Journal of the Egyptian Mathematical Society (2014) 22, 501–510

Egyptian Mathematical Society

Journal of the Egyptian Mathematical Society

www.etms-eg.org
www.elsevier.com/locate/joems

The meshless local Petrov-Galerkin method for simulating unsteady incompressible fluid flow ☆

C. Sataprahm [a], A. Luadsong [a,b,*]

[a] Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok 10140, Thailand
[b] Centre of Excellence in Mathematics, CHE, Si Ayutthaya, Bangkok 10400, Thailand

Received 22 February 2013; revised 14 March 2013; accepted 8 October 2013
Available online 5 December 2013

KEYWORDS

MLS;
MLPG;
Improved Gaussian's weight;
Unsteady incompressible fluid flow

Abstract This article presents a numerical algorithm using the Meshless Local Petrov-Galerkin (MLPG) method for the incompressible Navier–Stokes equations. To deal with time derivatives, the forward time differences are employed yielding the Poisson's equation. The MLPG method with the moving least-square (MLS) approximation for trial function is chosen to solve the Poisson's equation. In numerical examples, the local symmetric weak form (LSWF) and the local unsymmetric weak form (LUSWF) with a classical Gaussian weight and an improved Gaussian weight on both regular and irregular nodes are demonstrated. It is found that LSWF1 with a classical Gaussian weight order 2 gives the most accurate result.

1. Introduction

Incompressible Navier–Stokes flow in two dimensions is one of several major problems in fluid mechanics that have been extensively studied both theoretically and numerically. In general, the formulation of primitive variables is popularly employed for the incompressible Navier–Stokes equation but it has a limitation in approximating the velocity and the pressure. The finite volume method (FVM) and finite element method (FEM) have been widely applied to solve the incompressible Navier–Stokes flow problems. However, it is well-known that these methods depend strongly on the mesh properties. In computing problems with irregular complex geometries using these methods, mesh generation is a far more time-consuming and expensive task than solution of the partial differential equations (PDEs), particularly in three dimensional (3D) cases. To overcome such a problem, meshless methods, a new numerical method class have been developed.

Meshless methods were established with the objective of eliminating the requirement of mesh generation step, which is time-consuming and burdensome, in FEM. Owing to these

* Corresponding author at: King Mongkut's University of Technology Thonburi (KMUTT), Bangkok 10140, Thailand. Tel.: +66 24708837.
E-mail addresses: chon52501409@gmail.com (C. Sataprahm), anirut.lua@kmutt.ac.th (A. Luadsong).

Peer review under responsibility of Egyptian Mathematical Society.

ELSEVIER Production and hosting by Elsevier

☆ The research is financed by: This research is partially supported by the Centre of Excellence in Mathematics, the Commission on Higher Education, Thailand.

http://dx.doi.org/10.1016/j.joems.2013.10.002

reasons, meshless methods have received much attention as a number of meshless methods have been introduced by different authors. These include smooth particle hydrodynamics (SPH) [1,2], diffuse element method (DEM) [3], element-free Galerkin (EFG) [4], reproducing kernel particle method (RKPM) [5], finite point method (FPM) [6], partition of unity method (PU) [7], boundary node method (BNM) [8], local boundary integral equation (LBIE) [9], meshless local Petrov-Galerkin method (MLPG) [10], meshless regular local boundary integral equation (MRLBIE) [11], finite cloud method (FCM) [12], point interpolation method (PIM) [13], least-squares collocation meshless method (LSCM) [14], etc. The meshless local Petrov-Galerkin (MLPG) method is a truly meshless method, which requires no elements or background cells, for either the interpolation or the integration purposes. The concept of MLPG was first proposed by Atluri and Zhu [10], and later discussed in depth in Atluri and Shen [15]. The most significant difference between this method and the finite element method or any other meshless method is that the local weak forms are generated on overlapping local sub-domains, instead of using the global weak form. Integration of the weak form is performed in local sub-domains with simple geometrical shapes, therefore no elements or background cells are necessary either for interpolation purposes or for integration purposes. The MLPG approach is also different from the truly meshless method based on the local boundary integral equation (LBIE) method, because there are no singular integrals in the MLPG method. This method is characterized as meshless since distributed nodal points, covering the domain of interest, are employed.

Remarkable successes of the MLPG method in computational mechanics have been reported in recent years. The first article applying MLPG method to compute convection-diffusion and incompressible flow problems was by Lin and Atluri [16]. In their work, two kinds of upwind schemes were constructed to overcome oscillations produced by convection term. They applied upwind schemes to solve the incompressible flow problem based on the primitive variable formulation and added the perturbation term to continuity equation to satisfy the Babŭka-Brezzi condition. But when these schemes were applied to compute the high Reynolds number problems, the parameter of perturbation term was difficult to determine and it also suffered from the convergent difficulty. Wu et al. [17] applied MLPG to solve incompressible flow problems with vorticity-stream function method without addressing the stability problem. One year later, they applied MLPG to solve two-dimensional (2D) incompressible fluid flow and heat transfer problems with benchmark solutions. The streamline upwind Petrov-Galerkin method is applied to overcome oscillation velocity field and mixed formulation is employed to satisfy the Babŭka-Brezzi condition. The results show that SUPG method gives a convergent solution for high Reynolds number. Sanyasiraju and Chandhini [18] developed a local RBF gridfree scheme to solve unsteady incompressible Navier–Stokes equations for primitive variables. This novel fractional step algorithm has been proposed to achieve velocity-pressure decoupling, in which it has been validated over various problems.

In the present paper, the meshless local Petrov-Galerkin method with MLS interpolation scheme is applied to develop an algorithm for solving the unsteady incompressible Navier–Stokes flow problem.

2. The moving least-square (MLS) approximation for trial function

The moving least-square (MLS) is one of these interpolation schemes with a reasonable accuracy. Consider a sub-domain $\Omega_{\mathbf{x}}$, which is defined as the neighborhood of a point $\mathbf{x}$ and denoted as domain of definition of MLS approximation for the trial function at point $\mathbf{x}$. To approximate the distribution of function $u^n(\mathbf{x}) = u(\mathbf{x}, t_n)$ in $\Omega_{\mathbf{x}}$, over a number of randomly located nodes $\mathbf{x}_i$, $i = 1, 2, \ldots, N$. The moving least-squares (MLS) approximation $u_h^n(\mathbf{x})$ of u^n, $\forall \mathbf{x} \in \Omega_{\mathbf{x}}$, can be defined by

$$u_h^n = \mathbf{p}^T(\mathbf{x})\mathbf{a}^n(\mathbf{x}), \qquad \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{1}$$

where $\mathbf{p}(\mathbf{x})$ is a vector of basis function

$$\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})],$$

where m is the number of the basis functions. Usually the complete monomial basis is used to ensure the consistency of the approximations, whereby different types of the polynomials may be used. Depending on the problem, other type of functions may also be employed in order to enhance the solutions. For a two-dimensional (2D) case used in this paper, the complete monomial basis are defined as follows:

- Linear basis

$$\mathbf{p}^T(\mathbf{x}) = [1,\ x,\ y],$$

- Quadratic basis

$$\mathbf{p}^T(\mathbf{x}) = [1,\ x,\ y,\ x^2,\ xy,\ y^2],$$

where $\mathbf{x} = (x, y) \in \mathbb{R}^2$ and the term of the complete 2D basis may be obtained by employing the Pascal triangle. For the polynomial basis, the total number of terms is related to the order of the basis by expression $m = \frac{(l+1)(l+2)}{2}$ with l as the order of the basis. The vector $\mathbf{a}^n(\mathbf{x})$ contains the unknown coefficients

$$\mathbf{a}^n(\mathbf{x}) = [a_1^n(\mathbf{x}),\ a_2^n(\mathbf{x}),\ a_3^n(\mathbf{x}),\ \ldots,\ a_m^n(\mathbf{x})]^T,$$

which are the functions of $\mathbf{x}$, i.e. they have to be calculated for each point $\mathbf{x}$. The vector $\mathbf{a}^n(\mathbf{x})$ is determined by means of the discrete weighted L_2 norm, defined as follows:

$$J(\mathbf{a}^n(\mathbf{x})) = \sum_{i=1}^{N} w_i(\mathbf{x})[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}^n(\mathbf{x}) - \hat{u}_i^n]^2, \tag{2}$$

where $w_i(\mathbf{x})$ is a weight function associated with the node i, $w_i(\mathbf{x}) > 0$ for all $\mathbf{x}$ in the support of $w_i(\mathbf{x})$, $\mathbf{x}_i$ denotes the values of $\mathbf{x}$ at node i, N is a number of nodes in $\Omega_{\mathbf{x}}$ for which $w_i(\mathbf{x}) > 0$. Here it should be noted that $\hat{u}_i^n$, $i = 1, 2, \ldots, N$ in Eq. (2) are the fictitious nodes and not the actual nodes of unknown trial function $u_h^n(\mathbf{x})$. The minimization of $J(\mathbf{a}^n(\mathbf{x}))$ leads to the following system of equations

$$A(\mathbf{x})\mathbf{a}^n(\mathbf{x}) = B(\mathbf{x})\hat{\mathbf{u}}^n, \tag{3}$$

where,

$$A = \sum_{i=1}^{N} w_i(\mathbf{x})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i)$$

the matrix B is defined as

$$B = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1),\ w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2),\ w_3(\mathbf{x})\mathbf{p}(\mathbf{x}_3),\ \ldots,\ w_N(\mathbf{x})\mathbf{p}(\mathbf{x}_N)],$$

and the vector $\hat{\mathbf{u}}^n$ contains the fictitious values at the nodes $\mathbf{x}_i$

$$\hat{\mathbf{u}}^n = \left[\hat{u}_1^n,\ \hat{u}_2^n,\ \hat{u}_3^n,\ \ldots,\ \hat{u}_N^n\right]^T.$$

The coefficients $\mathbf{a}(\mathbf{x})$ are evaluated by solving the Eq. (3), leading to

$$\mathbf{a}^n(\mathbf{x}) = A^{-1}(\mathbf{x})B(\mathbf{x})\hat{\mathbf{u}}^n. \tag{4}$$

Substituting Eq. (4) into Eq. (1), the MLS approximation is obtained, may be written as follows:

$$u_h^n(\mathbf{x}) = \sum_{i=1}^{N}\phi_i(\mathbf{x})\hat{u}_i^n = \Phi(\mathbf{x})\hat{\mathbf{u}}^n, \qquad \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{5}$$

where $\Phi(\mathbf{x})$ is the vector of MLS shape functions corresponding to N nodes in the support domain of the point x, can be written as

$$\Phi(\mathbf{x}) = [\phi_1,\ \phi_2,\ \phi_3,\ \ldots,\ \phi_N]^T,$$

and $\phi_i(\mathbf{x})$ is the shape function associated with the node $\mathbf{x}_i$, which is calculated as

$$\phi_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\left[A^{-1}(\mathbf{x}).B(\mathbf{x})\right]_{ji} \tag{6}$$

The derivatives of the shape function $\phi_i(\mathbf{x})$ may be obtained by direct differentiation of Eq. (6) as

$$\phi_{i,k} = \sum_{j=1}^{m}[p_{j,k}(A^{-1}B)_{ji} + p_j\left(A^{-1}B_{,k} + A_{,k}^{-1}B\right)_{ji}], \tag{7}$$

where $A_{,k}^{-1} = (A^{-1})_{,k}$ represents the derivative of the inverse of A with respect to $x_k = x$ (or y), which is given by

$$A_{,k}^{-1} = -A^{-1}A_{,k}A^{-1}. \tag{8}$$

The second partial derivatives of $\phi_i(\mathbf{x})$ are obtained as

$$\begin{aligned}\phi_{i,kl} = \sum_{j=1}^{m}\Big[&p_{j,kl}(A^{-1}B)_{ji} + p_{j,k}(A^{-1}B_{,l} + A_{,l}^{-1}B)_{ji}\\ &+p_{j,l}(A^{-1}B_{,k} + A_{,k}^{-1}B)_{ji}\\ &+p_j(A^{-1}B_{,kl} + A_{,kl}^{-1}B + A_{,l}^{-1}B_{,k} + A_{,k}^{-1}B_{,l})_{ji}\Big],\end{aligned} \tag{9}$$

with

$$A_{,kl}^{-1} = -A^{-1}A_{,l}A^{-1}A_{,k}A^{-1} - A^{-1}A_{,kl}A^{-1} + A^{-1}A_{,k}A^{-1}A_{,l}A^{-1}. \tag{10}$$

The MLS approximation is well defined, only when the matrix in Eq. (3) is non-singular. From Eqs. (2) and (6), it may be seen that $\phi_i(\mathbf{x}) = 0$ when $w_i(\mathbf{x}) = 0$. The fact that $\phi_i(\mathbf{x})$ vanishes, for $\mathbf{x}$ not in support of node $\mathbf{x}_i$ preserves the local character of the moving least-squares approximation. It is known that the smoothness of the shape functions $\phi(\mathbf{x})$ is determined by that of the basis functions and of the weight functions. Let $C^k(\Omega)$ be the space of kth continuously differentiable functions. If $w_i(\mathbf{x}) \in C^k(\Omega)$, $i = 1, 2, \ldots, N$ and $p_j(\mathbf{x}) \in C^l(\Omega)$, $j = 1, 2, \ldots, m$, then $\phi(\mathbf{x}) \in C^r(\Omega)$ with $r = \min(k, l)$. A number of choices are available for the basis functions and the weight functions. In this paper, the quadratic basis is chosen and a Gaussian weight function is used,

$$w_i(\mathbf{x}) = \begin{cases}\frac{\exp[-(d_i/c_i)^{2k}]-\exp[-(r_i/c_i)^{2k}]}{1-\exp[-(r_i/c_i)^{2k}]}, & 0 \leqslant d_i \leqslant r_i\\ 0, & d_i > r_i\end{cases}, \tag{11}$$

where $d_i = \|\mathbf{x} - \mathbf{x}^i\|$, c_i and k are constant controlling the shape of the weight function $w_i(\mathbf{x})$ and r_i is the size of the support domain. The size of the support domain should be large to have sufficient number of nodes covered in the domain of definition of every sample point ($n \geqslant m$) to ensure the regularity of the matrix A. A Gaussian weight function has a specific characteristic where the relative weights can be controlled by manipulating the constant c_i. When c_i decreases, higher weights are obtained on points $\mathbf{x}_i$ which stay close to $\mathbf{x}$ and lower weights on points that are far will be removed from $\mathbf{x}$ and vice versa. If the weight function $w_i(\mathbf{x})$ is continuous together with its first derivatives, the shape function $\phi_i(\mathbf{x})$ will be continuous along with its first derivatives. The exponential weight function has unlimited continuity. The first derivatives of a Gaussian weight function $w_i(\mathbf{x})$ can be calculated as

$$w_{i,x}(\mathbf{x}) = \begin{cases}\frac{-2k(\mathbf{x}-\mathbf{x}_i)d_i^{2k-2}\exp[-(d_i/c_i)^{2k}]}{c_i^{2k}(1-\exp[-(r_i/c_i)^{2k}])}, & 0 \leqslant d_i \leqslant r_i\\ 0, & d_i > r_i\end{cases}, \tag{12}$$

and,

$$w_{i,y}(\mathbf{x}) = \begin{cases}\frac{-2k(\mathbf{y}-\mathbf{y}_i)d_i^{2k-2}\exp[-(d_i/c_i)^{2k}]}{c_i^{2k}(1-\exp[-(r_i/c_i)^{2k}])}, & 0 \leqslant d_i \leqslant r_i\\ 0, & d_i > r_i\end{cases}. \tag{13}$$

For the case that d_i is equal to zero, the first derivatives of $w_i(\mathbf{x})$ may be written as

$$w_{i,x}(\mathbf{x}) = w_{i,y}(\mathbf{x}) = 0. \tag{14}$$

Similarly, the second derivatives of the weight function can be computed as

$$w_{i,xx}(\mathbf{x}) = \begin{cases}\frac{-2kd_i^{2k-4}\exp[-(d_i/c_i)^{2k}]}{c_i^{2k}(1-\exp[-(r_i/c_i)^{2k}])}\left[\frac{-2k(x-x_i)^2d_i^{2k}}{c_i^{2k}} + (2k-2)(x-x_i)^2 + d_i^2\right], & d_i \leqslant r_i\\ 0, & d_i > r_i\end{cases}, \tag{15}$$

and,

$$w_{i,yy}(\mathbf{x}) = \begin{cases}\frac{-2kd_i^{2k-4}\exp[-(d_i/c_i)^{2k}]}{c_i^{2k}(1-\exp[-(r_i/c_i)^{2k}])}\left[\frac{-2k(y-y_i)^2d_i^{2k}}{c_i^{2k}} + (2k-2)(y-y_i)^2 + d_i^2\right], & d_i \leqslant r_i\\ 0, & d_i > r_i\end{cases}. \tag{16}$$

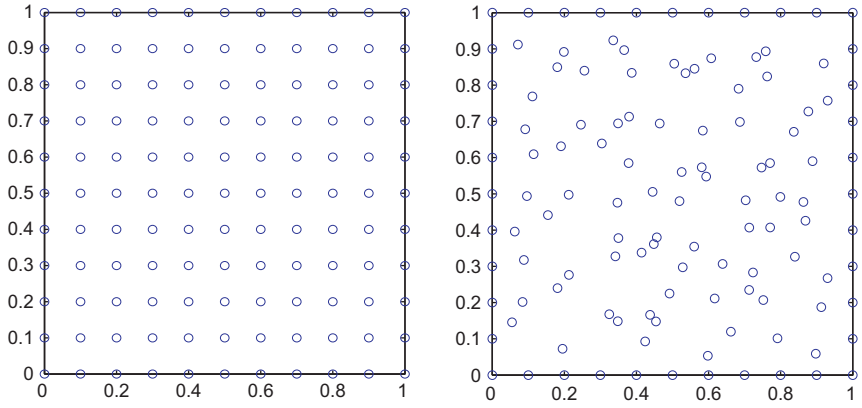

Fig. 1 Node distribution for Example 1 (11×11 nodes).

Likewise, for the case that d_i is equal to zero, the second derivatives of $w_i(\mathbf{x})$ may be written as

$$w_{i,xx}(\mathbf{x}) = w_{i,yy}(\mathbf{x}) = \frac{-2k\exp[-(d_i/c_i)^{2k}]}{c_i^{2k}(1-\exp[-(r_i/c_i)^{2k}])}. \tag{17}$$

The idea of all these derivatives can be found in [19].

3. Problem formulation

The governing equations for unsteady incompressible viscous fluid in the square domain $\Omega = [0,\ 1]\times[0,\ 1]$ are the two-dimensional Navier–Stokes equation together with the continuity equation in the convection term, i.e., non-conservative form. These equations can be written as

$$\frac{\partial u}{\partial t} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} - \frac{\partial p}{\partial x} + f_x, \tag{18}$$

$$\frac{\partial v}{\partial t} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} - \frac{\partial p}{\partial y} + f_y, \tag{19}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{20}$$

where u and v are the velocities in x and y direction respectively, p is the pressure, f_x and f_y are the body force, Re is the Reynolds number. Eqs. (18) and (19) are the momentum equation and Eq. (20) is the continuity equation. The boundary conditions can be assumed to be:

$$u = \bar{u}, \quad v = \bar{v}, \quad p = \bar{p} \qquad \text{on} \qquad \Gamma_u, \tag{21}$$

$$\frac{\partial u}{\partial n} \equiv q_u = \bar{q}_u, \quad \frac{\partial v}{\partial n} \equiv q_v = \bar{q}_v, \quad \frac{\partial p}{\partial n} \equiv q_p = \bar{q}_p \quad \text{on} \quad \Gamma_q, \tag{22}$$

where $\bar{u}$, $\bar{v}$, $\bar{p}$, $\bar{q}_u$, $\bar{q}_v$, and $\bar{q}_p$ are the prescribed potential and normal flux. Γ_u and Γ_q are subset of Γ satisfying $\Gamma_u \cap \Gamma_q = \phi$ and $\Gamma_u \cup \Gamma_q = \Gamma$.

4. Discretization of time derivatives and the algorithm

To deal with the time derivatives, a time stepping method is employed. For this paper the following approximations are written as

$$\frac{\partial u}{\partial t}(\mathbf{x}, t_n) \simeq \frac{u^{n+1}(\mathbf{x}) - u^n(\mathbf{x})}{\Delta t},$$

$$\frac{\partial v}{\partial t}(\mathbf{x}, t_n) \simeq \frac{v^{n+1}(\mathbf{x}) - v^n(\mathbf{x})}{\Delta t},$$

where $\mathbf{x} = (x, y)^T$, $u^n = u(\mathbf{x}, t_n)$ and $v^{(n)} = v(\mathbf{x}, t_n)$.Discretizing Eqs. (18) and (19) at times level n, yielding,

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{Re}\left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}\right) - u\frac{\partial u^n}{\partial x} - v\frac{\partial u^n}{\partial y} - \frac{\partial p^n}{\partial x} + f_x^n,$$

$$u^{n+1} = u^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}\right) - u\frac{\partial u^n}{\partial x} - v\frac{\partial u^n}{\partial y} + f_x^n\right] - \Delta t\frac{\partial p^n}{\partial x}.$$

Let,

$$F^n = u^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}\right) - u\frac{\partial u^n}{\partial x} - v\frac{\partial u^n}{\partial y} + f_x^n\right]. \tag{23}$$

Therefore,

$$u^{n+1} = F^n - \Delta t\frac{\partial p^n}{\partial x}, \tag{24}$$

and

$$\frac{v^{n+1} - v^n}{\Delta t} = \frac{1}{Re}\left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2}\right) - u\frac{\partial v^n}{\partial x} - v\frac{\partial v^n}{\partial y} - \frac{\partial p^n}{\partial y} + f_y^n,$$

$$v^{n+1} = v^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2}\right) - u\frac{\partial v^n}{\partial x} - v\frac{\partial v^n}{\partial y} + f_y^n\right] - \Delta t\frac{\partial p^n}{\partial y}.$$

Let,

$$G^n = v^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2}\right) - u\frac{\partial v^n}{\partial x} - v\frac{\partial v^n}{\partial y} + f_y^n\right], \tag{25}$$

therefore,

$$v^{n+1} = G^n - \Delta t\frac{\partial p^n}{\partial y}. \tag{26}$$

From Eqs. (24) and (26), we obtain

$$\frac{\partial u^{n+1}}{\partial x} = \frac{\partial F^n}{\partial x} - \Delta t\frac{\partial^2 p^n}{\partial x^2}, \tag{27}$$

$$\frac{\partial v^{n+1}}{\partial y} = \frac{\partial G^n}{\partial y} - \Delta t\frac{\partial^2 p^n}{\partial y^2}. \tag{28}$$

Substitution of Eqs. (27) and (28) into Eq. (20), the equation can be written as

$$\frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} = 0, \tag{29}$$

and,

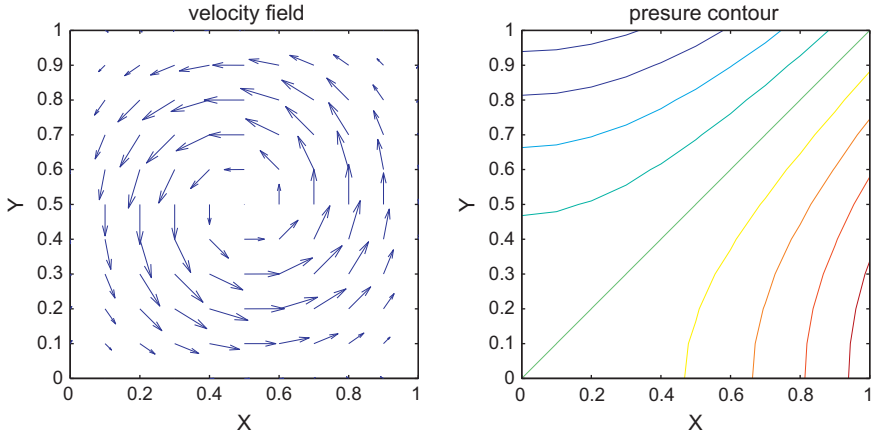

Fig. 2 The numerical solution of velocities and pressure at $\Delta t = 0.10$ of LSWF1 on regular nodes in Example 1.

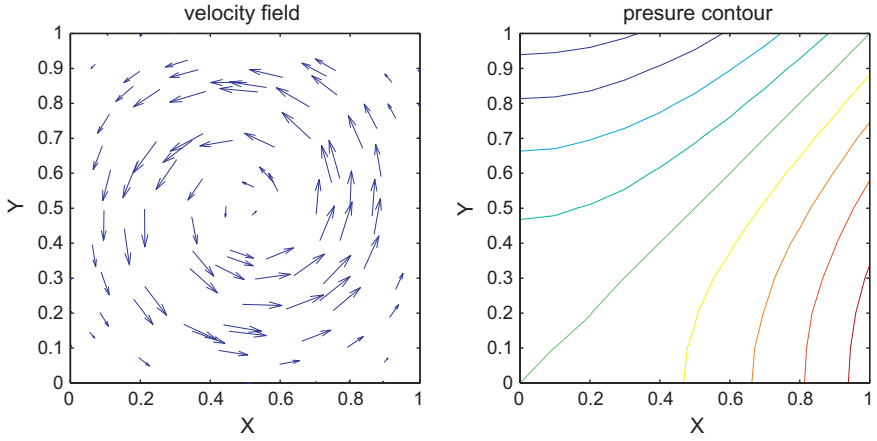

Fig. 3 The numerical solution of velocities and pressure at $\Delta t = 0.10$ of LSWF1 on irregular nodes in Example 1.

$$\frac{\partial^2 p^n}{\partial x^2}+\frac{\partial^2 p^n}{\partial y^2}=\frac{1}{\Delta t}\left(\frac{\partial F^n}{\partial x}+\frac{\partial G^n}{\partial y}\right). \tag{30}$$

Eq. (30) is the Poisson's equation with non-zero source term. Numerical implementation of the procedure described from Eqs. (26)–(30) can be summarized as follows.

1. Compute intermediate velocity field using

$$F^n = u^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u^n}{\partial x^2}+\frac{\partial^2 u^n}{\partial y^2}\right)-u\frac{\partial u^n}{\partial x}-v\frac{\partial u^n}{\partial y}+f_x^n\right],$$

$$G^n = v^n + \Delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v^n}{\partial x^2}+\frac{\partial^2 v^n}{\partial y^2}\right)-u\frac{\partial v^n}{\partial x}-v\frac{\partial v^n}{\partial y}+f_y^n\right].$$

2. Solve the pressure Poisson's equation,

$$\frac{\partial^2 p^n}{\partial x^2}+\frac{\partial^2 p^n}{\partial y^2}=\frac{1}{\Delta t}\left(\frac{\partial F^n}{\partial x}+\frac{\partial G^n}{\partial y}\right),$$

with boundary conditions $p^n|_{\Gamma_u}=\bar{p}^n$ and $\frac{\partial p^n}{\partial n}\big|_{\Gamma_q}=\bar{q}^n$.

3. Update the velocity field to $(n+1)$th time level using

$$u^{n+1} = F^n - \Delta t\frac{\partial p^n}{\partial x},$$

$$v^{n+1} = G^n - \Delta t\frac{\partial p^n}{\partial y}.$$

5. The MLPG method and the local weak forms

In the present, the meshless local Petrov-Galerkin is constructed over a local sub-domain Ω_s which is located inside the global domain Ω. The local sub-domain Ω_s is a taken to be either circle or a part of a circle.

A generalized local weak form Eq. (30) over a local sub-domain Ω_s, can be written as

$$\int_{\Omega_s}\left[\left(\frac{\partial^2 p^n}{\partial x^2}+\frac{\partial^2 p^n}{\partial y^2}\right)-\frac{1}{\Delta t}\left(\frac{\partial F^n}{\partial x}+\frac{\partial G^n}{\partial y}\right)\right]w d\Omega - \alpha\int_{\Gamma_{su}}(p^n-\bar{p}^n)w d\Gamma-\alpha\int_{\Gamma_{sq}}\left(\frac{\partial p^n}{\partial n}-\bar{q}^n\right)w d\Gamma = 0 \tag{31}$$

where p is the trial function, w is the test function and Γ_{su} is a part of the boundary $\partial\Omega_s$ of Ω_s, over which the essential boundary conditions are specified. In general, $\partial\Omega_s = \Gamma_s \cup L_s$, with Γ_s is a part of the local boundary located on the global boundary and L_s is the other part of the local boundary over which no boundary conditions are specified, i.e., $\Gamma_s = \partial\Omega_s \cap \Gamma$ with $\Gamma_s = \partial\Omega_s - L_s$. In Eq. (31), α is a penalty parameter, $\alpha \gg 1$ is used to impose the essential and natural boundary conditions. In this paper, the value of $\alpha = 10^{12}$ gives good results.

Table 1 Relative errors of velocities and pressure for each time step of LUSWF on regular nodes in Example 1.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	4.2592×10^{-5}	4.2590×10^{-5}	3.4783×10^{-7}
0.02	8.5312×10^{-5}	8.5309×10^{-5}	3.4435×10^{-7}
0.03	1.2764×10^{-4}	1.2764×10^{-4}	3.5868×10^{-7}
0.04	1.6959×10^{-4}	1.6958×10^{-5}	3.4960×10^{-7}
0.05	2.1115×10^{-4}	2.1115×10^{-4}	3.4503×10^{-7}
0.06	2.5234×10^{-4}	2.5233×10^{-4}	3.4535×10^{-7}
0.07	1.4623×10^{-4}	1.4628×10^{-4}	1.5057×10^{-7}
0.08	2.9315×10^{-4}	2.9314×10^{-4}	3.4568×10^{-7}
0.09	3.3359×10^{-4}	3.3358×10^{-4}	3.4326×10^{-7}
0.10	3.7367×10^{-4}	3.7366×10^{-4}	3.3986×10^{-7}

Table 2 Relative errors of velocities and pressure for each time step of LUSWF on irregular nodes in Example 1.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	1.4190×10^{-5}	2.1539×10^{-5}	1.8264×10^{-6}
0.02	2.8169×10^{-5}	4.2860×10^{-5}	1.8082×10^{-6}
0.03	4.2023×10^{-5}	6.3985×10^{-5}	1.7897×10^{-6}
0.04	5.5753×10^{-5}	8.4917×10^{-5}	1.7600×10^{-6}
0.05	6.9360×10^{-5}	1.0566×10^{-4}	1.7542×10^{-6}
0.06	8.2843×10^{-5}	1.2621×10^{-4}	1.7641×10^{-6}
0.07	9.6207×10^{-5}	1.4657×10^{-4}	1.7271×10^{-6}
0.08	1.0945×10^{-4}	1.6675×10^{-4}	1.7110×10^{-6}
0.09	1.2258×10^{-4}	1.8674×10^{-4}	1.6889×10^{-6}
0.10	1.3558×10^{-4}	2.0655×10^{-4}	1.6819×10^{-6}

Table 3 Relative errors of velocities and pressure for each time step of LSWF on regular nodes in Example 1.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	4.2588×10^{-5}	4.2586×10^{-5}	3.3424×10^{-7}
0.02	8.5306×10^{-5}	8.5303×10^{-5}	3.3465×10^{-7}
0.03	1.2763×10^{-4}	1.2763×10^{-4}	3.4284×10^{-7}
0.04	1.6958×10^{-4}	1.6958×10^{-4}	3.3944×10^{-7}
0.05	2.1114×10^{-4}	2.1114×10^{-4}	3.3528×10^{-7}
0.06	2.5233×10^{-4}	2.5232×10^{-4}	3.2452×10^{-7}
0.07	2.9313×10^{-4}	2.9313×10^{-4}	3.1736×10^{-7}
0.08	3.3357×10^{-4}	3.3357×10^{-4}	3.1902×10^{-7}
0.09	3.7364×10^{-4}	3.7364×10^{-4}	3.1695×10^{-7}
0.10	4.1335×10^{-4}	4.1334×10^{-4}	3.1457×10^{-7}

Table 4 Relative errors of velocities and pressure for each time step of LSWF on irregular nodes in Example 1.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	1.4139×10^{-5}	2.1700×10^{-5}	2.7453×10^{-6}
0.02	2.8094×10^{-5}	4.3101×10^{-5}	2.7201×10^{-6}
0.03	4.1923×10^{-5}	6.4304×10^{-5}	2.6885×10^{-6}
0.04	5.5629×10^{-5}	8.5313×10^{-5}	2.6658×10^{-6}
0.05	6.9211×10^{-5}	1.0613×10^{-4}	2.6377×10^{-6}
0.06	8.2672×10^{-5}	1.2676×10^{-4}	2.6139×10^{-6}
0.07	9.6012×10^{-5}	1.4619×10^{-4}	2.5878×10^{-6}
0.08	1.0923×10^{-4}	1.6744×10^{-4}	2.5622×10^{-6}
0.09	1.2233×10^{-4}	1.8751×10^{-4}	2.5411×10^{-6}
0.10	1.3532×10^{-4}	2.0739×10^{-4}	2.5122×10^{-6}

Using $(\nabla^2 p)w = \left[\frac{\partial}{\partial x}\left(\frac{\partial p}{\partial x}w\right) + \frac{\partial}{\partial y}\left(\frac{\partial p}{\partial y}w\right)\right] - \left(\frac{\partial p}{\partial x}\frac{\partial w}{\partial x} + \frac{\partial p}{\partial y}\frac{\partial w}{\partial y}\right)$ and the divergence theorem in Eq. (31) leads to

$$\begin{aligned}
&\int_{\Omega_s}\left(\frac{\partial p^n}{\partial x}\frac{\partial w}{\partial x}+\frac{\partial p^n}{\partial y}\frac{\partial w}{\partial y}\right)d\Omega-\int_{\partial\Omega_s}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad+\alpha\int_{\Gamma_{su}}p^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad=\alpha\int_{\Gamma_{su}}\bar{p}^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\bar{q}^n wd\Gamma-\frac{1}{\Delta t}\\
&\qquad\times\int_{\Omega_s}\left(\frac{\partial F^n}{\partial x}+\frac{\partial G^n}{\partial y}\right)wd\Gamma.
\end{aligned}\tag{32}$$

Similarly, Eq. (32) has changed to

$$\begin{aligned}
&\int_{\Omega_s}\left(\frac{\partial p^n}{\partial x}\frac{\partial w}{\partial x}+\frac{\partial p^n}{\partial y}\frac{\partial w}{\partial y}\right)d\Omega-\int_{\partial\Omega_s}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad+\alpha\int_{\Gamma_{su}}p^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad=\alpha\int_{\Gamma_{su}}\bar{p}^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\bar{q}^n wd\Gamma-\frac{1}{\Delta t}\\
&\qquad\times\int_{\partial\Omega_s}(F^n n_1 w+G^n n_2 w)d\Gamma+\frac{1}{\Delta t}\\
&\qquad\times\int_{\Omega_s}\left(F^n\frac{\partial w}{\partial x}+G^n\frac{\partial w}{\partial y}\right)d\Omega,
\end{aligned}\tag{33}$$

where n_1 and n_2 are components of outward unit normal vector to the boundary $\partial\Omega_s$ which is usually composed of three parts; the internal boundary L_s, the boundaries, Γ_{su} and Γ_{sq} over which the Dirichlet and Neumann conditions are applied respectively. If there is no intersection between $\partial\Omega_s$ and the global boundary Γ, $\partial\Omega_s = L_s$. In the MLPG method the trial functions and the test functions are not necessarily from the same functional spaces. To simplify Eq. (33), we can deliberately select the test functions w such that they vanish over $\partial\Omega_s$ except when $\partial\Omega_s$ intersects with the global boundary Γ, we obtain the following local weak form, as

$$\begin{aligned}
&\int_{\Omega_s}\left(\frac{\partial p^n}{\partial x}\frac{\partial w}{\partial x}+\frac{\partial p^n}{\partial y}\frac{\partial w}{\partial y}\right)d\Omega-\int_{\Gamma_{su}}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad+\alpha\int_{\Gamma_{su}}p^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\frac{\partial p^n}{\partial n}wd\Gamma\\
&\quad=\alpha\int_{\Gamma_{su}}\bar{p}^n wd\Gamma+\alpha\int_{\Gamma_{sq}}\bar{q}wd\Gamma+\int_{\Gamma_{sq}}\bar{q}wd\Gamma-\frac{1}{\Delta t}\\
&\qquad\times\int_{\partial\Omega_s}(F^n n_1 w+G^n n_2 w)d\Gamma+\frac{1}{\Delta t}\\
&\qquad\times\int_{\Omega_s}\left(F^n\frac{\partial w}{\partial x}+G^n\frac{\partial w}{\partial y}\right)d\Omega.
\end{aligned}\tag{34}$$

Because, the trial functions p_h within the sub-domain Ω_s, in the MLS approximation, is determined by the fictitious nodal values $\hat{p}_i$, within the *domain of definition* for all points $\mathbf{x}$ falling

Table 5 Relative errors of velocities and pressure for each time step of LSWF on regular nodes in Example 1.

Time (t)	An improve Gaussian weight function		
	e_u	e_v	e_p
0.01	3.1488×10^{-5}	3.1485×10^{-5}	2.9851×10^{-7}
0.02	6.3175×10^{-5}	6.3169×10^{-5}	2.9361×10^{-7}
0.03	9.4546×10^{-5}	9.4538×10^{-5}	2.9129×10^{-7}
0.04	1.2560×10^{-4}	1.2559×10^{-4}	2.9129×10^{-7}
0.05	4.5635×10^{-4}	1.5634×10^{-4}	2.8657×10^{-7}
0.06	1.8680×10^{-4}	1.8678×10^{-4}	2.8357×10^{-7}
0.07	2.1694×10^{-4}	2.1692×10^{-4}	2.7895×10^{-7}
0.08	2.4678×10^{-4}	2.4676×10^{-4}	2.7719×10^{-7}
0.09	2.7632×10^{-4}	2.7630×10^{-4}	2.7315×10^{-7}
0.10	3.0556×10^{-4}	3.0554×10^{-4}	2.7653×10^{-7}

Table 6 Relative errors of velocities and pressure for each time step of LSWF on irregular nodes in Example 1.

Time (t)	An improve Gaussian weight function		
	e_u	e_v	e_p
0.01	1.0561×10^{-5}	1.3191×10^{-5}	2.2247×10^{-6}
0.02	2.0783×10^{-5}	2.6326×10^{-5}	2.2036×10^{-6}
0.03	3.0915×10^{-5}	3.9330×10^{-5}	2.1825×10^{-6}
0.04	4.0957×10^{-5}	5.2205×10^{-5}	2.1638×10^{-6}
0.05	5.0910×10^{-5}	6.4953×10^{-5}	2.1386×10^{-6}
0.06	6.0775×10^{-5}	7.7575×10^{-5}	2.1182×10^{-6}
0.07	9.6012×10^{-5}	9.0072×10^{-5}	2.1000×10^{-6}
0.08	8.0243×10^{-5}	1.0245×10^{-5}	2.0782×10^{-6}
0.09	8.9848×10^{-5}	1.1470×10^{-4}	2.0564×10^{-6}
0.10	9.9368×10^{-5}	1.2683×10^{-4}	2.0380×10^{-6}

within Ω_s. The local weak form, Eq. (34) gives one algebraic equation relating to all these $\hat{p}_i$. Thus, one obtains as many equations as the number of nodes. Therefore, we need as many local domains Ω_s as the number of nodes in the global domain to obtain as many equations as the number of unknowns. To obtain the discrete equations from the Eq. (34), the MLS approximation in Eq. (5) is used to approximate the test function w. Substitution of Eq. (5) into Eq. (34), and summing over all nodes leads to the following discretized system of linear equations

$$\mathbf{K}\cdot\hat{\mathbf{p}}=\mathbf{f}, \tag{35}$$

where, $\hat{\mathbf{p}}=[\hat{p}_1,\hat{p}_2,\hat{p}_3,\ldots,\hat{p}_N]^T$, the entries of the "stiffness" matrix $\mathbf{K}$ and the "load" vector $\mathbf{f}$ are defined by

$$K_{ij}=\int_{\Omega_s^i}\left(\frac{\partial\phi_j}{\partial x}\frac{\partial w_i}{\partial x}+\frac{\partial\phi_j}{\partial y}\frac{\partial w_i}{\partial y}\right)d\Omega-\int_{\Gamma_{su}^i}\frac{\partial\phi_j}{\partial n}w_i d\Gamma+\alpha\int_{\Gamma_{su}^i}\phi_j w_i d\Gamma+\alpha\int_{\Gamma_{sq}^i}\left(\frac{\partial\phi_j}{\partial n}w_i\right)d\Gamma, \tag{36}$$

and

$$f_i=\alpha\int_{\Gamma_{su}^i}\bar{p}^n w_i d\Gamma+\alpha\int_{\Gamma_{sq}}\bar{q}^n w d\Gamma+\int_{\Gamma_{sq}}\bar{q}^n w d\Gamma-\frac{1}{\Delta t}\times\int_{\Gamma_{su}^i}(F^n n_1 w_i+G^n n_2 w_i)d\Gamma-\frac{1}{\Delta t}\times\int_{\Gamma_{sq}^i}(F^n n_1 w_i+G^n n_2 w_i)d\Gamma+\frac{1}{\Delta t}\times\int_{\Omega_s^i}\left(F^n\frac{\partial w_i}{\partial x}+G^n\frac{\partial w_i}{\partial y}\right)d\Omega. \tag{37}$$

6. Numerical examples

In this section, some numerical results are shown to illustrate the implementation of the present MLPG method for solving unsteady incompressible fluid flow problem. For the purpose of error estimation study, the Sobolev norm $\|\cdot\|_k$ is calculated. In the following numerical examples, the Sobolev norm for $k=0$ is considered and defined as

$$e_{r_k}=\frac{\|\mathbf{u}_{num}-\mathbf{u}_{exact}\|_k}{\|\mathbf{u}_{exact}\|_k}\qquad\text{where}\qquad\|\mathbf{u}\|_k=\left(\int_\Omega\mathbf{u}^2 d\Omega\right)^{\frac{1}{2}}.$$

The computational results indicate that the present meshless method based on the Local symmetric weak form (LSWF1, LSWF2) and the Local unsymmetric weak form (LUSWF) passes all examples. In calculation of LSWF1, both a classical Gaussian weight function and test function, which are C^2 functions, are required. In LSWF2 case, it requires an improved Gaussian weight function as trial function, and both trial and test functions are C^1 functions. For the LUSWF case, the calculation requires test function that is at least C^1 functions, while trial function, we choose a classical Gaussian weight function which is at least C^2 functions, that we call "a classical Gaussian weight order 2" herein. The boundary

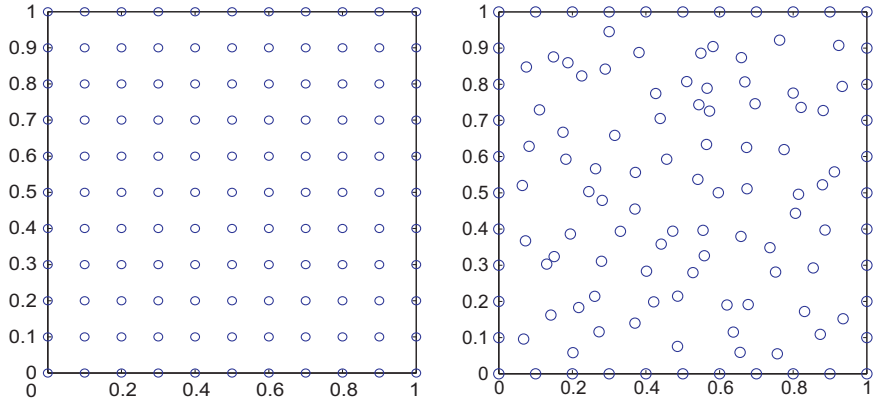

Fig. 4 Node distribution for Example 2 (11×11 nodes).

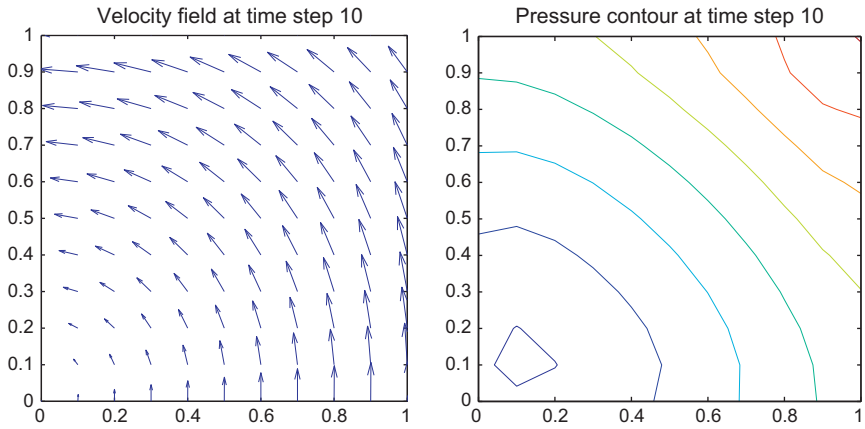

Fig. 5 The numerical solution of velocities and pressure at $\Delta t = 0.10$ of LSWF1 on regular nodes in Example 2.

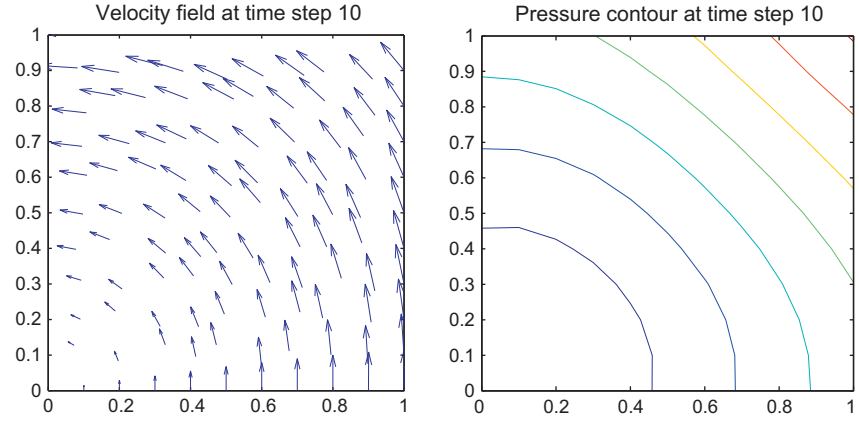

Fig. 6 The numerical solution of velocities and pressure at $\Delta t = 0.10$ of LSWF1 on irregular nodes in Example 1.

Table 7 Relative errors of velocities and pressure for each time step of LUSWF on regular nodes in Example 2.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	2.0360×10^{-2}	2.0358×10^{-2}	2.9767×10^{-2}
0.02	2.0319×10^{-2}	2.0318×10^{-2}	2.9707×10^{-2}
0.03	2.0279×10^{-2}	2.0277×10^{-2}	2.9648×10^{-2}
0.04	2.0238×10^{-2}	2.0237×10^{-2}	2.9589×10^{-2}
0.05	2.0198×10^{-2}	2.0196×10^{-2}	2.9529×10^{-2}
0.06	2.0157×10^{-2}	2.0156×10^{-2}	2.9470×10^{-2}
0.07	2.0117×10^{-2}	2.0166×10^{-2}	2.9412×10^{-2}
0.08	2.0077×10^{-2}	2.0075×10^{-2}	2.9353×10^{-2}
0.09	2.0037×10^{-2}	2.0035×10^{-2}	2.9294×10^{-2}
0.10	1.9997×10^{-2}	1.9995×10^{-2}	2.9236×10^{-2}

Table 8 Relative errors of velocities and pressure for each time step of LUSWF on irregular nodes in Example 2.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	1.2866×10^{-2}	7.2319×10^{-3}	3.0390×10^{-2}
0.02	1.2840×10^{-2}	7.2175×10^{-3}	3.0329×10^{-2}
0.03	1.2815×10^{-2}	7.2031×10^{-3}	3.0268×10^{-2}
0.04	1.2789×10^{-2}	7.1887×10^{-3}	3.0208×10^{-2}
0.05	1.2764×10^{-2}	7.1743×10^{-3}	3.0148×10^{-2}
0.06	1.2738×10^{-2}	7.1600×10^{-3}	3.0087×10^{-2}
0.07	1.2713×10^{-2}	7.1457×10^{-3}	3.0027×10^{-2}
0.08	1.2687×10^{-2}	7.1314×10^{-3}	2.9967×10^{-2}
0.09	1.2662×10^{-2}	7.1172×10^{-3}	2.9908×10^{-2}
0.10	1.2637×10^{-2}	7.1030×10^{-3}	2.9848×10^{-2}

and initial conditions of all quantities in each example can be evaluated from the exact solutions.

6.1. Example 1

This problem has an analytical solution to the two-dimensional (2D) unsteady incompressible fluid flow problem in a square domain $[0, 1] \times [0, 1]$ as shown in Fig. 1. The exact solution of the problem are

$$u(x, y, t) = 2x^2y(1 - x)^2(1 - y)(1 - 2y)e^{-t},$$
$$v(x, y, t) = -2xy^2(1 - x)(1 - 2x)(1 - y)^2e^{-t},$$
$$p(x, y, t) = (x^2 - y^2)e^{-t},$$

and the body force is

$$f(x, y, t) = 2x^2y(1 - x)^2(1 - y)(1 - 2y)e^{-t}.$$

Table 9 Relative errors of velocities and pressure for each time step of LSWF on regular nodes in Example 2.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	1.2664×10^{-3}	1.2658×10^{-3}	1.7720×10^{-3}
0.02	1.2639×10^{-3}	1.2632×10^{-3}	1.7685×10^{-3}
0.03	1.2614×10^{-3}	1.2607×10^{-3}	1.7649×10^{-3}
0.04	1.2589×10^{-3}	1.2582×10^{-3}	1.7614×10^{-3}
0.05	1.2564×10^{-3}	1.2557×10^{-3}	1.7579×10^{-3}
0.06	1.2538×10^{-3}	1.2532×10^{-3}	1.7544×10^{-3}
0.07	1.2513×10^{-3}	1.2507×10^{-3}	1.7509×10^{-3}
0.08	1.2488×10^{-3}	1.2482×10^{-3}	1.7474×10^{-3}
0.09	1.2463×10^{-3}	1.2457×10^{-3}	1.7439×10^{-3}
0.10	1.2439×10^{-3}	1.2432×10^{-3}	1.7404×10^{-3}

Table 10 Relative errors of velocities and pressure for each time step of LSWF on irregular nodes in Example 2.

Time (t)	A classical Gaussian weight function		
	e_u	e_v	e_p
0.01	2.7759×10^{-2}	4.0519×10^{-2}	1.3411×10^{-1}
0.02	2.7703×10^{-2}	4.0438×10^{-2}	1.3384×10^{-1}
0.03	2.7648×10^{-2}	4.0357×10^{-2}	1.3357×10^{-1}
0.04	2.7593×10^{-2}	4.0276×10^{-2}	1.3331×10^{-1}
0.05	2.7538×10^{-2}	4.0196×10^{-2}	1.3304×10^{-1}
0.06	2.7483×10^{-2}	4.0116×10^{-2}	1.3277×10^{-1}
0.07	2.7428×10^{-2}	4.0035×10^{-2}	1.3251×10^{-1}
0.08	2.7373×10^{-2}	3.9955×10^{-2}	1.3224×10^{-1}
0.09	2.7318×10^{-2}	3.9876×10^{-2}	1.3198×10^{-1}
0.10	2.7264×10^{-2}	3.9796×10^{-2}	1.3172×10^{-1}

We present our computed results for both regular nodes and irregular nodes on 11×11 with $\Delta t = 0.01, Re = 100$, $r_i = 0.55$, $r_0 = r_i + 0.05$ and $c_i = 4r_i$. The numerical results of LSWF1 for velocities and pressure contour on regular and irregular nodes at $t = 0.10$ are shown in Figs. 2 and 3 with 11×11 nodes. The relative errors of velocities and pressure for each time step are shown in Tables 1–6.

6.2. Example 2

This problem of Taylor decaying vortices is frequently used for validation of numerical schemes for simulating unsteady flow problems. An analytical solution in a square domain $[0, 1]\times[0, 1]$ (see Fig. 4) of problem satisfying the two-dimensional (2D), are

$$u(x,y,t) = -\cos(x)\sin(y)\exp(-2t/Re),$$
$$v(x,y,t) = \cos(y)\sin(x)\exp(-2t/Re),$$
$$p(x,y,t) = -0.25(\cos 2x + \cos 2y)\exp(-4t/Re).$$

We present our results computed both regular nodes and irregular nodes on 11×11 with $\Delta t = 0.05$ and $Re = 100$. The numerical results of LSWF for velocities and pressure contour on regular and irregular nodes at $t = 0.10$ are shown in Figs.5 and 6 with 11×11 nodes. The relative errors of velocities and pressure for each time step are shown in Tables 7–12.

Table 11 Relative errors of velocities and pressure for each time step of LSWF on regular nodes in Example 2.

Time (t)	An improve Gaussian weight function		
	e_u	e_v	e_p
0.01	4.2149×10^{-2}	4.0399×10^{-2}	2.9768×10^{-2}
0.02	6.4823×10^{-2}	6.2899×10^{-2}	2.9710×10^{-2}
0.03	8.7934×10^{-2}	8.7317×10^{-2}	2.9652×10^{-2}
0.04	1.1098×10^{-1}	1.1402×10^{-1}	2.9595×10^{-2}
0.05	1.3350×10^{-1}	1.4347×10^{-1}	2.9538×10^{-2}
0.06	1.5505×10^{-4}	1.7626×10^{-1}	2.9482×10^{-2}
0.07	1.7531×10^{-1}	2.1373×10^{-1}	2.9427×10^{-2}
0.08	1.9403×10^{-1}	2.5917×10^{-1}	2.9372×10^{-2}
0.09	2.3049×10^{-1}	3.1251×10^{-1}	2.9320×10^{-2}
0.10	2.8892×10^{-1}	3.7612×10^{-1}	2.9268×10^{-2}

Table 12 Relative errors of velocities and pressure for each time step of LSWF on irregular nodes in Example 2.

Time (t)	An improve Gaussian weight function		
	e_u	e_v	e_p
0.01	2.4992×10^{-2}	1.4528×10^{-2}	3.0390×10^{-2}
0.02	3.6312×10^{-2}	2.1858×10^{-2}	3.0330×10^{-2}
0.03	4.6918×10^{-2}	2.9215×10^{-2}	3.0270×10^{-2}
0.04	5.6865×10^{-2}	3.6573×10^{-2}	3.0211×10^{-2}
0.05	6.6206×10^{-2}	4.3955×10^{-2}	3.0151×10^{-2}
0.06	7.4986×10^{-2}	5.1300×10^{-2}	3.0092×10^{-2}
0.07	8.3249×10^{-2}	5.8537×10^{-2}	3.0033×10^{-2}
0.08	9.1034×10^{-2}	6.5624×10^{-2}	2.9974×10^{-2}
0.09	9.8380×10^{-2}	7.2523×10^{-2}	2.9915×10^{-2}
0.10	1.0532×10^{-1}	7.9200×10^{-2}	2.9856×10^{-2}

7. Results and discussion

From two examples above, we can see that the present numerical algorithm can work very well for all schemes including LSWF and LUSWF with a classical Gaussian weight and an improved Gaussian weight on regular and irregular nodes. However, the local symmetric weak form with the classical Gaussian weight order 2 gives slightly more accurate result.

8. Conclusions

In this article, a numerical algorithm using the Meshless Local Petrov-Galerkin (MLPG) method for the incompressible Navier–Stokes equations is demonstrated. To deal with the time derivatives, the forward time differences are considered to obtain a Poisson's equation. The MLPG method with the moving least-square (MLS) approximation for trial function is used to solve a Poisson's equation. In numerical examples, a classical Gaussian weight and an improved Gaussian weight are present, and the results show that LSWF1 with a classical Gaussian weight order 2 gives the most accurate result.

Acknowledgements

This research is partially supported by the Centre of Excellence in Mathematics, the Commission on Higher Education, Thai-

land. The authors would like to thank their adviser for providing advice and taking care of this research. Finally, the authors would like to thank the referees for their useful comments and language editing which have greatly improved the manuscript.

References

[1] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, Astron. J. 88 (1977) 1013–1024.
[2] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and applications to non-spherical stars, Mon. Not. Roy. Astrou. Soc. 181 (1977).
[3] B. Nayroles, G. Touzot, P. Villon, The diffuse element method, C.R. Acad. Sci. Paris Ser. II 313 (1991) 293–296.
[4] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin method, Int. J. Numer. Methods Eng. 37 (1994) 229–256.
[5] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Methods Fluids 20 (1995) 1081–1106.
[6] E. Oate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, A finite point method in computational mechanics: applications to convective transport and fluid flow, Int. J. Numer. Methods Eng. 39 (1996) 3839–3866.
[7] I. Babuška, J.M. Melenk, The partition of unity methods, Int. J. Numer. Methods Eng. 40 (1997) 727–758.
[8] Y.X. Mukherjee, S. Mukherjee, The boundary node method for potential problems, Int. J. Numer. Methods Eng. 40 (1997) 797–815.
[9] T. Zhu, J.D. Zhang, S.N. Atluri, A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach, Comput. Mech. 21 (1998) 223–235.
[10] S.N. Atluri, T. Zhu, A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics, Comput. Mech. 22 (1998) 117–127.
[11] T.L. Zhu, A new meshless regular local boundary integral equation (MRLBIE) approach, Int. J. Numer. Methods Eng. 46 (1999) 1237–1252.
[12] N.R. Aluri, G. Li, Finite cloud method: a true meshless technique based on reproducing kernel approximation, Int. J. Numer. Methods Eng. 50 (2001) 2373–2410.
[13] G.R. Liu, Y.T. Gu, A point interpolation method for two-dimensional solids, Int. J. Numer. Methods Eng. 50 (2001) 937–951.
[14] X. Zhang, X.H. Liu, Z.K. Song, M.W. Lu, Least-square collocation meshless method, Int. J. Numer. Methods Eng. 51 (2001) 1089–1100.
[15] S.N. Atluri, S. Shen, The Meshless Local Petrov-Galerkin (MLPG) Method, Tech Science Press, Los Angeles, CA, 2002.
[16] H. Lin, S.N. Atluri, Meshless Local Petrov-Galerkin (MLPG) method for convection-diffusion problems, CMES Comput. Model. Eng. Sci. 1 (2) (2000) 45–60.
[17] Y.L. Wu, G.R. Liu, Y.T. Gu, Application of Meshless Local Petrov-Galerkin (MLPG) approach to simulation of incompressible flow, Numer. Heat Tran. 48 (2005) 459–475.
[18] Y.V.S.S. Sanyasiraju, G. Chandhini, Local radial basis function based gridfree scheme for unsteady incompressible viscous flows, Comput. Phys. 227 (2008) 8922–8948.
[19] E. Tanojo, Derivative of moving least squares approximation shape functions and its derivatives uaing the exponential weight function, Civil Eng. Dimension (2007) 19–24.