INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1885–1907
Published online 16 October 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1529

On the use of characteristic-based split meshfree method for solving flow problems

Abazar Shamekhi*,† and Kayvan Sadeghy

Faculty of Mechanical Engineering, University of Tehran, P.O. Box 11365-4563, Tehran, Iran

SUMMARY

This study presents characteristic-based split (CBS) algorithm in the meshfree context. This algorithm is the extension of general CBS method which was initially introduced in finite element framework. In this work, the general equations of flow have been represented in the meshfree context. A new finite element and MFree code is developed for solving flow problems. This computational code is capable of solving both time-dependent and steady-state flow problems. Numerical simulation of some known benchmark flow problems has been studied. Computational results of MFree method have been compared to those of finite element method. The results obtained have been verified by known numerical, analytical and experimental data in the literature. A number of shape functions are used for field variable interpolation. The performance of each interpolation method is discussed. It is concluded that the MFree method is more accurate than FEM if the same numbers of nodes are used for each solver. Meshfree CBS algorithm is completely stable even at high Reynolds numbers. Copyright © 2007 John Wiley & Sons, Ltd.

Received 20 July 2006; Revised 24 December 2006; Accepted 20 April 2007

KEY WORDS: meshfree method; fluid flow; characteristic-based split algorithm; finite element method; fluid dynamics

1. INTRODUCTION

Simulation of physical phenomena requires solving partial differential equations that govern those problems. It is not possible to find exact mathematical solution for some partial differential equations. To solve the problem, scientists often use numerical and computational solutions. Over the years many numerical methods have been developed such as finite difference method, finite volume method and finite element method. In these methods, the spatial domain where the partial differential governing equations are defined is often discretized into meshes. Using simple mathematical function for interpolation over every small mesh, partial differential equation can easily be changed

*Correspondence to: Abazar Shamekhi, Faculty of Mechanical Engineering, University of Tehran, P.O. Box 11365-4563, Tehran, Iran.
†E-mail: ashamekh@ut.ac.ir

to a system of algebraic equations. Creation of suitable meshes is very essential for acquiring accurate results. Mesh generation process is so time consuming that engineering scientists decided to develop meshless numerical methods to solve partial differential equations. The meshfree or MFree method is defined as a method for numerically solving partial differential equations without the use of any predefined mesh. MFree methods only use a set of nodes scattered within the problem domain as well as sets of nodes scattered on the boundaries of the domain to represent the problem domain and its boundaries [1]. The meshfree methods appear particularly effective for three-dimensional irregular domains and moving boundary problems where mesh generation can be very expensive.

There are some meshfree methods which have been developed by now such as element-free Galerkin method (EFG), meshless local Potrove Galerkin method (MLPG) and smooth particle hydrodynamics or SPH method. Generally, there are two types of meshfree method, weak-form meshfree method such as EFG method and strong-form meshfree method such as SPH method. The SPH method one of the pioneering meshfree methods, was initially introduced by Lucy, Gingold and Monoghan to solve three-dimensional astrophysical problems [2, 3]. To deal with complex problems having moved boundaries, the idea of using meshfree method in engineering problems has been arisen for eliminating meshing and simplifying refining processes and consequently, lowering computational cost. In mesh generation point of view, meshfree method can be categorized into the domain discretization methods such as FDM and the boundary only discretization methods such as meshless boundary methods [4–6]. Meshfree method has been employed by many authors for solving engineering problems [7–9]. Among them numerical simulations of flow problems are very challengeable due to the nonlinear behaviour of their governing equations. These complexities cause the flow patterns to be completely different in various velocities and geometries, so that many algorithms are not suitable enough to handle all phenomena happening in fluid dynamics. The nonlinearity of fluid dynamics equations is due to the derivation of fluid dynamics equation in Eulerian view. Fluids continuously tend to deform when they are subjected to deviatoric stresses, therefore some researchers prefer to use fluid dynamics equations in Eulerian view. In Eulerian view, the major difference between fluid dynamics equations and the governing equations of solids which are derived in Lagrangian view is the presence of convective term in its equations. These terms cause the equation of fluid dynamics not to be self-adjoint. Therefore, for convection-dominant problems, the Galerkin procedure is no longer optimal and it is here that most of the fluid mechanics problems lie.

For fluid dynamics problems, the meshfree methods have recently been employed. Liu *et al.* [10] used the reproducing kernel particle method (RKPM) with SUPG formulation to solve two-dimensional advection–diffusion equation. Sadat and Couturier [11] employed the diffuse element method (DEM) with the project method to study the laminar natural convection problem. Yagawa and Shirazaki [12] applied free mesh method (FMM) with the weighed residual-Galerkin method to unsteady two-dimensional incompressible viscous flow. Cheng and Liu [13] adopted the finite point method (FPM) with the discretization defined by the positions of points to analyse two-dimensional driven cavity flow. Kim and Kim [14] presented some analyses of fluids by meshfree point collocation method (MPCM). A new RBF collocation scheme and kernel radial basis functions (RBFs) have been discussed by Chen. The kernel RBF which he applied in his work is a characteristic meshfree shape function [15].

This study presents characteristic-based split (CBS) algorithm in the meshfree context. This algorithm is the extension of general CBS method which was initially introduced by Zienkiewicz and Codina in finite element framework [16]. The foremost advantage of this method is the

DOI: 10.1002/fld

capability of solving either incompressible or compressible subsonic and supersonic flows by the same algorithm [16].

In this work, the general equations of flow have been discretized in the meshfree context. A finite element and MFree code is developed for solving flow problems. This computational code is capable of solving both time-dependent and steady-state flow problems. Numerical simulation of some known benchmark flow problems has been studied. Computational results of MFree method have been compared to those of finite element method. The results obtained, have been verified by known numerical and experimental data in the literature. A number of shape functions are used for field variable interpolation. The performance of each interpolation method is discussed.

2. NUMERICAL FORMULATION

2.1. General equations

The general equations of fluid mechanics, with small modifications, can be written as follows [17]:
Mass conservation:

$$\frac{\partial \rho}{\partial t} = \frac{1}{c^2}\frac{\partial p}{\partial t} = -\frac{\partial U_i}{\partial x_i} \tag{1}$$

Momentum conservation:

$$\frac{\partial U_i}{\partial t} = -\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i} - \rho g_i \tag{2}$$

In the above, we define the mass flow fluxes as

$$U_i = \rho u_i \tag{3}$$

Energy conservation:

$$\frac{\partial(\rho E)}{\partial t} = -\frac{\partial}{\partial x_j}(u_j \rho E) + \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) - \frac{\partial}{\partial x_j}(u_j p) + \frac{\partial}{\partial x_i}(\tau_{ij} u_j) \tag{4}$$

In all of the above equations u_i is the velocity component, ρ is the density, E is the specific energy, p is the pressure, T is the absolute temperature, ρg_i represents body forces and other source terms, k is the thermal conductivity, and τ_{ij} are the deviatoric stress components given by the following equation:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{5}$$

where δ_{ij} is the Kroenecker delta $\delta_{ij} = 1$, if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. The equations are completed by the universal gas law when the flow is coupled and compressible:

$$p = \rho R T \tag{6}$$

where R is the universal gas constant.

DOI: 10.1002/fld

2.2. Characteristic-based split algorithm

The split process was initially introduced by Chorin [18] for incompressible flow problems in the finite difference framework. After that split method was extended to finite element context and employed for different applications of incompressible flows [19–21]. However, the algorithm in its full form was first introduced in 1995 by Zienkiewicz and Codina [16] to solve the fluid dynamics equations of both compressible and incompressible flows. The foremost advantage of this method is the capability of solving either incompressible or compressible subsonic and supersonic flows by the same algorithm [17].

2.3. The split temporal discretization

Using Equation (2), the momentum conservation equation can be written as follows [17]:

$$\frac{\partial U_i}{\partial t} = -\frac{\partial}{\partial x_j}(u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + Q_i^{n+\theta_2} \tag{7}$$

with $Q^{n+\theta_2}$ being treated as a known quantity evaluated at $t = t^n + \theta_2 \Delta t$ in a time increment Δt. In the above equation

$$Q_i^{n+\theta_2} = -\frac{\partial p^{n+\theta_2}}{\partial x_i} \tag{8}$$

with

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = \theta_2 \frac{\partial p^{n+1}}{\partial x_i} + (1-\theta_2)\frac{\partial p^n}{\partial x_i} \tag{9}$$

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \tag{10}$$

In this

$$\Delta p = p^{n+1} - p^n \tag{11}$$

Using the characteristic-Galerkin process, Equation (7) can be rewritten as follows [17]:

$$U^{n+1} - U^n = \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i)^n + \frac{\partial \tau_{ij}^n}{\partial x_j} + Q_i^{n+\theta_2} - (\rho g_i)^n + \left(\frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j}(u_j U_i) - Q_i + \rho g_i \right) \right)^n \right] \tag{12}$$

At this stage by using the split procedure a suitable approximation for Q is substituted into the equations which allows the calculation to proceed before p^{n+1} is evaluated. Two alternative approximations are useful as Splits A and B, respectively. In this paper, Split A was used. In this method, an auxiliary variable U_i^* is defined such that [17]:

$$\Delta U_i^* = U_i^* - U_i^n = \Delta t \left[-\frac{\partial}{\partial x_j}(u_j U_i)^n + \frac{\partial \tau_{ij}^n}{\partial x_j} - (\rho g_i)^n + \left(\frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j}(u_j U_i)_i + \rho g_i \right) \right)^n \right] \tag{13}$$

DOI: 10.1002/fld

This equation will be solved subsequently by an explicit time step applied to the discretized form and a complete solution is now possible. The 'correction' given below is available once the pressure increment is evaluated:

$$\Delta U_i = U_i^{n+1} - U_i^n = \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} - \frac{\Delta t^2}{2} u_k \frac{\partial Q_i^n}{\partial x_k} \tag{14}$$

From Equation (1), we have

$$\Delta\rho = \left(\frac{1}{c^2}\right)^n \Delta p = -\Delta t \frac{\partial U_i^{n+\theta_1}}{\partial x_i} = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i}{\partial x_i}\right] \tag{15}$$

Replacing U^{n+1} by the known intermediate, auxiliary variable U^* and rearranging after neglecting higher-order terms ($\Delta t^3, \Delta t^4, \ldots$), we have [17]

$$\Delta\rho = \left(\frac{1}{c^2}\right)^n \Delta p = -\Delta t \left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \left(\frac{\partial^2 p^n}{\partial x_i \partial x_i} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i \partial x_i}\right)\right] \tag{16}$$

The above equation is fully self-adjoint in the variable Δp or $\Delta\rho$ which is the unknown. Now a standard Galerkin-type procedure can be optimally used for spatial approximation. After completing the calculation to establish ΔU_i, and Δp (or $\Delta\rho$) the energy equation is dealt with independently and the value of $(\rho E)^{n+1}$ is obtained by the characteristic-Galerkin process applied to Equation (4).

Choosing different values of θ_1, θ_2, the above equations can be solved in the form of explicit, semi-implicit and nearly implicit. In fully explicit form, $\frac{1}{2} \leqslant \theta_1 \leqslant 1$ and $\theta_2 = 0$. In semi-implicit form, $\frac{1}{2} \leqslant \theta_1 \leqslant 1$, $\frac{1}{2} \leqslant \theta_2 \leqslant 1$. Finally, in nearly implicit form both parameters θ_1 and θ_2 are unit [17]. It is clear that the governing equations can be solved after spatial discretization.

2.4. Spatial discretization using meshfree method

The meshfree method is used to establish a system of algebraic equations for the whole problem domain without the use of a predefined mesh. MFree methods use a set of nodes scattered within the problem domain as well as sets of nodes scattered on the boundaries of the domain to represent the problem domain and its boundaries. These sets of scattered nodes do not form a mesh, which means that no information on the relationship between the nodes is required, at least for field variable interpolation. Therefore, construction of shape functions is the major part of MFree methods.

2.4.1. Shape functions construction. Construction of MFree shape functions is the central and most important topic in MFree methods. The challenge is how to create shape functions using only nodes scattered arbitrarily in a domain without any predefined mesh to provide connectivity of the nodes. Development of more effective methods for creating shape functions is thus one of the newest areas of research in the area of MFree methods. The size of domain that is used for interpolation and consequently the number of nodes that is used for shape function construction in each arbitrary point in problem domain must be very small compared with the whole problem domain. Therefore, a support domain is defined as a small arbitrary area in each point. The shape of support domains can be any arbitrary suitable simple shape such as circle or rectangle. The size of support domains can be adequately chosen by analyst for every problem. Figure 1 shows a problem domain with some kinds of arbitrary support domains.

DOI: 10.1002/fld

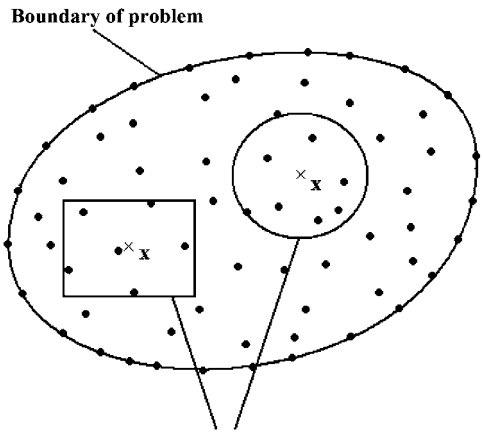

Figure 1. Some kinds of support domain definition.

A number of ways to construct shape functions have been developed such as SPH method by Lucy [2] and Gingold and Monaghan [3] and RKPM by Liu *et al.* [22]. These methods are classified in finite integral representation category. In finite series representation form, there are some procedures such as moving least squares (MLS) method which was developed by Lancaster and Salkauskas [23], PIM by Liu and Gu [24] and RPIM by Wang and Liu [25]. In this paper, we describe MLS procedure in brief. For more information about shape function constructions in meshfree method, reader can refer the literature [1].

2.4.2. Moving least squares approximation. MLS was initially introduced for data fitting and surface construction in 1981 [23]. MLS was used for shape function construction in DEM by Nayroles *et al.* [26]. The MLS approximation has two major features that make it popular: (1) the approximated field function is continuous and smooth in the entire problem domain and (2) it is capable of producing an approximation with the desired order of consistency [1].

2.4.3. MLS Procedure. Let $u(x)$ be the function of the field variable defined in the problem domain. The approximation of $u(x)$ at point x is denoted $u^h(x)$. MLS approximation first writes the field function in the form [1]

$$u^h(x) = \sum_j^m p_j(x) a_j(x) \equiv p^{\mathrm{T}}(x) a(x) \tag{17}$$

where x is vector of position and m is the number of terms of monomials (polynomial basis), and $a(x)$ is a vector of coefficients. $p(x)$ is a vector of basis functions that consists most often of monomials of the lowest orders to ensure minimum completeness. A complete polynomial basis of order m in two-dimensional space, is given by

$$p^{\mathrm{T}}(x) = \{1, x, y, xy, x^2, y^2, \ldots, x^m, y^m\} \tag{18}$$

DOI: 10.1002/fld

Given a set of n nodal values for the field function $u_1, u_2, \ldots, u_n$, at n nodes $x_1, x_2, \ldots, x_n$ that are in the *support domain*, Equation (17) is then used to calculate the approximated values of the field function at these nodes:

$$u^h(x, x_I) = p^{\mathrm{T}}(x_I)a(x), \quad I = 1, 2, \ldots, n \tag{19}$$

The weighted residual is defined as follows:

$$J = \sum_I^n \widehat{W}(x - x_I)[u^h(x, x_I) - u(x_I)]^2 = \sum_I^n \hat{W}(x - x_I)[p^{\mathrm{T}}(x_I)a(x) - u_I]^2 \tag{20}$$

where $W(x - x_I)$ is a weight function, and u_I is the nodal parameter of the field variable at node I. In MLS approximation, at an arbitrary point x, $a(x)$ is chosen to minimize the weighted residual. The minimization condition requires [1]

$$\frac{\partial J}{\partial a} = 0 \tag{21}$$

which results in the following linear equation system:

$$A(x)a(x) = B(x)U_{\mathrm{s}} \tag{22}$$

where $A(x)$ and $B(x)$ can easily be obtained from Equation (20). U_{s} is the vector that collects the nodal parameters of the field variables for all the nodes in the support domain:

$$U_{\mathrm{s}} = \{u_1, u_2, \ldots, u_n\}^{\mathrm{T}} \tag{23}$$

Solving Equation (22) for $a(x)$, we obtain

$$a(x) = A^{-1}(x)B(x)U_{\mathrm{s}} \tag{24}$$

Substituting the above equation back into Equation (17) leads to

$$u^h(x) = p^{\mathrm{T}}A^{-1}(x)B(x)U_{\mathrm{s}} \tag{25}$$

or

$$u^h(x) = N(x)U_{\mathrm{s}} \tag{26}$$

where $N(x)$ is the matrix of MLS shape functions corresponding to n nodes in the support domain. Using these shape functions, the field variables can be easily written as follows:

$$\begin{aligned} &U_i = N_u\widetilde{U}_i, \quad \Delta U_i = N_u\Delta\widetilde{U}_i, \quad \Delta U_i^* = N_u\Delta\widetilde{U}_i^* \\ &u_i = N_u\widetilde{u}_i, \quad p = N_p\widetilde{p} \quad \text{and} \quad \rho = N_\rho\widetilde{\rho} \end{aligned} \tag{27}$$

In the above equation

$$\widetilde{U}_i = [U_i^1, U_i^2, \ldots U_i^k \ldots U_i^m]^{\mathrm{T}}$$

and

$$N = [N^1, N^2, \ldots N^k \ldots N^m] \tag{28}$$

DOI: 10.1002/fld

where k is the node (or variable) identifying number (and varies between 1 and m). Before introducing the above relations, we have the following weak form of Equation (13) for the standard Galerkin approximation (weighting functions are the shape functions) [17]:

$$\int_\Omega N_u^k \Delta U_i^* \,\mathrm{d}\Omega = \Delta t \left[-\int_\Omega N_u^k \frac{\partial}{\partial x_j}(u_j U_i)\,\mathrm{d}\Omega - \int_\Omega \frac{\partial N_u^k}{\partial x_j}\tau_{ij}\,\mathrm{d}\Omega - \int_\Omega N_u^k(\rho g_i)\,\mathrm{d}\Omega \right]^n + \frac{\Delta t^2}{2}\left[\int_\Omega \frac{\partial}{\partial x_l}(u_l N_u^k)\left(-\frac{\partial}{\partial x_j}(u_j U_i) + \rho g_i \right)\mathrm{d}\Omega \right]^n + \Delta t \left[\int_\Gamma N_u^k \tau_{ij} n_j \,\mathrm{d}\Gamma \right]^n \tag{29}$$

It should be noted that in the above equations the weighting functions are the shape functions as the standard Galerkin approximation is used. Also here, the viscous and stabilizing terms are integrated by parts and the last term is the boundary integral arising from integrating by parts the viscous contribution.

The weak form of the density–pressure equation is

$$\int_\Omega N_p^k \Delta\rho = \int_\Omega N_p^k \frac{1}{c^2}\Delta p \,\mathrm{d}\Omega = \Delta t \int_\Omega \frac{\partial N_p^k}{\partial x_i}\left[U_i^n + \theta_1 \left(\Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) \right]\mathrm{d}\Omega - \Delta t \theta_1 \int_\Gamma N_p^k \left(U_i^n + \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) n_i \,\mathrm{d}\Gamma \tag{30}$$

In the above, the pressure and ΔU_i^* terms are integrated by parts. The weak form of the correction equation is

$$\int_\Omega N_u^k \Delta U_i^{n+1} \,\mathrm{d}\Omega = \int_\Omega N_u^k \Delta U_i^* \,\mathrm{d}\Omega - \Delta t \int_\Omega N_u^k \left(\frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \right)\mathrm{d}\Omega - \frac{\Delta t^2}{2}\int_\Omega \frac{\partial}{\partial x_j}(u_j N_u^k)\frac{\partial p^n}{\partial x_i}\,\mathrm{d}\Omega \tag{31}$$

The weak form of the energy equation can be written as follows

$$\int_\Omega N_E^k \Delta(\rho E)^{n+1}\,\mathrm{d}\Omega = \Delta t \left[-\int_\Omega N_E^k \frac{\partial}{\partial x_i}(u_i(\rho E + p))\,\mathrm{d}\Omega - \int_\Omega \frac{\partial N_E^k}{\partial x_i}\left(\tau_{ij}u_j + k\frac{\partial T}{\partial x_i} \right)\mathrm{d}\Omega \right]^n + \frac{\Delta t^2}{2}\left[\int_\Omega \frac{\partial}{\partial x_j}(u_j N_E^k)\left[\frac{\partial}{\partial x_i}(-u_i(\rho E + p)) \right]\mathrm{d}\Omega \right]^n + \Delta t \left[\int_\Gamma N_E^k \left(\tau_{ij}u_j + k\frac{\partial T}{\partial x_i} \right) n_i \,\mathrm{d}\Gamma \right]^n \tag{32}$$

DOI: 10.1002/fld

The last step is the imposition of essential boundary conditions which will be discussed in the next section.

2.5. Applying essential boundary conditions

Boundary conditions play a critical role in the solution of partial differential equations. In FEM, inputting these conditions is not very difficult. There are standard procedures or techniques for the implementation of the boundary conditions, in the form of either single point or multipoint constraints. All the techniques developed in FEM are applicable (with some modifications) to MFree methods.

MFree methods using PIM and RPIM shape functions possess the Kroenecker delta property. In these methods, the imposition of the essential boundary conditions is the same as that in FEM. In MFree methods using MLS approximations for constructing shape functions, special techniques are required to impose essential boundary conditions, because the shape functions created do not satisfy the Kroenecker delta conditions. There are some techniques for imposing essential boundary conditions in meshfree methods using MLS approximations for constructing shape functions such as Lagrange multipliers and penalty method. In this work, penalty method has been used for applying essential boundary conditions. This method has some advantages and some disadvantages [1].

- The dimension and positive definite property of the matrix are preserved, as long as the penalty factors chosen are positive.
- The symmetry and bandedness of the system matrix are preserved.
- The results obtained are in general less accurate, compared with the method of Lagrange multipliers.
- An essential boundary condition can never be precisely imposed. It is imposed only approximately.

Despite these minor disadvantages, the penalty method is much more favourable for many researchers. The penalty method has been frequently used in FEM for enforcement of single or multipoint constraints [27]. The essential boundary conditions needed to be enforced have the form

$$\sum_{i=1}^{n} N_i \phi_i = N(x)\Phi_s = \overline{\phi}(x,t) \quad \text{on } \Gamma_\phi \tag{33}$$

where $\overline{\phi}(x,t)$ is the prescribed field variable on the essential boundary. It can be varied with position vector and time. For example for pressure equation (Equation (30)) we have

$$p = \overline{p}(x,t) \quad \text{on } \Gamma_p \tag{34}$$

Using the Taylor series and eliminating higher-order terms ($\Delta t^2, \Delta t^3, \ldots$), we have:

$$\Delta p = \frac{\partial \overline{p}}{\partial t}\Delta t \tag{35}$$

Equation (35) can be written in following form:

$$\int_{\Gamma_p} N_p^k \Delta p \, \mathrm{d}\Gamma = \Delta t \int_{\Gamma_p} N_p^k \frac{\partial \overline{p}}{\partial t} \, \mathrm{d}\Gamma \tag{36}$$

DOI: 10.1002/fld

Multiplying Equation (36) by a suitable coefficient, α (penalty factor) and adding this equation to Equation (30), we have

$$\begin{aligned}
&\alpha \int_{\Gamma_p} N_p^k \Delta p \, \mathrm{d}\Gamma + \int_\Omega N_p^k \frac{1}{c^2} \Delta p \, \mathrm{d}\Omega \\
&= \Delta t \int_\Omega \frac{\partial N_p^k}{\partial x_i} \left[U_i^n + \theta_1 \left(\Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) \right] \mathrm{d}\Omega \\
&\quad - \Delta t \theta_1 \int_{\Gamma_n} N_p^k \left(U_i^n + \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) n_i \, \mathrm{d}\Gamma + \alpha \Delta t \int_{\Gamma_p} N_p^k \frac{\partial \overline{p}}{\partial t} \, \mathrm{d}\Gamma
\end{aligned} \tag{37}$$

Choosing suitable large value of α, Equation (37) satisfies the essential boundary condition which is represented in Equation (36). It should be noted that applying very large penalty coefficient causes ill conditioning.

Other boundary conditions can be imposed in the same manner. After spatial representation of the general equations of fluid dynamics, the relating partial differential equations can be easily changed to a system of linear algebraic equations in every time steps. Solving these algebraic equations, the field variables of the fluid flow can be obtained.

3. RESULTS AND DISCUSSION

In this work, a new finite element and MFree code is developed for solving flow problems. The finite element code is developed based on the general algorithm of CBS method which has been completely described in the literature [17]. The meshfree computational code is developed based on the meshfree CBS algorithm which has been discussed in the above. This computational code is capable of solving both time-dependent and steady-state flow problems. Numerical simulation of some known benchmark flow problems has been studied.

3.1. Case study #1

Lid-driven cavity flow in a square $[0, 1] \times [0, 1]$, as shown in Figure 2 has been studied. To have more accurate solution, numerical analysis has been done for non-uniform scattered nodes (M1: 40×40 nodes) as shown in Figure 3(a). As may be seen in Figure 3(a), the highest density of nodes is to be found near the lid and walls. This is done in order to resolve adequately the very thin boundary layers on the lid and cavity walls. Figure 3(b) shows three different grids with different irregularities of scattered nodes: (M2: 20×20 nodes), (M3: 20×20 nodes) and (M4: 20×20 nodes). Using these grids, the effect of irregularity of nodes on the solution has been discussed.

The first numerical results correspond to the solution of the Navier–Stokes equations on grids (M1). Figure 4(a) compares profiles of u along the line $x = 0.5$ computed with MFree method and FEM with the different Reynolds numbers. Figure 4(b) compares profiles of v along the line $y = 0.5$ computed with MFree method and FEM with the different Reynolds numbers. In this solution MLS shape functions are used for field variables interpolation. The Reynolds number for this flow is based on the lid velocity and cavity height. Good agreement in the both graphs can be seen. Figure 5 compares profiles of u along the line $x = 0.5$ computed with MFree method

DOI: 10.1002/fld

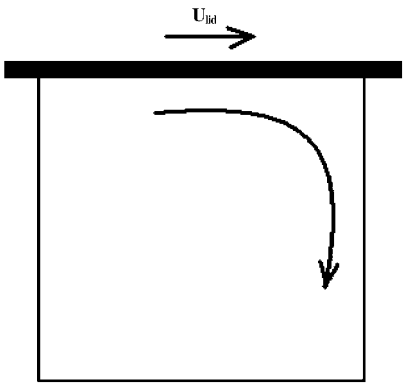

Figure 2. Lid-driven cavity flow in a square $[0, 1] \times [0, 1]$.

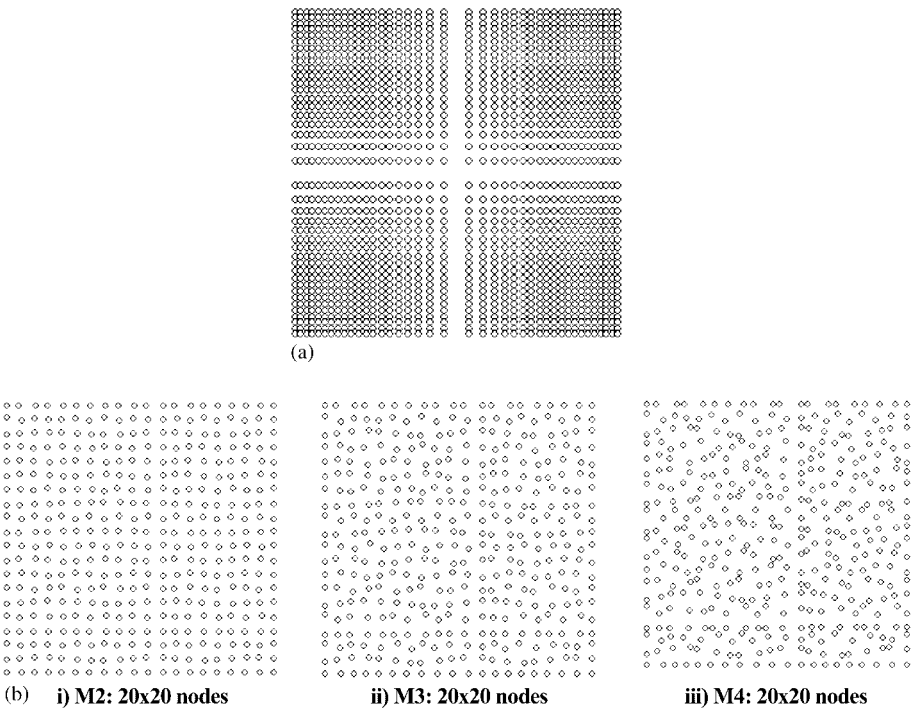

Figure 3. (a) M1: 40×40 nodes distributed non-uniformly in the problem domain. (b) Three different grids with different irregularities.

and FEM with nodes irregularly distributed in the problem domain. Some perturbation is observed in the solution which used very irregular nodes distributions. However, MFree method shows stable solution even with irregular nodes distributions. Figure 6 compares MFree results by using

DOI: 10.1002/fld

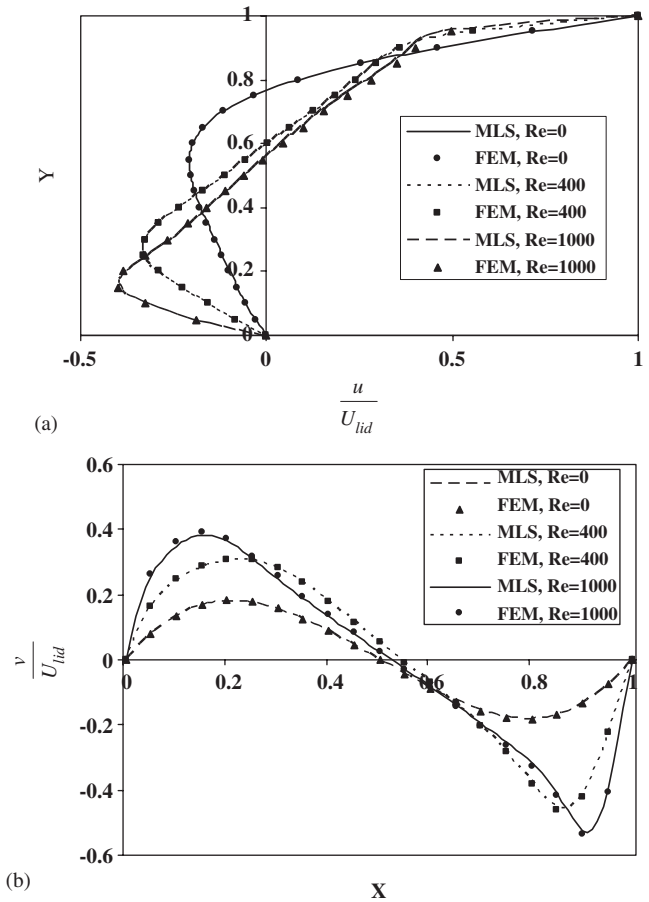

Figure 4. (a) Profiles of u along the line $x = 0.5$ computed with MFree method and FEM at different Reynolds numbers. (b) Profiles of v along the line $y = 0.5$ computed with MFree method and FEM at different Reynolds numbers.

different shape functions. Table I compares MFree results with some known data in the literature [28]. The table shows the results of MFree method by using different shape functions. According to the results obtained from 40×40 nodes, MFree method has good agreement with other results obtained from very finer grids [28]. It shows maximum error of 3% at $Re = 1000$. It can be observed that the Mfree method is more accurate than FEM. Figure 7 compares profiles of u along the line $x = 0.5$ computed with MFree method and FEM at high Reynolds number ($Re = 10\,000$). It can be observed that Meshfree CBS algorithm is completely stable even at high Reynolds numbers. Figure 8 shows the effect of the number of Gaussian integration points on the accuracy of the solution. The normalized error is evaluated from following formula:

$$\|e\| = \left| \frac{\mathrm{Max}(u - u_{\mathrm{accurate}})}{u_{\mathrm{accurate}}} \right| \tag{38}$$

DOI: 10.1002/fld

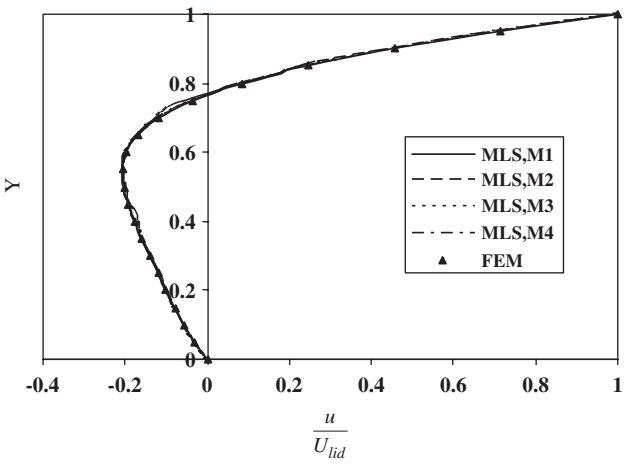

Figure 5. Profiles of u along the line $x = 0.5$ computed with MFree method and FEM using irregular node distributions ($Re = 0$).

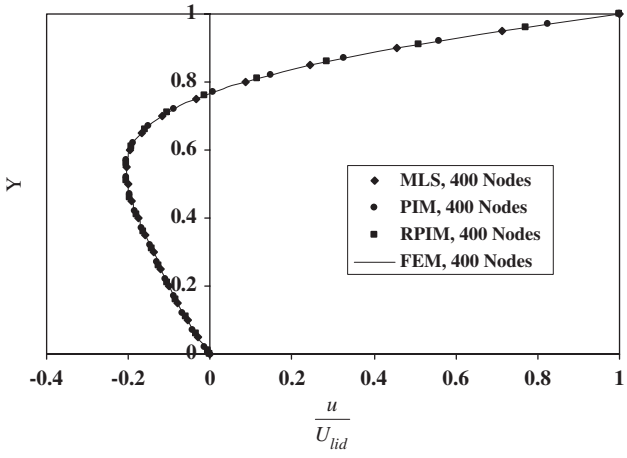

Figure 6. MFree results using different shape functions ($Re = 0$).

where u is the evaluated absolute velocity of arbitrary point in the problem domain. The accuracy of the solution increases with the increase of the Gaussian points. Using high number of Gaussian points increases the CPU time significantly.

3.2. Case study #2

Numerical simulation of flow past a backward facing step, as shown in Figure 9 has been studied. The geometric parameter of the problem is as follows:

$L_{\text{inlet}} = 5\,\text{cm}$,
$L_{\text{channel}} = 12\,\text{cm}$,

DOI: 10.1002/fld

Table I. Verification of MFree results with some data in the literature.

Reference	$u_{\min}$	$y_{\min}$	$v_{\max}$	$x_{\max}$	$v_{\min}$	$x_{\min}$
$Re=0$						
FEM	−0.20695	0.53	0.1831	0.21	−0.18387	0.79
Present, MLS	−0.20825	0.54	0.180826	0.20	−0.18163	0.80
Present, PIM	−0.21038	0.54	0.183491	0.21	−0.184167	0.79
Present, RPIM	−0.208634	0.54	0.182211	0.21	−0.183134	0.79
Ghia *et al.* [28]	—	—	—	—	—	—
$Re=400$						
FEM	−0.3334	0.28	0.31098	0.23	−0.4646	0.87
Present, MLS	−0.33038	0.28	0.30857	0.22	−0.4559	0.86
Ghia *et al.* [28]	−0.32726	0.2813	0.30203	0.2266	−0.44993	0.8594
$Re=1000$						
FEM	−0.4040	0.16	0.3939	0.16	−0.5504	0.91
Present, MLS	−0.3934	0.17	0.383125	0.16	−0.5301	0.91
Ghia *et al.* [28]	−0.38289	0.1719	0.37095	0.1563	−0.51550	0.9063

Note: $u_{\min}$: The minimum u velocity along the line $x=0.5$; $y_{\min}$: The y position of the minimum u velocity along the line $x=0.5$; $v_{\max}$: The maximum v velocity along the line $y=0.5$; $x_{\max}$: The x position of the maximum v velocity along the line $y=0.5$; $v_{\min}$: The minimum v velocity along the line $y=0.5$; $x_{\min}$: The x position of the minimum v velocity along the line $y=0.5$.

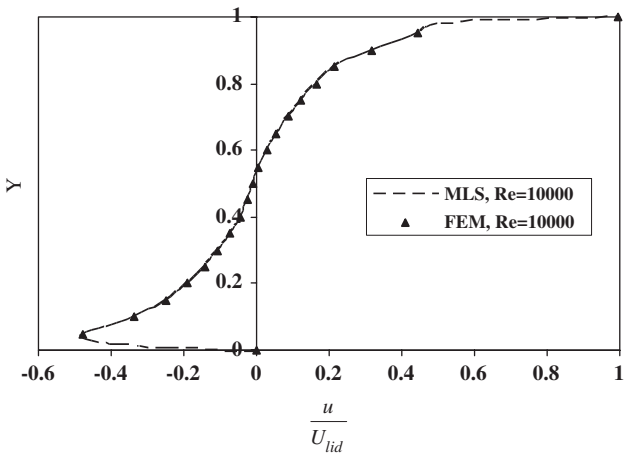

Figure 7. Profiles of u along the line $x=0.5$ computed with MFree method and FEM at high Reynolds number ($Re=10\,000$) (with 3600 nodes).

$h_{\text{inlet}}=0.5\,\text{cm}$,
$s=0.471\,\text{cm}$.

The Reynolds number is defined as follows:

$$Re=\tfrac{2}{3}\rho u_{\max}2h_{\text{inlet}}/\mu \tag{39}$$

DOI: 10.1002/fld

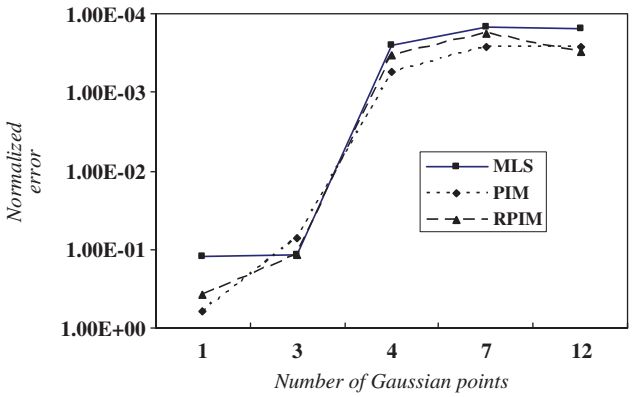

Figure 8. The effect of number of Gaussian points on the accuracy of the solution.

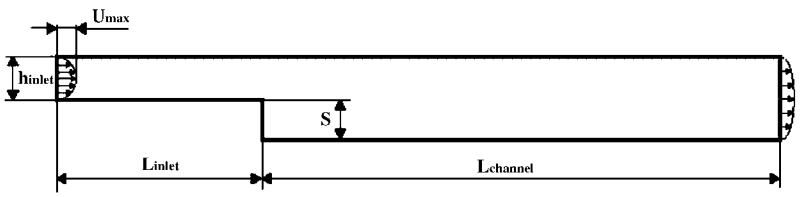

Figure 9. A schematic shows flow past backward facing step.

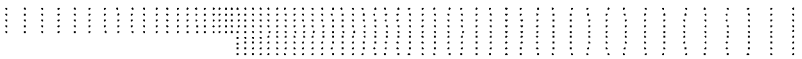

Figure 10. N1: 634 nodes.

Figure 11. N2: 1756 nodes.

In the present work, two different grids are employed: coarse (Ml: 634 nodes), and fine (M2: 1756 nodes) as shown in Figures 10 and 11, in order to investigate grid dependency of the solution. To have more accurate solution, numerical analysis has been done for non-uniform scattered nodes as shown in Figures 10 and 11.

The first numerical results correspond to the solution of the Navier–Stokes equations on two different grids (M1, M2) at different Reynolds numbers. Figures 12–14 show streamline in flow past a backward facing step at different Reynolds numbers. As the Reynolds number increases, the position of reattachment point increases too. In higher Reynolds numbers, for example, $Re = 800$ there is no stable location for reattachment point and it changes frequently with time. Figure 15

DOI: 10.1002/fld

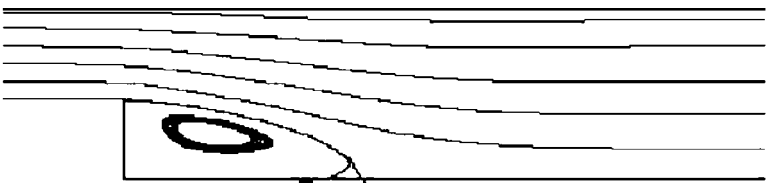

Figure 12. Streamlines in flow past a backward facing step, $Re = 100$.

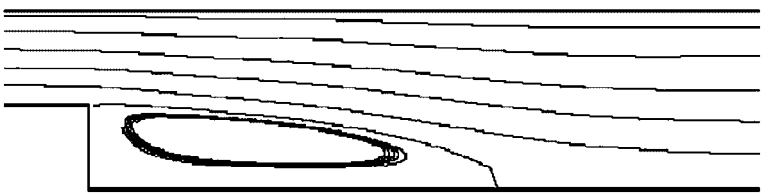

Figure 13. Streamlines in flow past a backward facing step, $Re = 200$.

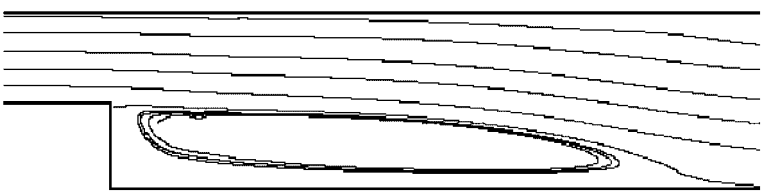

Figure 14. Streamlines in flow past a backward facing step, $Re = 400$.

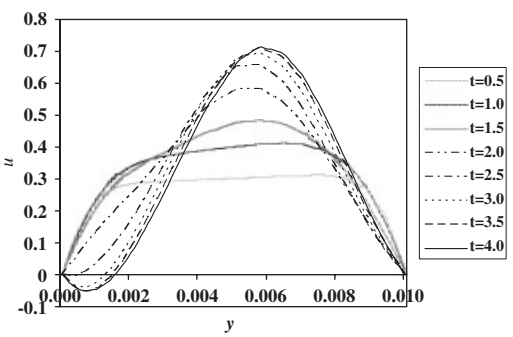

Figure 15. Unsteady profiles of u along the line $x = 6$ at different times.

shows unsteady profiles of u along the line $x = 6$ at different times. The inlet velocity has been calculated as shown in Figure 16, from the following equation:

$$\frac{U_{\text{inlet}}(t)}{U_{\max}(y)} = 1 - \exp(-3t) \tag{40}$$

DOI: 10.1002/fld

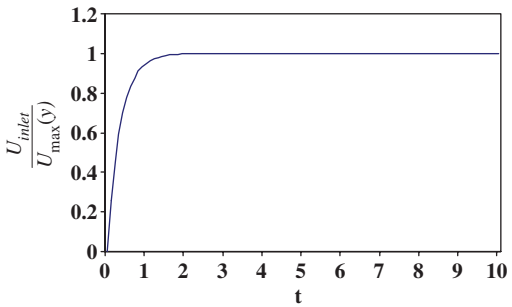

Figure 16. The inlet velocity.

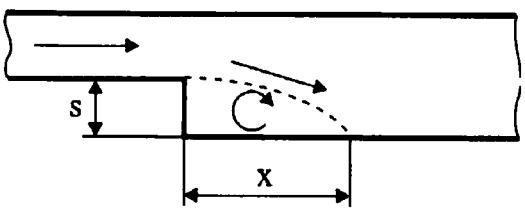

Figure 17. Position of the reattachment point.

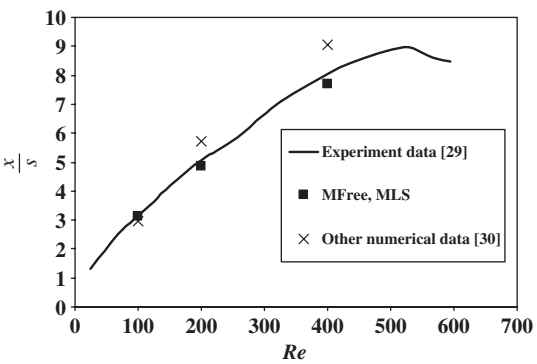

Figure 18. The comparison of the position of reattachment point predicted by meshfree method and experimental [24] data and other numerical data [29].

$U_{\max}(y)$ is a parabolic curve which is zero at walls:

$$\frac{U_{\max}(y)}{u_{\max}} = \frac{4}{h_{\text{inlet}}^2} y(h_{\text{inlet}} - y) \tag{41}$$

In Figure 17 schematic can be seen that shows the position of reattachment point. Figure 18 compares the values of position of reattachment point computed by MFree method and other experimental [30] and numerical data [29] in the literature. It is clear that MFree results are very

DOI: 10.1002/fld

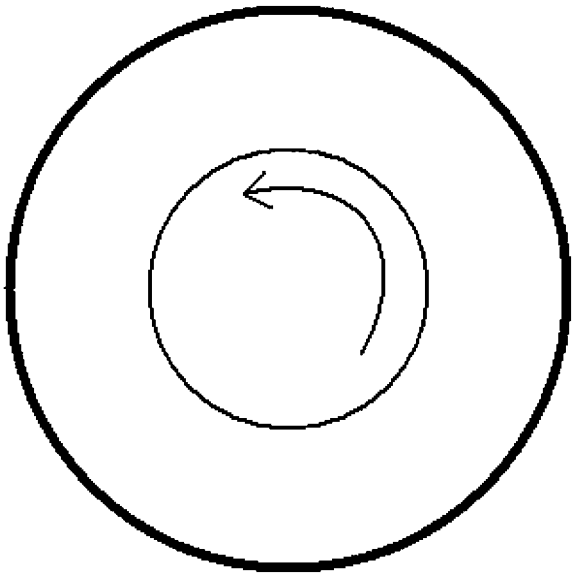

Figure 19. Geometry of centric cylinders.

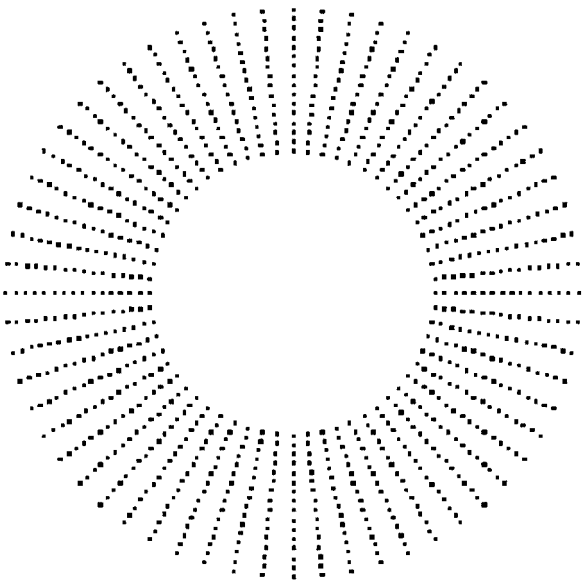

Figure 20. Nodes distribution in the problem domain.

close to experimental data. This meshfree algorithm (Meshfree CBS) shows more accurate results compared with other meshfree results reported in the literature [29].

DOI: 10.1002/fld

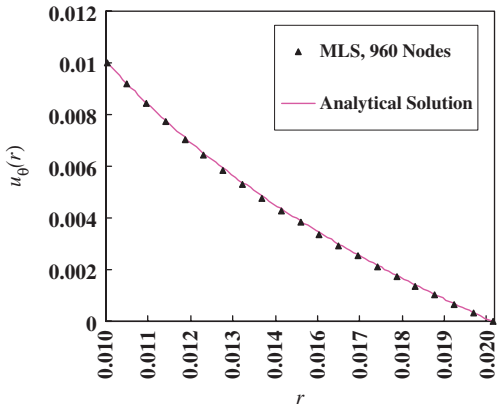

Figure 21. Radial component of velocity between centric rotating cylinders.

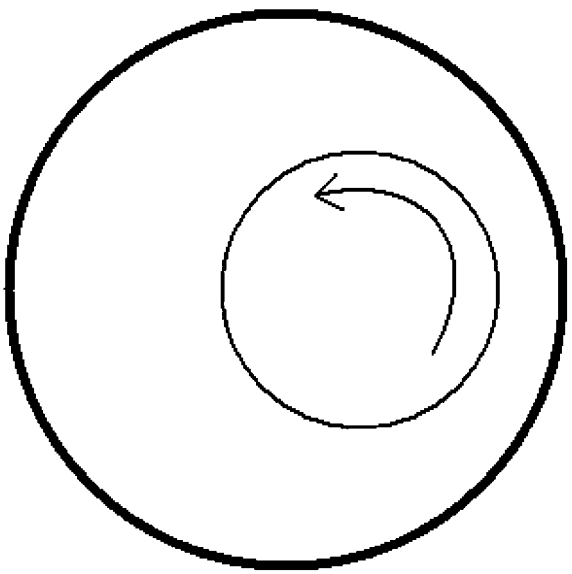

Figure 22. Geometry of eccentric cylinders.

3.3. Case study #3

In this case, numerical simulation of fluid flow between rotating cylinders is studied. Outer cylinder is fixed and inner cylinder is rotating. At first, as shown in Figure 19, it is assumed that cylinders are centric. Figure 20 shows nodes distribution used for numerical analysis of this flow. According to the analytical solution of this flow radial component of velocity is as follows [31]:

$$u_\theta(r) = A\frac{R}{2} + \frac{B}{r} \tag{42}$$

DOI: 10.1002/fld

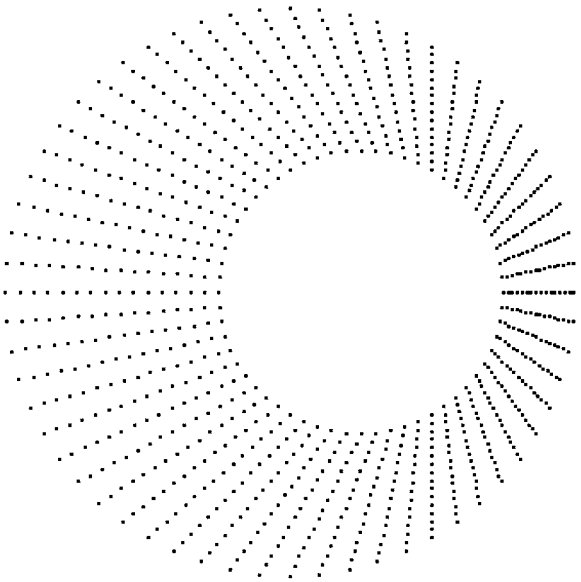

Figure 23. Nodes distribution in the problem domain.

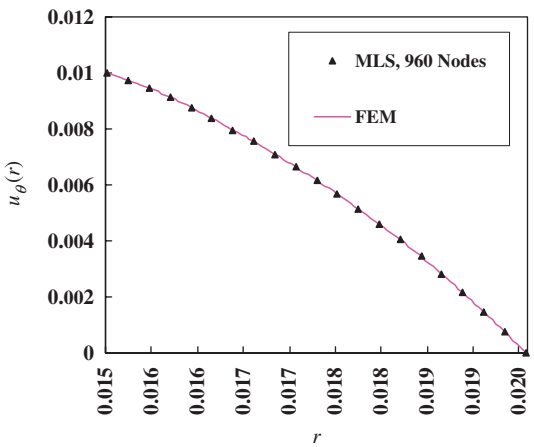

Figure 24. Radial component of velocity between eccentric rotating cylinders.

where A and B can easily be obtained by applying boundary conditions. Figure 21 compares numerical analysis obtained from meshfree CBS method with analytical solution using these data:

Inner diameter: 0.02.
Outer diameter: 0.04.

DOI: 10.1002/fld

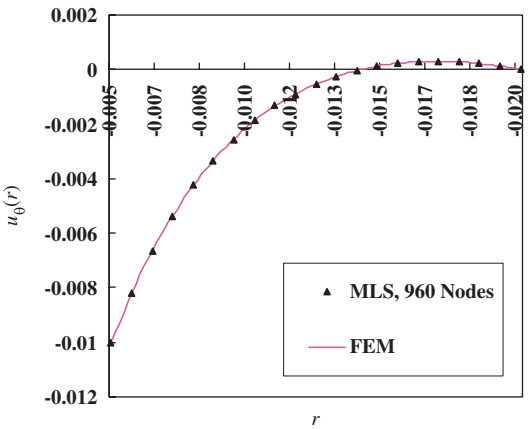

Figure 25. Radial component of velocity between eccentric rotating cylinders.

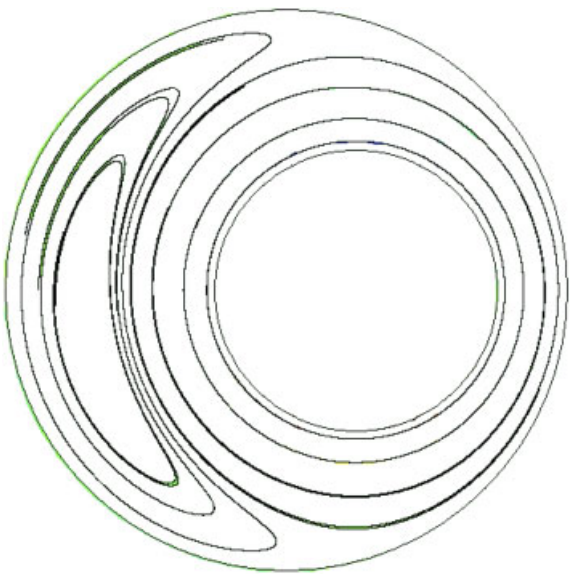

Figure 26. Streamlines between eccentric rotating cylinders.

Inner angular velocity: 1.0.
Outer angular velocity: 0.0.

It is clear that numerical analysis completely agree with analytical solution.
Numerical analysis of fluid flow between eccentric cylinders has also been studied. Figures 22 and 23 show the geometry and nodes distribution of the problem domain. The value of eccentricity is 0.005. Figure 24 compares radial component of velocity in the right side of rotating cylinder obtained from the meshfree method with those of the finite element. Figure 25 compares radial

DOI: 10.1002/fld

component of velocity in the left side of rotating cylinder obtained from the meshfree method with those of the finite element. It is clear that there is a good agreement between meshfree results and FEM results. Figure 26 shows streamlines between eccentric rotating cylinders. Eccentricity causes a crescent-shaped vortex to be developed.

4. CONCLUSION

The general equations of fluid dynamics have been discretized in the meshfree context. CBS algorithm has been used for this purpose. A new finite element and MFree code was developed for solving flow problems. This computational code is capable of solving both time-dependent and steady-state flow problems. Numerical simulation of some known benchmark flow problems has been studied. Computational results of MFree method have been compared to those of finite element method. The results obtained have been verified by known experimental, analytical and numerical data in the literature. According to the results obtained, it can be observed that the Mfree method is more accurate than FEM if the same numbers of nodes are used for each solver (Table I). It can be observed that Meshfree CBS algorithm is completely stable even at high Reynolds numbers (Figure 7). MFree results are very close to experimental data. Meshfree CBS algorithm shows more accurate results compared with other meshfree results reported in the literature [29] (Figure 18). A number of shape functions are used for field variable interpolation. The performance of each interpolation method is discussed. The effect of the number of Gaussian integration points on the accuracy of the solution has been discussed. The accuracy of the solution increases with the increase of the Gaussian points.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. M. Nik-khah Bahrami, Prof. M. H. Nai and Prof. V. Esfahanian in advance for their great help and comments in this work.

REFERENCES

1. Liu GR. *Mesh Free Methods* (1st edn). CRS Press LLC: Boca Raton, FL, 2002.
2. Lucy L. A numerical approach to testing the fission hypotheses. *Astronomical Journal* 1977; **82**:1013–1024.
3. Gingold RA, Monaghan JJ. Smooth particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **81**:375–389.
4. Chen W, Tanaka M. A meshless, exponential convergence, integration-free, and boundary-only RBF technique. *Computers and Mathematics with Applications* 2002; **43**:379–391.
5. Chen W. Symmetric boundary knot method. *Engineering Analysis with Boundary Elements* 2002; **26**(6):489–494.
6. Chen W. Meshfree boundary particle method applied to Helmholtz problems. *Engineering Analysis with Boundary Elements* 2002; **26**(7):577–581.
7. Shamekhi A, Nai MH. Buckling analysis of circular FGM plate having variable thickness under uniform compression by mesh free method. *International Journal of Computational Methods* 2005; **2**(3):327–340.
8. Nik-Khah-Bahrami M, Shamekhi A. Using mesh free method for free vibration analysis of circular fgm plates. *ICCES MM 2006 Conference*, Dubrovnik, Croatia.
9. Atluri SN, Tulong Z. New concepts in meshless method. *International Journal for Numerical Methods in Engineering* 2000; **47**:537–556.
10. Liu WK, Jun S, Sihling DT, Chen YJ, Hao W. Multiresolution reproducing kernel particle method for computational fluid dynamics. *International Journal for Numerical Methods in Fluids* 1997; **24**(12):1391–1415.

DOI: 10.1002/fld

11. Sadat H, Couturier S. Performance, accuracy of a meshless method for laminar natural convection. *Numerical Heat Transfer*, *Part B* 2000; **37**:455–467.
12. Yagawa G, Shirazaki M. Parallel computing for incompressible flow using a nodal-based method. *Computational Mechanics* 1999; **23**:209–217.
13. Cheng M, Liu GR. A novel finite point method for flow simulation. *International Journal for Numerical Methods in Fluids* 2002; **39**:1161–1178.
14. Kim DW, Kim YS. Point collocation methods using the fast moving least square reproducing kernel approximation. *International Journal for Numerical Methods in Engineering* 2003; **56**(10):1445–1464.
15. Chen W. *New RBF Collocation Schemes and Kernel RBFs with Applications*. Lecture Notes in Computational Science and Engineering, vol. 26. Springer: Berlin, 2002; 75–86.
16. Zienkiewicz OC, Codina R. A general algorithm for compressible and incompressible flow. Part I. The split characteristic based scheme. *International Journal for Numerical Methods in Fluids* 1995; **20**:869–885.
17. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (5th edn), vol. 3. Butterworth-Heinemann: London, 2000.
18. Chorin AJ. Numerical solution of Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
19. Comini G, Del Guidice S. Finite element solution of incompressible Navier–Stokes equations. *Numerical Heat Transfer*, *Part A* 1972; **5**:463–478.
20. Kawahara M, Ohmiya K. Finite element analysis of density flow using the velocity correction method. *International Journal for Numerical Methods in Fluids* 1985; **5**:981–993.
21. Ramaswamy B, Jue TC, Akin JE. Semi-implicit and explicit finite element schemes for coupled fluid thermal problems. *International Journal for Numerical Methods in Engineering* 1992; **34**:675–696.
22. Liu WK, Jun S, Zhang Y. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
23. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 1981; **37**:141–158.
24. Liu GR, Gu YT. A point interpolation method. *Proceedings of the 4th Asia-Pacific Conference on Computational Mechanics*, Singapore, December 1999; 1009–1014.
25. Wang JG, Liu GR. Radial point interpolation method for elastoplastic problems. *Proceedings of the 1st International Conference on Structural Stability and Dynamics*, Taipei, Taiwan, 7–9 December 2000; 703–708.
26. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
27. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (4th edn), vol. 1. McGraw-Hill: New York, 1989.
28. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**(3):387–411.
29. Tsukanov I, Shapiro V, Zhang S. A meshfree method for incompressible fluid dynamics problems. *International Journal for Numerical Methods in Engineering* 2003; **58**:127–158.
30. Armaly BF, Durst F, Pereira JCF, Schonung B. Experimental and theoretical investigation of backward facing step flow. *Journal of Fluid Mechanics* 1983; **127**:473–496.
31. Currie IG. *Fundamental Mechanics of Fluids* (2nd edn). McGraw-Hill: New York, 1974.

DOI: 10.1002/fld