Scientia Iranica, Vol. 12, No. 2, pp 151–166

SCIENTIA
IRANICA

Navier-Stokes Calculations Using a Finite Point Meshless Method

S.K. Hannani* **and M.M. Sadeghi**[1]

The objective of this research is to study the ability of a meshless method, called finite point method, in solving incompressible fluid flow problems using two stabilization schemes. The main goal of meshless methods is to reduce or remove the cost of grid generation. This issue is implemented using the satisfaction of governing differential equations on a regular or irregular set of nodes by interpolation functions, based on special least-squares approximations. In this research, the finite point method is used to solve the Stokes and the Navier-Stokes equations by employing two different stabilization schemes. In addition, the effects of least-squares approximations are studied.

INTRODUCTION

During the past three decades, numerical methods have been used extensively to simulate various engineering problems. The basis of existing numerical techniques depends on domain meshing. Grid generators produce efficient grids in two-dimensional and smooth domains, but, in complicated domains, producing an efficient grid is difficult and very time-consuming [1]. This problem is serious and, in complicated three-dimensional domains, this part of the solution process requires more time and cost [2]. On the other hand, in moving-boundary problems, ordinary methods require an updated grid that coincides with the new boundary at each time step. The cost of these problems, using ordinary numerical methods, will be very high. In addition, these methods that require a large number of remeshings, introduce numerous difficulties, such as the need to project between meshes at successive stages of the problem, which leads to a degradation of accuracy and more complexity in computer programming [3].

The objective of meshless methods is to eliminate the above difficulties, using a method that needs only the spatial position of some distributed nodes in the domain of the problem. Thus, it becomes possible to solve large classes of problems, which are sometimes very awkward with mesh-based methods.

*. *Corresponding Author, Department of Mechanical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*

1. *Department of Mechanical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*

The first attempts made using meshless methods were reported by a few Finite Difference (FD) practitioners deriving FD schemes in arbitrary irregular grids using the Taylor series [4-7]. Smooth Particle Hydrodynamics (SPH) is an alternative class of meshless methods that requires only a set of irregular nodes. This method, which was introduced by Lucy and complemented by Gingold and Moraghan [8], works well in problems without boundaries. However, it is not as accurate as regular finite element methods [1]. Nayroles et al. [9] proposed a technique, namely, the Diffuse Element Method (DEM), where only a collection of nodes and a boundary description is needed to formulate the Galerkin equations. Although this method works without any finite element mesh, still, some kind of auxiliary grid is necessary in order to compute, numerically, the integral expressions derived from the Galerkin approach. Belytschko et al. [10,11] proposed an extension of the DEM approach, namely, Element-Free Galerkin (EFG). In this method, a regular cell structure is needed to compute the integrals by means of high order quadratures. Duarte and Oden [12] represented a new approximation scheme that is a subdomain of Partition of Unity (PU) methods and, then, introduced a meshless method with this new approximation. Liu et al. [13,14] introduced another meshless method, based on wavelet analysis, which is called Reproducing Kernel Particle (RKP).

Onate et al. introduced the Finite Point Method (FPM) [1,15,16] that uses some kind of approximations similar to DEM and EFG approximations. After representing the FIC stabilization method by Onate [17], this method was complemented and its evolution has

continued up to now [18,19]. Recently, Meshless Local Boundary Integral Equation (MLBIE) and Meshless Local Petrov-Galerkin (MLPG) methods were represented based on integral equations [20]. These methods work without any grid and integrations are implemented in specific subdomains. The Least-Square Collocation Method (LSCM) is a kind of FPM method that uses some auxiliary nodes, in addition to basic distributed nodes. This method satisfies discretized equations in the least-squares sense by applying auxiliary nodes to improve accuracy [21].

Among the mentioned meshless methods, only FPM, LSCM, MLBIE and MLPG do not use any auxiliary or background grid. In this paper, the FPM method is used to solve classical problems such as heat conduction, the Stokes problem and Navier-Stokes equations for incompressible fluid flow in different domains. The accuracy of this method is validated, compared with some analytical, numerical and experimental solutions and its advantages and disadvantages are elucidated.

FPM FORMULATION

Generally, one of the most important parts of each numerical method is the discretization method. This part is very important and determines the governing relation in each element or set of nodes. The weighted residual method is the first step of many discretization methods.

Weighted Residual Method

Let one assume a scalar problem governed by a differential equation:

$$A(u) = b \qquad \text{in } \Omega, \tag{1}$$

and with the following Neumann and Dirichlet boundary conditions:

$$B(u) = t \qquad \text{in } \Gamma_t, \tag{2a}$$

$$u - u_p = 0 \qquad \text{in } \Gamma_u. \tag{2b}$$

In the above, A and B are appropriate differential operators, u is the problem unknown and b and t represent sources and external forces acting over the domain, Ω, and along the boundary, Γ_t, respectively. Finally, u_p is the prescribed value of u on the boundary, Γ_u.

The weighted residual method is the most general procedure for solving, numerically, the above system of differential equations. According to this method, the unknown function, u, is approximated by some trial approximation, $\hat{u}$, and the following equation is yielded:

$$\int_\Omega W_i[A(\hat{u}) - b]d\Omega + \int_{\Gamma_t} \overline{W}_i[B(\hat{u}) - t]d\Gamma + \int_{\Gamma_u} \overline{\overline{W}}_i[\hat{u} - u_p]d\Gamma = 0. \tag{3}$$

Selecting weighting functions, $W_i, \overline{W}_i$ and $\overline{\overline{W}}_i$ in different ways results in different numerical methods, such as FDM, FVM and FEM [22].

In order to keep the local character of the problem, function u should be approximated by a combination of locally defined functions as:

$$u(x) \cong \hat{u}(x) = \sum_{i=1}^{n_p} N_i(x)u_i^h = \mathbf{N}^T(x)\mathbf{u}^h, \tag{4}$$

with n_p being the total number of points in the domain and the interpolation functions, $N_i(x)$, satisfying the following conditions:

$$N_i(x) \neq 0 \qquad x \in \Omega_i, \tag{5a}$$

$$N_i(x) = 0 \qquad x \notin \Omega_i. \tag{5b}$$

where Ω_i is a subdomain of Ω containing n points and $n < n_p$.

Point Collocation

In the FPM, discretized equations are derived by point collocation [1]. Therefore, in the weighted residual form of the governing equations, one has:

$$W_i = \overline{W}_i = \overline{\overline{W}}_i = \delta_i, \tag{6}$$

where δ_i is the Dirac delta function. This is necessary in order to preserve the meshfree character of the method, because, with this weight function no surface or volume integration is needed.

Finally, the weighted residual form of the governing equations yields the following equations:

$$[A(\hat{u})]_i - b_i = 0 \qquad \text{in } \Omega, \tag{7a}$$

$$[B(\hat{u})]_i - t_i = 0 \qquad \text{in } \Gamma, \tag{7b}$$

$$\hat{u}_i - u_p = 0 \qquad \text{in } \Gamma_u. \tag{7c}$$

The above equations may be written in the matrix form as:

$$\mathbf{K}\mathbf{u}^h = \mathbf{F}, \tag{8}$$

where $\mathbf{K}$ is a $n_p \times n_p$ matrix, $\mathbf{u}$ is the unknown vector, consisting of $u_i^h (i = 1, 2, \cdots, n_p)$ and $\mathbf{f}$ is a vector

containing the contributions from the force terms b and t and the prescribed values, u_p.

Meshless methods based on point collocation are true meshless schemes and are very efficient. However, in collocation-based methods, equilibrium conditions are only satisfied at n_Ω collocation points within the domain, Ω. If collocated points are not sufficient, significant error may result. In Galerkin-based methods, equilibrium conditions are satisfied within the domain in an integral sense, so that information at Gauss quadrature points is included. Usually, the accuracy of Galerkin-based methods is better than collocation-based methods, but the computational effort required for Galerkin-based methods is also much more. This point is important, especially in meshless methods. Meshless methods that use integral equations lose some important properties of a real meshless method. Generally, these methods need an auxiliary grid named background grid [21].

INTERPOLATION IN THE FPM

In meshless methods, there are no elements to interpolate variables on them like FEM. Therefore, this part of meshless methods is the most different compared with ordinary methods.

Let Ω_i be the interpolation domain (cloud) of a function, $u(x)$, and let s_j with $j = 1, 2, \cdots, n$ be a collection of n points with coordinates $x_j \in \Omega_j$. The unknown function, u, may be approximated within Ω_i by the following equation:

$$u(x) \cong \hat{u}(x) = \sum_{L=1}^{m} p_L(x)\alpha_L = \mathbf{p}(x)^T \boldsymbol{\alpha}, \tag{9}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_m]^T$ and vector $\mathbf{p}(x)$ contains typically monomials, hereafter termed "basis interpolating functions", in the space coordinates ensuring that the basis is complete. For a 2D problem one can specify:

$$\mathbf{p} = [1, x, y]^T, \qquad m = 3, \tag{10a}$$

$$\mathbf{p} = [1, x, y, x^2, xy, y^2]^T, \qquad m = 6. \tag{10b}$$

Function $u(x)$ can now be sampled at the n points belonging to Ω_i, giving:

$$u^h = \begin{Bmatrix} u_1^h \\ u_2^h \\ \vdots \\ u_n^h \end{Bmatrix} \cong \begin{Bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_n^T \\ \vdots \\ \mathbf{p}_n^T \end{Bmatrix} \boldsymbol{\alpha} = \mathbf{C}\boldsymbol{\alpha}, \tag{11}$$

where $u_j^h = u(x_j)$ are the unknown but sought for values of function u at point $j, \hat{u}_j = \hat{u}(x_j)$ are the approximate values and $\mathbf{p}_j = \mathbf{p}(x_j)$.

In finite element approximation, the number of points is chosen so that $m = n$. In this case, $\mathbf{C}$ is a square matrix, $\boldsymbol{\alpha}$ can be obtained readily and, finally, the approximation is completed:

$$\boldsymbol{\alpha} = \mathbf{C}^{-1}\mathbf{u}^h, \tag{12}$$

$$u \cong \hat{u} = \mathbf{p}^T\mathbf{C}^{-1}\mathbf{u}^h = \mathbf{N}^T\mathbf{u}^h = \sum_{j=1}^{m} N_j^i u_j^h, \tag{13}$$

$$\mathbf{N}^T = \left[N_1^i, \cdots, N_n^i\right] = \mathbf{p}^T\mathbf{C}^{-1},$$

$$N_j^i = \sum_{i=1}^{m} p_i(x) C_{ij}^{-1}. \tag{14}$$

Obtained shape functions, N_j^i, satisfy the standard condition:

$$N_j^i(x_i) = 1 \qquad (i = j), \tag{15a}$$

$$N_j^i(x_i) = 0 \qquad (i \neq j). \tag{15b}$$

In this part, interpolation schemes for meshless methods are introduced concisely. This context is one of the important bases in meshless methods. Complete discussions are mentioned in [1].

Standard Least-Squares (LSQ) Approximation

If $n > m, \mathbf{C}$ is no longer a square matrix and the approximation cannot fit all the u_j^h values. This problem can simply be overcome by minimizing the sum of the squared distances of the error at each point with respect to α_i parameters. The following equation expresses this summation:

$$J = \sum_{j=1}^{n} (u_j^h - \hat{u}(x_j))^2 = \sum_{j=1}^{n} (u_j^h - \mathbf{p}_j^T\boldsymbol{\alpha})^2. \tag{16}$$

Standard minimization leads to:

$$\boldsymbol{\alpha} = \mathbf{C}^{-1}\mathbf{u}^h \quad \text{with} \quad \mathbf{C}^{-1} = \mathbf{A}^{-1}\mathbf{B}, \tag{17}$$

where:

$$\mathbf{A} = \sum_{j=1}^{n} \varphi(x_j)\mathbf{p}(x_j)\mathbf{p}^T(x_j), \tag{18a}$$

$$\mathbf{B} = [\varphi(x_1)\mathbf{p}(x_1), \varphi(x_2)\mathbf{p}(x_2), \cdots, \varphi(x_n)\mathbf{p}(x_n)]. \tag{18b}$$

The final approximation is still given by Equation 13. The shape functions are, therefore:

$$N_j^i(x) = \sum_{L=1}^{m} p_L(x)\mathbf{C}_{Lj}^{-1} = \mathbf{p}^T(x)\mathbf{C}^{-1}, \tag{19}$$

where superindex i emphasizes that shape functions, N_j^i, can now be defined differently for each cloud, Ω_i. It must be noted that, according to the least-squares character of the approximation,

$$u(x_j) \cong \hat{u}(x_j) \neq u_j^h, \tag{20}$$

i.e., the local values of approximating function do not fit the nodal unknown values.

Indeed, the approximate function, $\hat{u}$, is the function that must satisfy the discretized form of the governing differential equation and its boundary conditions and u_j^h are simply the unknown parameters sought [1,15].

The main drawback of the LSQ approach is that the approximation rapidly deteriorates if the number of used points largely exceeds the polynomial terms [1].

Weighted Least-Squares (WLS) Approximation

The LSQ approximation can be enhanced near a desired point by weighting the squared distances using the function φ with the following characteristics:

$$\begin{cases} \varphi_i(x_i) = 1 & \\ \varphi_i(x) \neq 0 & x \in \Omega_i \\ \varphi_i(x) = 0 & x \notin \Omega_i \end{cases}. \tag{21}$$

This function usually takes unit value (or its maximum value) near the point called the star point where the derivatives of the unknown function are to be evaluated. In WLS, the following summation should be minimized:

$$J = \sum_{j=1}^{n} \varphi_i(x_j)(u_j^h - \hat{u}(x_j))^2. \tag{22}$$

So, $\mathbf{A}$ and $\mathbf{B}$ are obtained in the following equations:

$$\mathbf{A} = \sum_{j=1}^{n} \varphi_i(x_j)\mathbf{p}(x_j)\mathbf{p}^T(x_j), \tag{23a}$$

$$\mathbf{B} = [\varphi_i(x_1)\mathbf{p}(x_1), \varphi_i(x_2)\mathbf{p}(x_2), \cdots, \varphi_i(x_n)\mathbf{p}(x_n)]. \tag{23b}$$

Moving Least-Squares (MLS) Approximation

In the MLS approximation, the weighting function takes its maximum value over each point that the unknown function should be evaluated. In the WLS approach, the peak of the weighting function is placed only on distributed nodes, but in the MLS approach, the peak of the weighting function can be placed on each desired point in the domain.

For every point, the following summation should be minimized:

$$J(x_k) = \sum_{j=1}^{n} \varphi_k(x_j)(u_j^h - \mathbf{p}_j^T\boldsymbol{\alpha})^2. \tag{24}$$

Now, $\mathbf{A}, \mathbf{B}$ and α_i parameters are functions of x_k.

$$\mathbf{A} = \sum_{j=1}^{n} \varphi_k(x_j)\mathbf{p}(x_j)\mathbf{p}^T(x_j), \tag{25a}$$

$$\mathbf{B} = [\varphi_k(x_1)\mathbf{p}(x_1), \varphi_k(x_2)\mathbf{p}(x_2), \cdots, \varphi_k(x_n)\mathbf{p}(x_n)]. \tag{25b}$$

In LSQ and WLS methods, α parameters are constant in each subdomain, Ω_i, and the approximation order is, directly, the order included in the set of basis functions. On the other hand, in the MLS approach, α parameters are functions of position and the resultant unknown function may include higher order functions.

There is an important characteristic in the MLS approach. The shape functions of this method are global and can be used all over the domain. Of course, this property is not so important in the FPM method from a computational point of view, because only the values of the unknown function on distributed nodes are needed.

Least-Squares Approximations Properties

Like the finite element approximation, the least-squares approximation is exact for any function consisting of a linear combination of the basis functions [9,15]. This property is called shape functions consistency.

Another important property is that, if the weighting function is continuous and the matrix $\mathbf{A}$ is regular, then the approximate function and the estimates of its derivatives are continuous [9].

In a good least-squares approximation, there is a possibility to introduce new points independently of the distance between existing points [1]. In the finite element approximation, two nodes that are very close together cause numerical error, therefore, this adds severe limitations in the mesh generator and adaptivity criteria.

The above properties are general characteristics of each least-squares approximation. In addition, a good least-squares approximation for FPM is insensitive to the number of points chosen within each cloud. This condition is necessary to preserve the freedom of adding, moving or removing points for a given order of interpolation.

Although all least-squares approximations have the first three properties, the fourth one is the most important for an applicable method in FPM. Of the above mentioned schemes, the following are unsuitable for application to the FPM method [1]:

1. LSQ approximations,
2. WLS with linear basis functions,
3. MLS with linear basis functions.

Indeed, the remaining approximation methods can work properly.

Shape Functions Derivatives

Let one assume a least-squares approximation for function u in the following equation:

$$u \cong \hat{u}(x) = \mathbf{p}^T \mathbf{A}^{-1} \mathbf{B} \mathbf{u}^h = \mathbf{N}^T \mathbf{u}^h. \tag{26}$$

In LSQ and WLS methods, $\mathbf{A}$ and $\mathbf{B}$ are constant over each cloud of interpolation and, therefore, derivatives can be computed readily, for example:

$$\frac{\partial \mathbf{N}^T}{\partial x} = \frac{\partial \mathbf{p}^T}{\partial x} \mathbf{A}^{-1} \mathbf{B}. \tag{27}$$

However, in the MLS method, $\mathbf{A}$ and $\mathbf{B}$ are functions of position and derivatives of shape functions are influenced by the derivatives of these matrices [15]. Nayroles et al., in DEM, neglected this effect and assumed it dispensable [9]. On the other hand, Belytschko et al., in EFG, considers this effect and computes the derivatives of $\mathbf{A}$ and $\mathbf{B}$ [10,11].

Weight Functions

The weight function affects the approximation function. As an illustration, consider the three cases depicted in Figure 1 where function $u(x)$ in one dimension is approximated using five data points at $x = 0, 1, 2, 3, 4$. The least-squares approximation function is constructed using a linear polynomial basis, $\mathbf{p}^T = [1, x]$. In the first, second and third examples, approximation methods are LSQ over the entire domain (constant weight over the entire domain), MLS with constant weight function and compact support on two nodes and MLS with smooth weight function and compact support, respectively. Differences between the results of these approximation methods are obvious. Figure 1 reveals that efficient approximation is obtained by smooth weight function with compact support.

Some examples of weight functions are mentioned in the following equations:

Gaussian:

$$w(\overline{s}) = \begin{cases} \frac{e^{(s/c)^2} - e^{-(s_{\max}/c)^2}}{1 - e^{-(s_{\max}/c)^2}} & s \le s_{\max} \\ 0 & s > s_{\max} \end{cases}, \tag{28a}$$

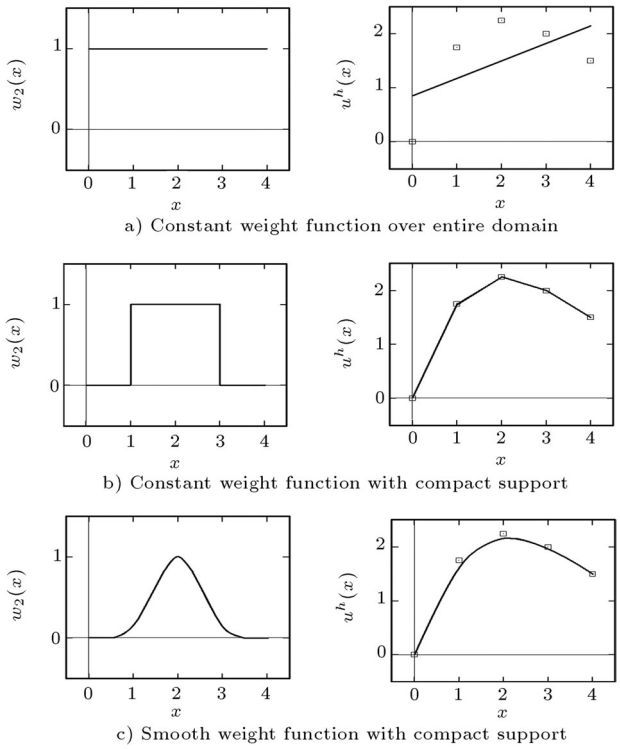

Figure 1. Effect of weight function on resulting approximation function.

exponential:

$$w(\overline{s}) = \begin{cases} e^{-(\overline{s}/\alpha)^2} & \overline{s} \le 1 \\ 0 & \overline{s} > 1 \end{cases}, \tag{28b}$$

cubic spline:

$$w(\overline{s}) = \begin{cases} 2/3 - 4\overline{s}^2 + 4\overline{s}^3 & \overline{s} \le 1/2 \\ 4/3 - 4\overline{s} + 4\overline{s}^2 - 4/3\overline{s}^3 & 1/2 < \overline{s} \le 1 \\ 0 & \overline{s} > 1 \end{cases}, \tag{28c}$$

fourth order spline:

$$w(\overline{s}) = \begin{cases} 1 - 6\overline{s}^2 + 8\overline{s}^3 - 3\overline{s}^4 & \overline{s} \le 1 \\ 0 & \overline{s} > 1 \end{cases}, \tag{28d}$$

where s is the distance from the center of the weight function, $\overline{s} = s/s_{\max}$ and $s_{\max}$ is the radius of support. Parameter c in the Gaussian weight function determines the sharpness of the weight function. Parameter α in the exponential weight function is usually assumed as 0.4. Another parameter for weight functions is $d_{\max}$. After determining the radius of support, $s_{\max}$ is considered as this radius multiplied by $d_{\max}$.

STABILIZED GOVERNING EQUATIONS

The reliable numerical analysis of non-self-adjoint differential equations, such as governing equations for

fluid flow problems, requires some additional operations, namely, stabilization [23]. In FDM, FVM and FEM, in order to study fluid flow problems, the governing equations should be stabilized and the FPM method cannot be excluded from this drawback. Generally, equations are stabilized in various ways by adding balancing diffusion or artificial diffusion. The instability of numerical results has two resources, convective terms at high Reynolds numbers and an inconvenient combination of pressure and velocity interpolation.

In recent years, various methods were presented in the field of stabilization, such as artificial diffusion, Petrov-Galerkin, streamline upwind Petrov-Galerkin, subgrid scale, Galerkin least-squares, Lax-Wendroff schemes and characteristic approximation methods [17].

Stabilization methods, which are based on integral equations, are not applicable for FPM. Among the remaining methods, some of them cannot work well and a method is needed that modifies errors caused by the local satisfaction of differential equations. In this paper, SSUPG and FIC methods are used to stabilize governing equations.

Simplified SUPG (SSUPG) Formulation

Assume the governing equations of incompressible fluid flow problems:

$$\rho \mathbf{u}.\nabla \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{F}, \tag{29a}$$

$$\nabla.\mathbf{u} = 0. \tag{29b}$$

In Streamline Upwind/Petrov Galerkin (SUPG) formulation, weighted residual equations are the following [24]:

$$\int \widetilde{W}(\rho \mathbf{u}.\nabla \mathbf{u} + \nabla p - \mu \nabla^2 \mathbf{u} - \rho \mathbf{F}) d\Omega = 0, \tag{30a}$$

$$\int W_p(\nabla.\mathbf{u}) d\Omega + \int \alpha \nabla W_p(\rho \mathbf{u}.\nabla \mathbf{u} + \nabla p - \mu \nabla^2 \mathbf{u} - \rho \mathbf{F}) d\Omega = 0. \tag{30b}$$

In the above, W_p is the shape function of pressure and α is the stabilization parameter. $\widetilde{W}$ is considered as:

$$\widetilde{W} = W + \alpha \mathbf{u}.\nabla W, \tag{31}$$

where W is the shape function of velocity. Final integral equations for SUPG are obtained after implementing integration, by part, on Equations 30a and 30b.

In SSUPG, momentum equations remain unchanged and for the continuity equation, a simplified form of Equation 30b is used. Therefore, the final equations of SSUPG are:

$$\rho \mathbf{u}.\nabla \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{F}, \tag{32a}$$

$$\nabla.u + \alpha \nabla^2 p = 0. \tag{32b}$$

With this set of equations, instability potentials are circumvented.

Finite Increment Calculus (FIC) Formulation

Indeed, it is widely accepted that the origins and precise definition of the stabilization parameters used in numerical computations are, in most cases, unsolved mysteries [16]. In a quite recent work, Onate [17] has shown that the stabilization terms emerge naturally in the governing differential equations of the problem, once the concept of balance over a finite domain is accepted. This allows reinterpreting the stabilization terms as an intrinsic and natural contribution to the original differential equations, instead of an extrinsic correction term introduced at the discretization level, as usually understood by most FDM, FVM and FEM practitioners.

In FIC formulation, the stabilized governing equations for incompressible viscous flows are obtained by applying the standard conservation laws expressing balance of momentum and mass over a control domain. Assuming that the control domain has finite dimensions and representing the variation of mass and momentum over the domain, using Taylor series expansions of one order higher than those used in the standard infinitesimal theory, the following expressions are found [17,25]:

Momentum balance:

$$r_{m_i} - \frac{1}{2} h_{m_j} \frac{\partial r_{m_i}}{\partial x_j} = 0 \quad \text{in } \Omega. \tag{33}$$

Mass balance:

$$r_d - \frac{1}{2} h_{d_j} \frac{\partial r_d}{\partial x_j} = 0 \quad \text{in } \Omega, \tag{34}$$

where for a steady state case:

$$r_{m_i} = \rho \frac{\partial (u_i u_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} - \mathbf{b}_i, \tag{35}$$

$$r_d = \frac{\partial u_i}{\partial x_i}, \tag{36}$$

with $i, j = 1, 2$ for a two-dimensional flow. In the above, ρ is the fluid density (here assumed to be constant), u_i is the velocity component in the ith direction, p the pressure, b_i the body forces and τ_{ij}

the viscous stress components related to the velocity gradients through the fluid viscosity, μ, by:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right). \tag{37}$$

Equations 33 and 34 are the stabilized forms of the governing differential equations for an incompressible fluid flow. In these equations, the second terms introduce, naturally, the necessary stabilization at the discretization level. The so-called characteristic length vectors, $\mathbf{h}_m$ and $\mathbf{h}_d$, for two-dimensional problems, are defined as:

$$\mathbf{h}_m = \begin{Bmatrix} h_{m_1} \\ h_{m_2} \end{Bmatrix}, \quad \mathbf{h}_d = \begin{Bmatrix} h_{d_1} \\ h_{d_2} \end{Bmatrix}, \tag{38}$$

where h_{m_1} and h_{m_2} are the dimensions of the finite control domain, where balance of momentum is enforced. Similarly, h_{d_1} and h_{d_2} represent the dimensions of the domain where mass conservation is expressed. The components of vectors $\mathbf{h}_m$ and $\mathbf{h}_d$ introduce the necessary stabilization along the streamline and transverse directions to the flow in the discrete problem.

Equations 33 and 34 are complemented by the boundary conditions. Balance of momentum at the boundary is expressed by Equation 39. This equation is derived using a similar method to the one used for governing equations.

$$n_j \sigma_{ij} - t_i + \frac{1}{2} h_{m_j} n_j r_{m_i} = 0 \quad \text{on } \Gamma_t, \tag{39}$$

where n_i is the ith component of the unit normal vector to the boundary and t_i are the prescribed tractions at the Neumann boundary, Γ_t, of the analysis domain, Ω.

Prescribed velocity at the boundaries is imposed by the following equations:

$$u_t = u_t^p \quad \text{on } \Gamma_{u_t}, \tag{40a}$$

$$u_n - \frac{1}{2} h_{d_i} n_i r_d = u_n^p \quad \text{on } \Gamma_{u_n}, \tag{40b}$$

u_t and u_t^p denote the tangential velocity to the boundary and its prescribed value, respectively.

Equation 40b expresses the balance of mass on an arbitrary domain next to the boundary. u_n and u_n^p denote the velocity normal to the boundary and its prescribed value, respectively. The value of u_n^p is zero on solid walls and stationary free surfaces. Considering the finite domain leads to additional terms in Equations 39 and 40b. These terms introduce the necessary stabilization at the boundaries.

Alternative Form of Stabilized Equations

Components of the characteristic vector, h_d, can be expressed as:

$$h_{d_i} = 2\rho \tau_{d_i} u_i, \tag{41}$$

where τ_{d_i} parameters are termed intrinsic times per unit mass. After substituting the above equation in the stabilized form of the continuity equation and some operations and neglecting higher order terms, it changes to another applicable form of the stabilized continuity equation [18,19], represented in the following equation:

$$r_d - \tau_{d_i} \frac{\partial \overline{r}_{m_i}}{\partial x_i} = 0, \tag{42}$$

where, in Equation 42, one has:

$$\overline{r}_{m_i} = \rho u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} - b_i. \tag{43}$$

This form of continuity equation can be used for incompressible fluid flow calculations, as well as Equation 34.

FRACTIONAL STEP (FS) ALGORITHM

A fractional step algorithm is proposed by Zienkiewicz et al. [26,27] for transient calculations of Navier-Stokes equations. This time-marching algorithm can work in three features: Explicit, semi-implicit and implicit. The semi-implicit form is applicable for incompressible fluid flow problems and reduces the $3n \times 3n$ system of equations to $n \times n$.

FS Algorithm for Stabilized FIC Equations

Assume Navier-Stokes equations are stabilized by FIC. One can write:

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\rho} \left[\rho \frac{\partial (u_i u_j)^n}{\partial x_j} + \frac{\partial p^{n+1}}{\partial x_i} - \frac{\partial \tau_{ij}^n}{\partial x_j} \right] - \frac{\Delta t}{\rho} \left[-b_i^n - \frac{1}{2} \left(\mathbf{h}^T \nabla r_{m_i} \right)^n \right]. \tag{44}$$

Equation 44 can be separated into the following equations:

$$u_i^* = u_i^n - \frac{\Delta t}{\rho} \left[\rho \frac{\partial (u_i u_j)}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} - b_i \right]^n - \frac{\Delta t}{\rho} \left[-\frac{1}{2} \left(\mathbf{h}^T \nabla r_{m_i} \right) \right]^n, \tag{45a}$$

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho} \frac{\partial p^{n+1}}{\partial x_i}. \tag{45b}$$

Substitution of Equation 45b in the stabilized form of the continuity equation yields:

$$r_d^* - \frac{\Delta t}{\rho} \frac{\partial^2 \mathbf{p}^{n+1}}{\partial x_i \partial x_i} - \left[\tau_i \frac{\partial r_{m_i}}{\partial x_i} \right]^{n+1} = 0, \tag{46}$$

where, in the above equation, one has:

$$r_d^* = \frac{\partial u_i^*}{\partial x_i}, \tag{47}$$

$$\left[\tau_i \frac{\partial r_{m_i}}{\partial x_i}\right]^{n+1} = \left[\rho\left(\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j}\right) - \frac{\partial \tau_{ij}}{\partial x_j} - b_i\right]^n - \tau_i^{n+1} \frac{\partial^2 p^{n+1}}{\partial x_i \partial x_i}. \tag{48}$$

Therefore, the solution comprises the following three steps:

1. Solve explicitly for the so-called fractional velocities, u_i^*, using Equation 45a;
2. Compute the pressure field, p^{n+1}, by solving the equation for the Laplacian of pressure derived from Equation 46;
3. Compute the velocities, u_j^{n+1}, using Equation 45b.

Obviously, other forms of transient solution schemes, including the implicit computation of u_i^{n+1}, are also possible.

Numerical Solution Using the FPM

The implementation of the three-step scheme described in the previous section, in the context of FPM, is straightforward. This scheme is introduced by Onate et al. [18,19] for the first time.

Assume approximations for velocity components and pressure:

$$\hat{u}_m = \sum_{j=1}^{n} N_j u_{mj}^h, \quad \hat{p} = \sum_{j=1}^{n} N_j p_j^h. \tag{49}$$

In the first step, fractional velocities, $\hat{u}^*$, are computed:

$$\hat{u}_i^* = u_i^n - \frac{\Delta t}{\rho}\left[\rho \frac{\partial (\hat{u}_i \hat{u}_j)}{\partial x_j} - \frac{\partial \hat{\tau}_{ij}}{\partial x_j} - b_i\right]^n - \frac{\Delta t}{\rho}\left[-\frac{1}{2}\left(\mathbf{h}^T \nabla \hat{r}_{m_i}\right)\right]^n. \tag{50}$$

Then, parameters of fractional velocities, u_{mj}^h, are computed from the system of the following equation:

$$\hat{u}_m^* = \sum_{j=1}^{n} N_j u_{mj}. \tag{51}$$

These parameters are necessary to compute the derivatives of the velocity functions in the next steps.

In the second step, pressure distribution is obtained on the domain with the following equations:

$$\mathbf{K}(\mathbf{p}^h)^{n+1} = \hat{r}_d^*, \tag{52a}$$

$$K_{kj} = \left(\frac{\Delta t}{\rho} + \tau_i^{n+1}\right)\left(\frac{\partial^2 N_j^k}{\partial x_i \partial x_i}\right), \tag{52b}$$

$$\hat{r}_{d_k}^* = r_d^n - \tau_i^n \frac{\partial}{\partial x_i}\left[\rho\left(\frac{\partial \hat{u}_i}{\partial t} + \frac{\partial (\hat{u}_i \hat{u}_j)}{\partial x_j}\right) - \frac{\partial \hat{\tau}_{ij}}{\partial x_j}\right]_k^n, \tag{52c}$$

where $(.)_n^k$ represents the value of point k in the nth time step.

The last step is the computation of velocities in the $(n+1)$th time step explicitly.

$$\hat{u}_i^{n+1} = \hat{u}_i^* - \frac{\Delta t}{\rho}\frac{\partial \hat{p}^{n+1}}{\partial x_i}. \tag{53}$$

In the above equations, derivatives of approximation functions $\hat{u}_i$ and $\hat{p}_i$ are obtained from the following equations:

$$\frac{\partial \hat{u}_m}{\partial x_L} = \sum_{j=1}^{n} \frac{\partial N_j^i}{\partial x_L} u_{mj}^h, \tag{54a}$$

$$\frac{\partial \hat{p}}{\partial x_L} = \sum_{j=1}^{n} \frac{\partial N_j^i}{\partial x_L} p_j^h. \tag{54b}$$

For each time step, these three steps should be implemented. The local time step for each node is $\frac{d_i}{2|u_i|}$, where d_i is the minimum distance from the star node to its neighbors. The local time step can be used for steady state results, but, the transient solution requires the use of a global time step that is equal to the minimum value of the local time steps.

Boundary Conditions

In a fractional step algorithm, the first step is implemented without any boundary conditions. For the second step, two kinds of boundary conditions should be enforced. In the case of boundaries which have a prescribed normal velocity, one has:

$$u_n^p = u_i^* n_i - \frac{\Delta t}{\rho}\frac{\partial p^{n+1}}{\partial x_i} n_i - \frac{1}{2} h_{d_i} n_i r_d, \tag{55}$$

or, in other words;

$$u_n^p = u_i^* n_i - \frac{\Delta t}{\rho}\frac{\partial p^{n+1}}{\partial x_i} n_i - \frac{1}{2} h_{d_i} n_i \left[(\Delta t + \tau)\Delta p^n - \overline{r}_d^*\right]. \tag{56}$$

Equations 55 and 56 are stabilized boundary conditions for the pressure equation.

On outflow boundaries with $n_j \sigma_{ij} = 0$, the pressure is imposed to a constant value. In the FPM, essential boundary conditions, such as $p = 0$, are

imposed using the definition of the function itself as the following equation:

$$\hat{p}_i = \sum_{j=1}^{n} N_j^i p_j^h. \tag{57}$$

In the third step of the fractional step algorithm, u^{n+1} is computed via an explicit relation. This time, essential boundary conditions of velocity are imposed. For points on the Neumann boundaries, the following equation is applicable:

$$n_j \sigma_{ij} + \frac{1}{2} h_{m_j} n_j r_{m_i} = 0. \tag{58}$$

NUMERICAL EXPERIMENTS

This part includes analyses of some partial differential equations by FPM. Heat conduction problems, Stokes equations and Navier-Stokes equations are analyzed in different domains and the results are compared with FDM or FEM results by solving benchmark problems.

Heat Conduction

The heat conduction equation is a well-posed partial differential equation, so it is useful to study the general properties of each method. In non-well-posed partial differential equations, such as non-self-adjoint problems, other phenomena appear and affect the results, so, the basis of a method for general problems cannot be examined.

Assume the two-dimensional heat conduction problem mentioned in the following equations:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0,$$

$$0 \le x \le 1,\ 0 \le y \le 1, \tag{59a}$$

$$T(x,0) = T(0,y) = T(1,y) = 0,$$

$$0 \le x \le 1,\ 0 \le y \le 1, \tag{59b}$$

$$T(x,1) = 100,$$

$$0 < x < 1. \tag{59c}$$

The analytical solution of the above problem is obtained by the separation of variables method. This solution is expressed in the following equation:

$$T(x,y) = \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}+1}{n} \sin(n\pi x) \frac{\sinh(n\pi y)}{\sinh(n\pi)}. \tag{60}$$

As mentioned before, one of the good properties of each approximation is the insensitivity to the number of nodes. Figure 2 represents the average error variations versus cloud factor for three types of approximations. Average error is defined as the summation of the absolute values of errors in nodes divided by the number of distributed nodes in the domain and cloud factor is a coefficient that the size of clouds dilates, if this factor increases.

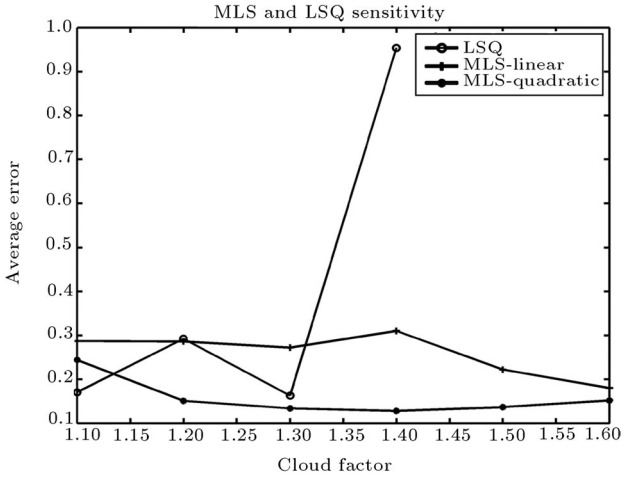

Figure 2. Type of approximation effect.

Figure 2 manifests that LSQ approximation cannot achieve reliable results in large cloud factors. In contrast, MLS can work well and will be better with quadratic basis functions.

FPM satisfies equations locally and, from this viewpoint, is similar to FDM, so their comparison can be good for validating FPM accuracy. Again, consider the problem mentioned in Equations 59. This problem was solved by FPM and FDM on similar node distributions. Figure 3 represents the average error of these methods versus the number of nodes. In this figure, the grid independency of FDM and FPM is observed. In addition, it manifests that the convergence rate in FPM is similar to FDM and, furthermore, the results of FPM with quadratic based functions are more accurate than

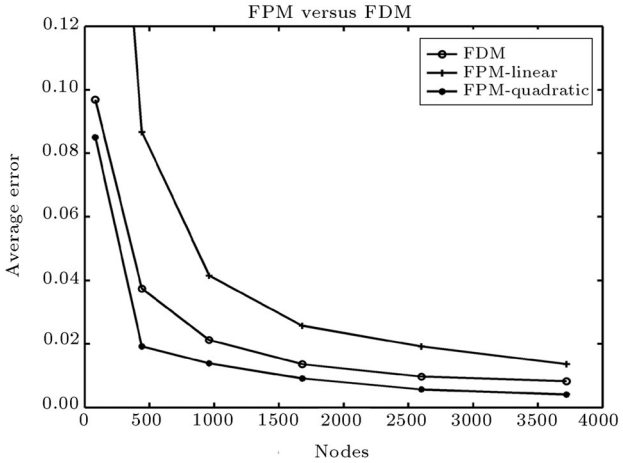

Figure 3. Comparison of FPM and FDM.

FDM results. These results are interesting, because the complicated mathematics of least-squares problems prevents them from being interpreted mathematically.

Heat conduction problems with different boundary conditions and variable sources can be handled by FPM readily and accurate solutions are resulted if relevant approximation is adapted.

Stokes Problem

Generally, the Stokes problem is one of the first challenging problems to be solved in testing methods. The solution has symmetric pressure contours and the positions of the maximum and minimum values of pressure are specified theoretically. In addition, these values should tend to extreme values, if the number of nodes is increased. The Stokes problem has been solved before in [28], but in this paper, other methods are also experimented.

This problem is expressed as the following equations:

$$-\nabla p + \mu \nabla^2 \mathbf{u} = 0,$$

$$0 \le x \le 1, \quad 0 \le y \le 1, \tag{61a}$$

$$u_i(x,0) = u_i(0,y) = u_i(1,y) = 0,$$

$$0 \le x \le 1, \quad 0 \le y \le 1, \tag{61b}$$

$$u_1(x,1) = 1, \qquad 0 < x < 1, \tag{61c}$$

$$u_2(x,1) = 0, \qquad 0 < x < 1. \tag{61d}$$

The above problem has been solved by the SSUPG and FIC methods using the FS algorithm. It is important to denote that in the SSUPG method, the stabilization parameter is computed according to the method represented by Franca and Frey [29] for determining the SUPG stabilization parameter. This method of calculating the stabilization parameter works well for the SSUPG method in FPM.

Both of these methods result in satisfactory solutions. Figure 4 represents one of these results on 961 sets of regular distributed nodes. One of the important results is the effect of the number of nodes on the maximum (or minimum) pressure. This effect is depicted in Figure 5.

Navier-Stokes Calculations

The final objective of this paper is the Navier-Stokes calculations. Navier-Stokes equations are non-self-adjoint coupled partial differential equations for which many ordinary numerical methods are not applicable.

In this paper, Navier-Stokes equations are solved by FPM using the following schemes:

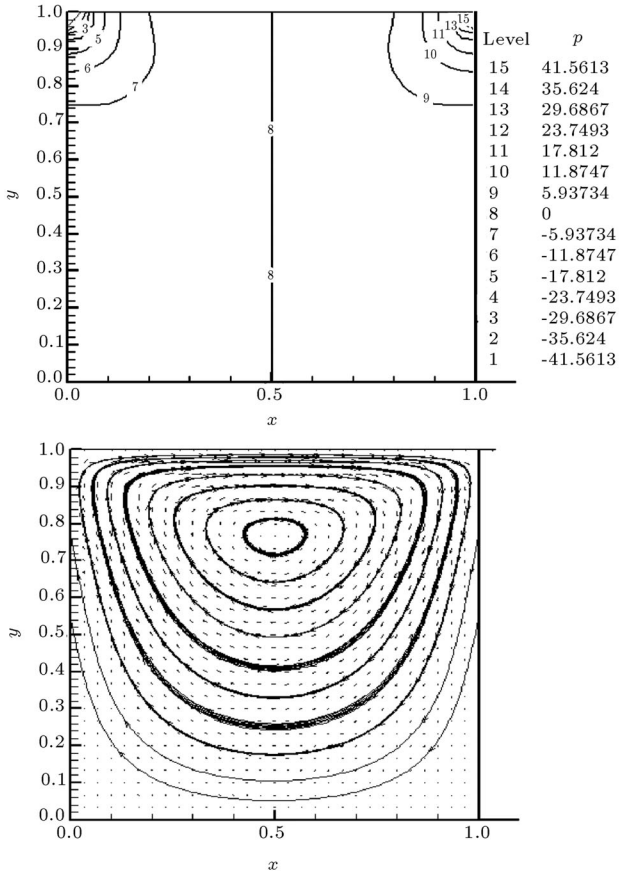

Figure 4. Stokes solution on cavity by FPM.

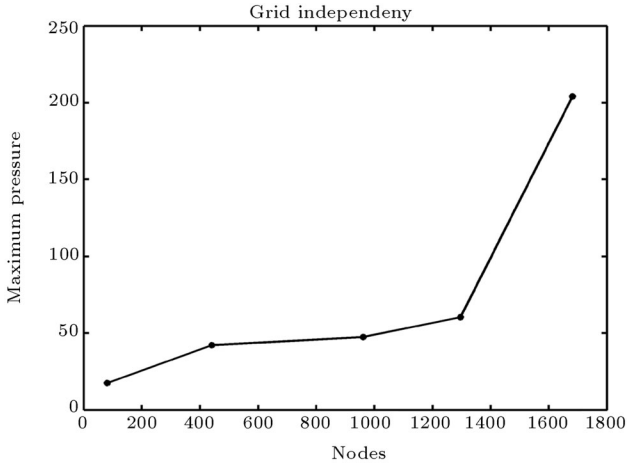

Figure 5. Effect of number of nodes.

1. The SSUPG stabilization method using Franca and Frey stabilization parameter calculation [29];
2. The FIC stabilization method and the FS time marching algorithm.

Applying SSUPG for FPM was first proposed by Hannani and Parsinejad [28]. The second scheme was first represented by Onate et al. [18,19]. Onate, in [17], introduced the FIC stabilization scheme and the FS algorithm was introduced by Zienkiewicz et al. [26,27].

FPM and FEM in Parallel Plates

One of the classical problems for validating the accuracy of numerical methods is the incompressible fluid flow between parallel plates. This problem has an analytical solution far from the entry, as below:

$$u = \frac{3}{4}\frac{q}{h}\left[1 - \left(\frac{y}{h}\right)^2\right], \tag{62a}$$

$$\frac{dp}{dx} = -\frac{3}{2}h^{-3}q\mu, \tag{62b}$$

where, in the above equations, u is the velocity along the plates direction, q is the flow per unit width, h is the half distance between plates, y is the distance from the symmetry line and μ is the viscosity of the fluid.

The numerical solution is implemented on the domain represented in Figure 6. This problem was solved for Reynolds number 100 with various node distributions by the SSUPG stabilization method. The minimum number of nodes in each cloud is nine, the basis functions are quadratic and the approximation method is MLS.

The mentioned problem has two analytical values that are useful for the validation of the numerical method. The first one is the variation of the horizontal velocity on the symmetry line. This value should tend to 1.5, asymptotically. The second value is the slope of the pressure graph far from the flow entry. The value for this domain and boundary conditions is obtained as 0.12.

Figure 7 represents horizontal velocity variations along the symmetry line using 1346 irregular distributed nodes on the domain. From Figure 7, the asymptotic tendency of the horizontal velocity on the symmetry line is obvious, but, there are two uncertainties in this result. The horizontal velocity on the symmetry line oscillates and its asymptotic value is less than 1.5.

The reason for this situation is that the distribution of nodes is irregular and, therefore, there are not any nodes exactly on the symmetry line. On the other hand, maximum velocity occurs on the symmetry line and when the value of the horizontal velocity on the symmetry line is calculated by interpolation from

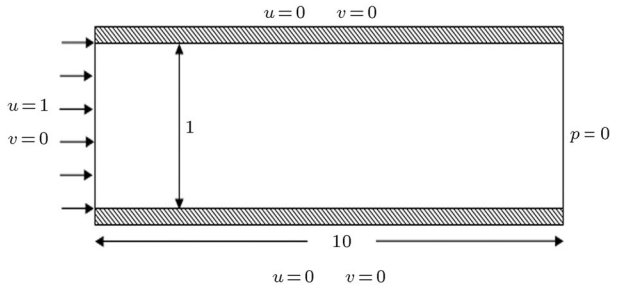

Figure 6. Parallel plates problem configuration.

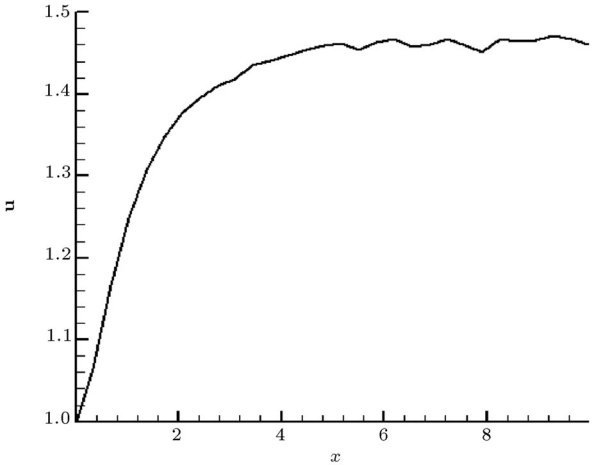

Figure 7. Velocity on symmetry line, 1346 irregular nodes.

neighboring nodes, its value is obtained less than the correct value. Likewise, when there are any nodes near the symmetry line, the value of the interpolated velocity is obtained higher and vice versa and this condition causes oscillation. It has to be mentioned that FEM suffers from the same drawback.

If nodes are distributed in the domain such that one has some nodes exactly on the symmetry line, oscillations will be suppressed. In the represented result in Figure 8, node distribution includes 1210 nodes and there are some points on the symmetry line. It is obvious that there are not any oscillations and the asymptotic value is 1.5, accurately.

The second analytical value is the pressure gradient in the fully developed region. Table 1 includes some values of pressure gradients that have been obtained by FPM and FEM on different node distributions. Regular node distribution means that there are some nodes on the symmetry line. In order to compare these results, node distributions for an equal number of nodes are similar for FPM and FEM. From this table, FPM results are more accurate than FEM results.

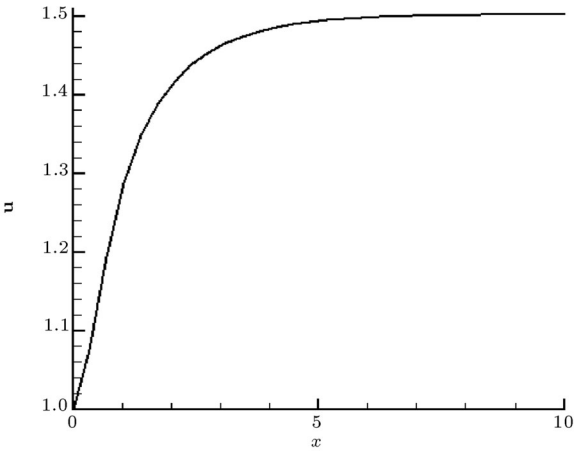

Figure 8. Velocity on symmetry line, 1210 regular nodes.

Table 1. Validation of FPM results for parallel plates problem.

Solution Method	dp/dx	Error%
Exact	-0.12	0.00
FPM Using 958 Irregular Nodes	-0.1128	6.00
FPM Using 1346 Irregular Nodes	-0.1180	1.67
FPM Using 810 Regular Nodes	-0.1237	3.09
FPM Using 1210 Regular Nodes	-0.1203	0.24
FEM Using 958 Irregular Nodes	-0.1091	9.04
FEM Using 1346 Irregular Nodes	-0.1139	5.12

The accuracy of the applied method was compared with other existing results. In Figure 9, pressure variations on the symmetry line are represented. This figure includes Shah's experimental results [30], Schmidt's numerical results [31] using FDM and the result of FPM, which shows its reliability.

FPM and FEM in Backward Facing Step

Incompressible fluid flow over a backward-facing step is another classical problem that, due to the existence of reliable experimental and numerical results, can be a good criterion for validation of a numerical technique. Figure 10 represents the assumed configuration and boundary conditions.

Both of the mentioned stabilization schemes, SSUPG and FIC, using the FS algorithm, can solve this problem. In addition, with the FIC method, transient solutions can be achieved.

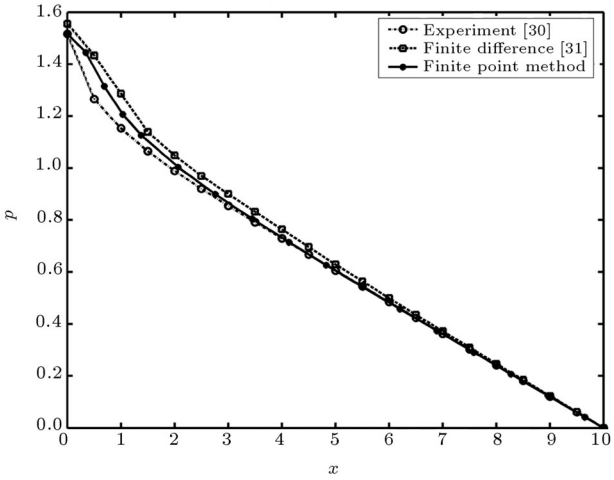

Figure 9. Comparison of FPM and validated results based on pressure variations on the symmetry line.

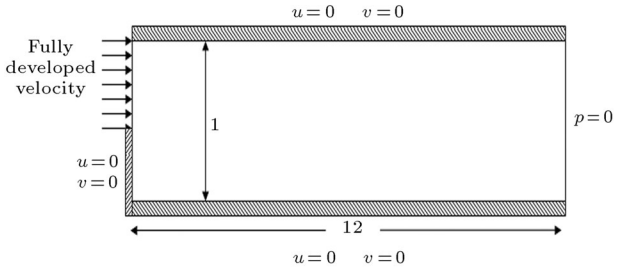

Figure 10. Backward-facing step configuration.

In the SSUPG method, the stabilization parameter is calculated from Franca and Freys suggestion. In computing stabilization parameters for FIC, the SUPG assumption is applied and the value of h is considered as half the distance between the star node and the closest neighboring node. In each cloud, the minimum number of nodes is nine and the basis functions are quadratic. As in the previous problem, the MLS approximation method is applied.

Calculations are done for Reynolds numbers 100 and 400. The Reynolds number is based on bulk velocity at the inlet boundary and the cross section width of the whole domain and fully developed flow is assumed at the inlet. Figures 11 and 12 include these results. Distributed nodes on the domain are 989 for both results and this node distribution is depicted in Figure 11.

In addition, the transient solution of incompressible fluid flow over a backward-facing step is resulted from the FIC stabilization and the FS time marching algorithm. Figure 13 represents some of these results. These results are similar to the FEM solution of this

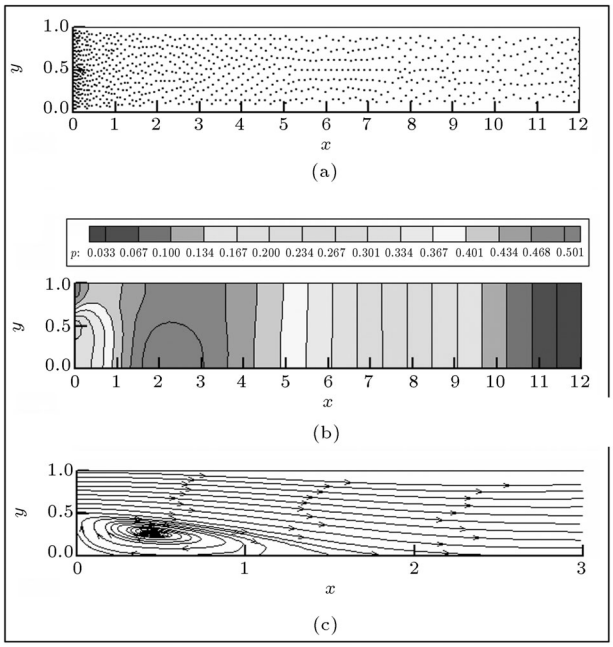

Figure 11. FPM results, 989 nodes, Re = 100, SSUPG stabilization.

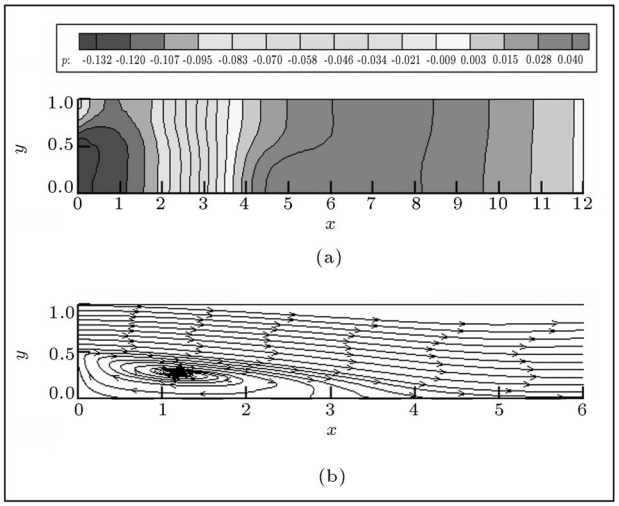

Figure 12. FPM results, 989 nodes, Re = 400, SSUPG stabilization.

problem on the same node distribution with triangular elements, using the SUPG stabilization scheme [32].

In Figure 14 the results of FEM using SUPG, FPM using the SSUPG stabilization scheme and FPM using the FIC stabilization method and the FS time marching algorithm are compared. This figure represents pressure variations on the bottom wall.

The FPM solution, using FIC and FS methods, results in a satisfactory solution, compared with the FEM solution, and their maximum difference in this figure is about three percent. On the other hand, the FPM solution using SSUPG obtains good results, but not as good as FPM using FIC and FS methods. The maximum difference between FPM using SSUPG and FEM is about seven percent. Generally, all of the above methods have almost similar results, but, in the region near to the flow inlet, because of the presence of a vortex region and the existence of two singular points,

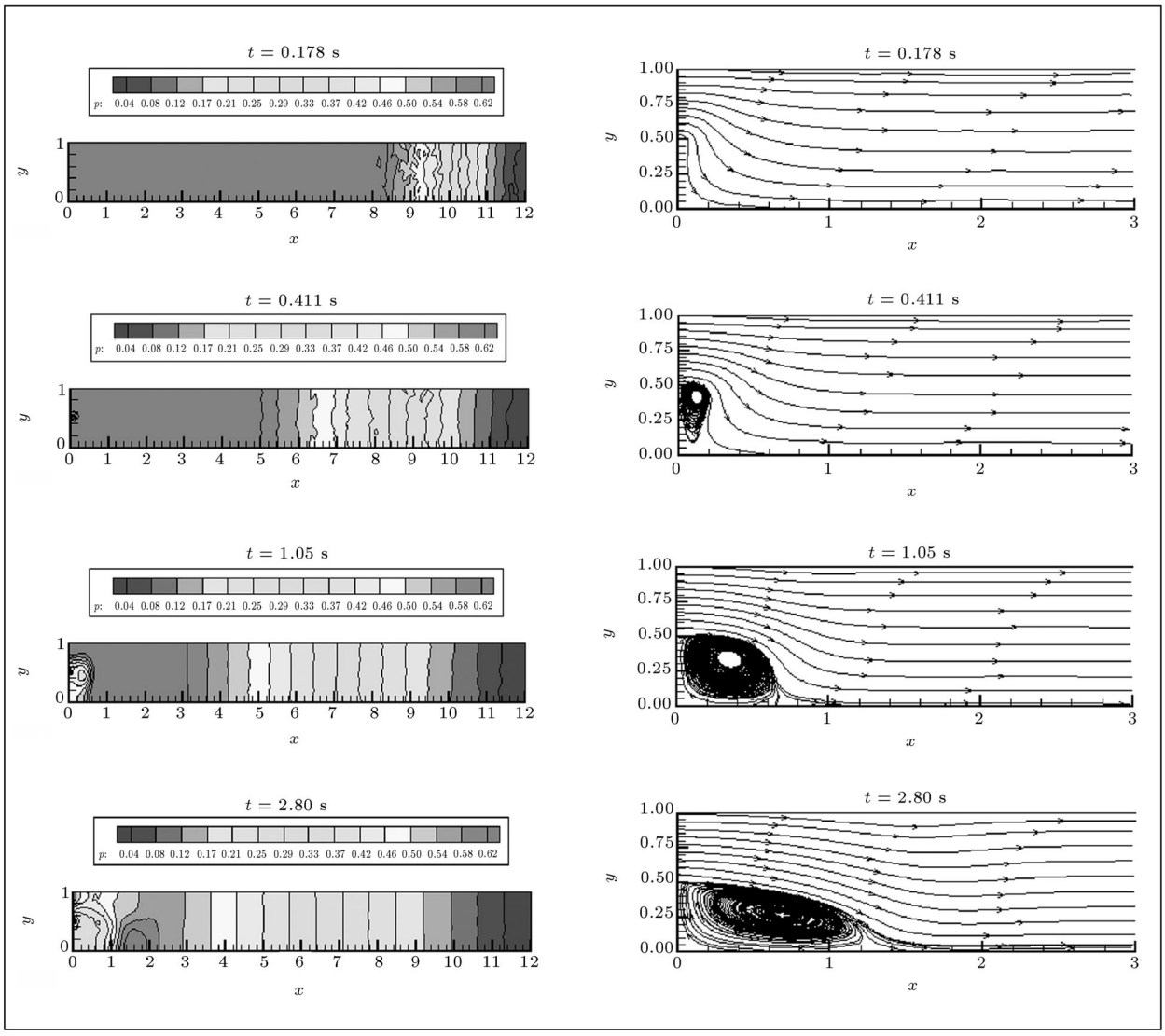

Figure 13. FPM transient solution, 989 nodes, Re=100, FIC stabilization, FS algorithm.

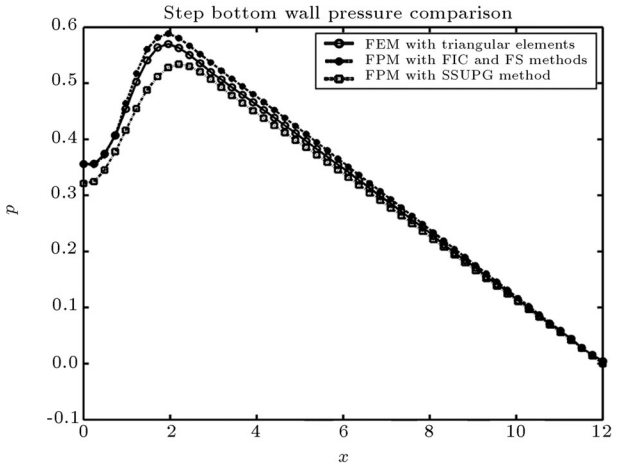

Figure 14. Comparison of FPM and FEM based on pressure variation on bottom wall.

the mesh independent solution is not obtained and the solutions differ from each other.

One of the important properties of the FIC stabilization scheme is that the stabilized equations are consistent, so, the real solution satisfies the stabilized form of the governing equations. Nevertheless, the SSUPG stabilization scheme has not this characteristic. Instead, the SSUPG method converges better than the FIC method and convergence in FIC occurs under harder conditions.

Another criterion for studying FPM accuracy is to compare it with the experimental results of Armaly et al. [33] or the numerical results of Hannani et al. [34]. Both of these reliable results pointed out that in Reynolds number 400, the horizontal distance between flow inlet to flow attachment on the bottom wall is 4.2 times the cross section width of the whole domain. Table 2 includes the position of this point, based on various FPM and FEM solutions.

Table 2 emphasizes the accuracy of the FIC stabilized FPM scheme in comparison with the SSUPG stabilized FPM method and shows the agreement between FPM using FIC and FEM. However, for Reynolds number 400, all of the results have an indispensable difference with the correct value, the result of FPM using FIC is better.

The most important problem that produces this difference is that there are not enough nodes in the domain of solution, especially near the boundary. As can be observed, the result of the FEM solution with the SUPG stabilization scheme is similar to the FPM solution using FIC and this proves that the source of the mentioned difference is the lack of enough distributed nodes in the domain. In FEM, this dilemma is solved by applying rectangular elements that are stretched along the flow direction and this causes better results. This method cannot be applied in the case of complicated flows.

Table 2. Comparison of FPM and FEM.

Solution Method	Re = 100	Re = 400
FPM and SSUPG	1.3	3.0
FPM and FIC and FS	1.4	3.8
FEM and SUPG	1.4	3.7

In FPM, another method can be used that works for general problems. Because nodes are not limited by any elements, one can consider their interaction similar to reality.

One of the important properties of each numerical method is the required time versus their accuracy. The required time for FPM is in the order of other unstructured methods and its accuracy is almost equal to regular FEM. This is very interesting for any method that does not use any grid and, therefore, it has great potential in many applications, such as adaptive numerical techniques or moving-boundary problems.

CONCLUSIONS

In this paper, the finite point method was studied and its stabilization was reviewed. This method is a truly meshless method that requires only the spatial position of distributed nodes in the domain.

The approximation method is the basis of each meshless method. On this subject, the results confirm the conclusions mentioned in [1,15].

FPM and FDM results were compared for a heat conduction problem and it was shown that FPM can produce reliable results. Furthermore, with an efficient approximation scheme, FPM accuracy is better than FDM accuracy.

The main goal of this paper was to solve the incompressible Navier-Stokes equations. This issue was implemented after solving a Stokes problem in a cavity for incompressible fluid and, then, flows between parallel plates and over backward-facing steps were analyzed. The FPM method can solve non-self-adjoint differential equations such as Navier-Stokes equations, if an efficient stabilization scheme is applied. In this field, the following results were obtained:

1. SSUPG and FIC methods can be used for stabilization of governing equations;
2. Results of FPM are as good as FEM;
3. Neumann boundary conditions should be stabilized;
4. Essential boundary conditions can be enforced directly according to the approximation;
5. Fractional step algorithms are applicable for transient solutions.

The general advantages of the FPM method in comparison with other numerical methods are:

1. Satisfactory accuracy is obtained;
2. The method is applicable in three-dimensional and complicated domains;
3. The approximation method used in FPM has some parameters that can be tuned, depending on the type of problem, such as weighting function properties, criteria for selecting passive nodes, basis functions etc;
4. The possibility of introducing new nodes independent of the position of existing ones.

Meshless methods are a new field in numerical methods and have unsolved problems, which need further studies. The following fields are suggested as interesting subjects to be investigated:

1. Time-consuming: Meshless method is interesting from many view points, but the time-consuming issue should be studied rigorously by suitable 2D and 3D benchmark problems;
2. Applying goal oriented approximation parameters: These parameters can be tuned such that they inject special characteristics, for example, using basis functions that have any relation with governing physics results in better solutions;
3. Using adaptive clouds: In the FPM, nodes are not limited by any elements, so one can consider their interaction similar to reality. For example, if elliptical clouds are used instead of circular clouds, such that the clouds contain upwind nodes of the star nodes, the results improve. Furthermore, the slenderness of the ellipse of cloud can be adapted as a function of velocity at the star node. These modifications improve the accuracy of FPM;
4. Discretization based on integral equations: Local satisfaction of equations increases errors and instability resources. Satisfaction of the integral form of governing equations may result in more stable and more accurate solutions.

REFERENCES

1. Onate, E., Idelsohn, S., Zienkiewicz, O.C. and Taylor, R.L. "A finite point method in computational mechanics: Applications to convective transport and fluid flow", *Int. J. Num. Methods Fluids*, **39**, pp 3839-3866 (1996).
2. Griebel, M. and Schweitzer, M.A., *Meshfree Methods for Partial Differential Equations. Lecture Notes in Computational Science and Engineering*, Springer, New York, USA (2003).
3. Belytschko, T., Krongauz, Y., Organ, P., Flemming, M. and Krysl, P. "Meshless methods: An overview and recent developments", *Comput. Methods Appl. Mech. Eng.*, **139**, pp 3-47 (1996).
4. Jensen, P.S. "Finite difference techniques for variable grids", *Comput. Struct.*, **2**, pp 17-29 (1972).
5. Perrone, N. and Kao, R. "A general finite difference method for arbitrary meshes", *Comput. Struct.*, **5**, pp 45-47 (1975).
6. Frey, W.H. "Flexible finite difference stencils from iso-parametric finite elements", *Int. J. Num. Methods Eng.*, **11**, pp 1653-1665 (1977).
7. Liszka, T. and Orkisz, J. "The finite difference methods at arbitrary irregular grids and its application in applied mechanics", *Comput. Struct.*, **11**, pp 83-95 (1980).
8. Gingold, R.A. and Moraghan, J.J. "Smoothed particle hydrodynamics: Theory and applications to non-spherical stars", *Man. Not. Roy. Astron. Soc.*, **181**, pp 375-389 (1977).
9. Nayroles, B., Touzot, G. and Villon, P. "Generalizing the FEM: Diffuse approximation and diffuse elements", *Comput. Mech.*, **10**, pp 307-318 (1992).
10. Belytschko, T., Lu, Y. and Gu, L. "Element free Galerkin methods", *Int. J. Num. Methods Eng.*, **37**, pp 229-256 (1994).
11. Lu, Y.Y., Belytschko, T. and Gu, L. "A new implementation of element free Galerkin method", *Comput. Methods Appl. Mech. Eng.*, **113**, pp 397-414 (1994).
12. Duarte, C.A. and Oden, J.T. "hp clouds: A meshless method to solve boundary value problems", *TICAM Report*, pp 95-05 (1995).
13. Liu, W.K., Jun, S. and Belytschko, T. "Reproducing kernel particle methods", *Int. J. Num. Methods Fluids*, **20**, pp 1081-1106 (1995).
14. Liu, W.K. and Chen, Y. "Wavelet and multiple scale producing kernel methods", *Int. J. Num. Methods Fluids*, **21**, pp 901-933 (1995).
15. Onate, E., Idelsohn, S., Zienkiewicz, O.C., Taylor, R.L. and Sacco, C. "A stabilized finite point method for analysis of fluid mechanics problems", *Comput. Methods Appl. Mech. Eng.*, **139**, pp 315-346 (1996).
16. Onate, E. and Idelsohn, S. "A mesh-free finite point method for advective-diffusive transport and fluid flow problems", *Comput. Mech.*, **21**, pp 283-292 (1998).
17. Onate, E. "Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems", *Comput. Methods Appl. Mech. Eng.*, **151**, pp 233-265 (1998).
18. Onate, E., Sacco, C. and Idelsohn, S. "Advances in the finite point method for meshless analysis of incompressible flow problems", *ECCOMAS Computational Fluid Dynamics Conference*, Wales, UK (Sept. 2001).
19. Onate, E., Sacco, C. and Idelsohn, S. "A finite point method for incompressible flow problems", *Comput. Vis. Sci.*, **3**, pp 67-75 (2000).
20. Atluri, S.N. and Zhu, T. "New concepts in meshless methods", *Int. J. Num. Methods Eng.*, **47**, pp 537-556 (2000).

21. Zhang, X., Liu, X.H., Song, K.Z. and Lu, M.W. "Least squares collocation meshless method", *Int. J. Num. Methods Eng.*, **51**, pp 1089-1100 (2001).

22. Idelsohn, S. and Onate, E. "Finite element and finite volume, two good friends", *Int. J. Num. Methods Eng.*, **37**, pp 3323-3341 (1994).

23. Zeinkiewicz, O.C. and Taylor, R.L., *The finite element method*, McGraw-Hill, New York, **I** (1989), **II** (1991).

24. Hughes, T.J.R. and Mallet, M. "A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive system", *Comput. Methods Appl. Mech. Eng.*, **58**, pp 305-328 (1986).

25. Onate, E. "A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation", *Comput. Methods Appl. Mech. Eng.*, **182**, pp 355-370 (2000).

26. Zienkiewicz, O.C. and Codina, R. "A general algorithm for compressible and incompressible flow. Part I: The split characteristic-based scheme", *Int. J. Num. Methods Fluids*, **20**, pp 869-885 (1995).

27. Codina, R., Vasquez, M. and Zienkiewicz, O.C. "A general algorithm for compressible and incompressible flow. Part III: The semi-implicit form", *Int. J. Num. Methods Fluids*, **27**, pp 13-32 (1998).

28. Hannani, S.K. and Parsinejad, F. "Incompressible Stokes flow calculation using a finite point method", *J. Scientica Iranica*, **10**, pp 1-13 (2003).

29. Franca, L.P. and Frey, S.L. "Stabilized finite element method: II. The incompressible Navier-Stokes equation", *Comput. Methods Appl. Mech. Eng.*, **99**, pp 209-233(1992).

30. Shah, R.K. "A correlation for laminar hydrodynamic entry length solutions for circular and noncircular ducts", *J. Fluids Eng.*, **100**, pp 177-179 (1978).

31. Schmidt, F.W. and Zeldin, B. "Laminar flow in the inlet section of tubes and ducts", *AIChE J.*, **15**, pp 612-614 (1969).

32. Sadeghi, M.M. "Navier-Stokes calculations using a meshless method", M.Sc. Thesis, Sharif University of Technology, Tehran, Iran (2003).

33. Armaly, B.F., Durst, F., Pereira, J.C.F. and Schönung, B. "Experimental and theoretical investigations of backward-facing step flow", *J. Fluid Mech.*, **127**, pp 473-496 (1983).

34. Hannani, S.K., Stanislas, M. and Dupont, P. "FE simulation of backward-facing step with stabilized bilinear and biquadratic equal-order-interpolation velocity-pressure elements", *Advances in Finite Element Analysis in Fluid Dynamics, ASME Conference*, Winter Annual Meeting, Chicago, USA, **200**, pp 1-9 (1994).