the Stokes problem (submitted).
29. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
30. Krysl P, Belytschko T. ESFLIB: a library to compute the element free Galerkin shape functions. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2181–2205.
31. Tang LQ, Tsang TTH. An efficient least-squares finite element method for incompressible flows and transport processes. *International Journal of Computational Fluid Dynamics* 1995; **4**:21–39.
32. Gorge PL. *Automatic Mesh Generation-Application to Finite Element Methods*. Wiley: Masson, 1991.
33. Berg MD, Kreveld MV, Overmars M, Schwwarzkopf O. *Computational Geometry-Algorithms and Applications*. Springer-Verlag: Berlin, 1997.
34. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:287–411.
35. Tang LQ, Tsang TTH. A least-squares finite element method for time-dependent incompressible flows with thermal convection. *International Journal for Numerical Methods in Fluids* 1993; **17**:271–289.
36. Leone JM, Gresho PM. Finite element simulations of steady, two-dimensional, viscous incompressible flow over a step. *Journal of Computational Physics* 1981; **41**:167–191.
37. Laval H, Quartapelle L. A fractional-step Taylor–Galerkin method for unsteady incompressible flows. *International Journal for Numerical Methods in Fluids* 1990; **11**:501–513.