U-TURN CHANNEL PRESSURE LOSS MINIMIZATION USING MESHLESS METHOD AND ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM

Abazar Shamekhi
Department of Mechanical Engineering, University of Alberta,
Edmonton AB T6G 2G8 Canada.
Email: shamekhi@ualberta.ca

ABSTRACT
In this work, a meshless method together with a differential evolution algorithm has been used for the numerical simulation and optimization of the flow inside a U-turn passage. Meshless characteristic based split (CBS) algorithm has been used for this purpose. Employing C++ language, a new computational code has been implemented. Using differential evolution algorithm together with a meshless computational simulation code, U-turn shape has been modified iteratively. The results show that the pressure loss of the U-turn passage of the optimal design has been decreased significantly compared with the initial shape of the channel.

Keywords: *Meshless Method, fluid flow, Differential Evolution, Optimization.*

1. INTRODUCTION

Performance improvement and cost lowering of new products are the major goals of manufacturers in today's competitive market. Computational simulation and optimization techniques are proved to be very helpful for engineers to designer better and cheaper products. The most important challenge in the optimization of mechanical systems is CPU time. Conventional mesh based methods like finite element method (FEM) or finite volume method (FVM) are very powerful tools for numerical simulation of flow, which is essential for shape optimization of mechanical systems. The major drawback of these methods is the necessity of remeshing of the problem domain in each iteration which is very time consuming. To overcome these difficulties, meshless (or meshfree) methods have been developed in recent decades. Meshless methods use only arbitrary scattered points instead of mesh/grid for discretization of the problem domain. Elimination of meshing and remeshing process which is essential in mesh based methods can increase the speed of meshless based optimization tools dramatically. Furthermore, because meshless methods use arbitrary irregular points for domain discretization, there is no limitation for numerical simulation of complex geometries having complicated boundaries [1].

Several meshless methods have been developed by now such as element free Galerkin method (EFG), finite point method (FPM), meshless local Potrove Galerkin method (MLPG) and smooth particle hydrodynamics or SPH method [1]. In general, there are two kinds of meshless methods, weak form meshless method such as EFG method and strong form meshless method such as SPH method [1]. SPH method which is one of the pioneering meshless methods, was initially introduced by Lucy, Gingold and Monoghan to solve three dimensional astrophysical problems [2, 3]. Meshless method has been employed by many authors for solving engineering problems [4-6]. Among the wide range of engineering problems, numerical simulations of flow problems are very challenging due to the nonlinear behavior of their governing equations. Recently some researchers have employed meshless methods for numerical simulation of flow problems. Liu et al. employed the reproducing kernel particle method (RKPM) with SUPG formulation to solve 2D advection-diffusion equation [7]. Yagawa and Shirazaki applied free mesh method (FMM) with the weighed residual-Galerkin method to unsteady two-dimensional incompressible viscous flow [8]. Kim and Kim [14] presented some analyses of fluids by meshless point collocation method (MPCM). Shamekhi et al. extended characteristic based split (CBS) method in meshless framework and employed this technique for solving several flow problems [9-11].

Shape optimization has attracted the attention of many academic and industrial researchers. Several kinds of optimization algorithms have been developed by now. Among them Evolutionary optimization algorithms have some advantages that allows optimization of fitness function of discrete variables. Evolutionary algorithms employ iterative progress, similar to growth or development and evolution, which happen in the creatures. The origin of evolutionary computing techniques initiated in 1950s, with the idea of using Darwinian principles for automated problem solving. John Henry Holland from University of Michigan at Ann Arbor, introduced genetic algorithm for solving practical optimization problems [29]. After that, many researchers employed genetic algorithm for optimization of engineering problems [30-32]. In 1995 R. Storn and K. V. Price invented another kind of evolutionary algorithm called Differential Evolution (DE) [33]. In 1996 they introduced their new optimization

algorithm at the First International Contest on Evolutionary Optimization [17]. Since 1996 differential evolution has been used in engineering design for shape optimization. Madavan used differential evolution for shape optimization of a turbine airfoil. The author used parallelization techniques on distributed computers to reduce CPU time [34]. Song et al. employed adaptive range differential evolution for shape optimization of a turbine stage. . They investigated the robustness and reliability of their adaptive method for the turbomachinery optimization design [35]. Chi et al. presented a self-adaptive center-mutation differential evolution for shape optimization of a turbine blade. Using this optimization technique, they could increase the turbine efficiency performances by 5.25 percent [36]. In most of the studies, which have been presented in the literature regarding shape optimization, researchers have employed conventional flow solvers such as FDM, FVM or FEM for solving the governing partial differential equations of fluid dynamics.

In this work, meshless method has been used for numerical simulation of flow inside a U-turn passage. Meshless characteristic based split (CBS) algorithm [11] has been employed for this purpose. Moving least square (MLS) technique has been used for construction of shape functions. The MLS approximation has two major advantages: 1) continuousness and smoothness of the approximated field function in the whole problem domain; and 2) capability of MLS method to produce an approximation with the desired order of consistency [1]. A new computational code has been implemented using C++ language for this purpose. Differential evolution (DE) algorithm has been used for shape optimization of U-turn geometry. Using differential evolution algorithm together with meshless computational simulation code, U-turn shape has been modified iteratively.

NOMENCLATURE

ρ : Fluid density
t: Time
u_i: Velocity component
p: Pressure
c: speed of sound
x_i: component of position
U_i: Mass flow rate
τ_{ij} : Deviatoric stress components
g_i: Body forces
η : Absolute viscosity
δ_{ij} : Kroneker delta
Δt : Time increment
U^*: Auxiliary velocity
u^h: PIM approximation of field variables
P: Vector of basis functions
n: Number of nodes or number of iterations
N(x): Shape function
N_u: Shape function for velocity
N_p: Shape function for pressure
$\tilde{U}$: Vector containing the values of mass flow rate at all nodes
$\tilde{P}$: Vector containing the values of pressure at all nodes
$\tilde{\rho}$: Vector containing the values of density at all nodes
N: individuals
F: scaling factor
C: real constant

2. NUMERICAL SIMULATION USING MESHLESS METHOD

In this section, numerical simulation of fluid flow using Characteristic Based Split Meshless method will be discussed in brief. This method is the extension of Characteristic Based Split algorithm in meshless framework. In 1968 Chorin introduced the split process for incompressible flow problems in the finite difference framework [21]. Some authors used and extended split method for different applications of incompressible flows [22-24]. In 1995 Zienkiewicz and Codina introduced general Characteristic Based Split (CBS) algorithm for solving both compressible and incompressible flows in finite element context [25]. This algorithm has been used and tested by

authors in different kinds of fluid dynamics problems [12]. Shamekhi *et. al.* extended CBS algorithm in meshless framework for simulation of both Newtonian and non-Newtonian flows [9-11].

2.1. GENERAL EQUATIONS

The major difference between solids and fluids is that fluids cannot support any shear stresses when at rest. To derive the governing equations of fluid mechanics, a fixed control volume is usually used. After that, the mass, momentum and energy fluxes of the flow are considered to derive the general equation of fluid flow, which is known as Navier-stokes equations. These equations, which are derived through Eulerian viewpoint, are coupled and nonlinear. Convective terms appeared in governing equation of fluid flow make the fluid dynamics equations non-self-adjoint. The Galerkin method which gives the minimum error in energy norm in self-adjoint problems such as solid mechanics equations, is no longer optimal approximating method to be used for solving governing equations of fluid mechanics [12]. In this section, the general equations of fluid mechanics which are used in CBS algorithm will be discussed briefly.

If we define the mass flow fluxes as $U_i = \rho u_i$, the equations of fluid mechanics, with some modifications, can be represented in the following forms [12]. The first governing equation of fluid flow is the conservation of mass or continuity equation. This equation in general form can be written as follows [12]:

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

In the above equation, *c* is the speed of sound. Assuming constant entropy, the relation between *c*, *p* and ρ can be written as follows [12]:

$$c^2 = \frac{\partial p}{\partial \rho} = \gamma \frac{p}{\rho}, \tag{2}$$

where $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heats. A different relation is used for a fluid exhibiting a small compressibility.

$$c^2 = \frac{K}{\rho}, \tag{3}$$

where K is the elastic bulk modulus. Finally, for fully incompressible fluids, the left hand of the equation 1 will be vanished.

The next equation is the equation of conservation of momentum. Conservation of momentum is the representation of the Newton's second law of classical mechanics in control volume. This equation can be written in each coordinate direction as follows:

$$\frac{\partial U_i}{\partial t} = -\frac{\partial}{\partial x_j}\left(u_j U_i\right) + \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i} - \rho g_i. \tag{4}$$

In all of the above equations u_i is the velocity component, ρ is the density, *p* is the pressure, ρg_i represents body forces and other source terms and τ_{ij} are the deviatoric stress components. For a Newtonian fluid, the relation between deviatoric stress and shear rate is linear and can be written as follows [12]:

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

where μ is absolute viscosity and δ_{ij} is the Kroneker delta ($\delta_{ij} = 1$, if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$).

The next equation is the equation of conservation of energy, which has been derived based on the first law of thermodynamics. The differential form of the equation of conservation of energy in control volume can be represented as follows [12]:

$$\frac{\partial(\rho E)}{\partial t} = -\frac{\partial}{\partial x_j}\left(u_j \rho E\right) + \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) - \frac{\partial}{\partial x_j}\left(u_j p\right) + \frac{\partial}{\partial x_i}\left(\tau_{ij} u_j\right). \tag{6}$$

In the above equation *E* is the specific energy, *T* is the absolute temperature and *k* is the thermal conductivity. Finally, when the flow is compressible, the above equation can be completed by the universal gas law as follows:

$$p = \rho RT, \tag{7}$$

where R is the universal gas constant. For incompressible flows, the above equation will be vanished from the system of equations of the fluid flow.

2.2. TEMPORAL DISCRETIZATION USING CBS ALGORITHM

To solve partial differential equations, the problem domain is usually discretized. Discretization is the process of transferring continuous models and equations into discrete counterparts. The governing partial differential equations of fluid dynamics are functions of time and space variables. In order to fully discretize partial differential equations of fluid dynamics, it is easier to separate this process into temporal discretization and spatial discretization. To temporally discretize the governing equations of fluid dynamics, characteristic based split algorithm is used. Using the characteristic Galerkin process [12] the momentum conservation equations can be written as follows [12]

$$\frac{\partial U_i}{\partial t} = -\frac{\partial}{\partial x_j}\left(u_j U_i\right) + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + Q_i^{n+\theta_2}, \tag{8}$$

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 [12]

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

with

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = \theta_2 \frac{\partial p^{n+1}}{\partial x_i} + \left(1-\theta_2\right)\frac{\partial p^n}{\partial x_i}, \tag{10}$$

$$\frac{\partial p^{n+\theta_2}}{\partial x_i} = \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i}. \tag{11}$$

where

$$\Delta p = p^{n+1} - p^n. \tag{12}$$

Using the characteristic-Galerkin process, the Eq. 8 can be rewritten as follows [12]:

$$U^{n+1} - U^n = \Delta t\left[-\frac{\partial}{\partial x_j}\left(u_j U_i\right)^n + \frac{\partial \tau_{ij}^n}{\partial x_j} + Q_i^{n+\theta_2} - \left(\rho g_i\right)^n + \left(\frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}\left(u_j U_i\right) - Q_i + \rho g_i\right)\right)^n\right] \tag{13}$$

At this step 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 exist as Split A and Split B respectively. In this paper Split A has been used. In this method, an auxiliary variable U_i^* is defined as follows [12]:

$$\Delta U_i^* = U_i^* - U_i^n = \Delta t\left[-\frac{\partial}{\partial x_j}\left(u_j U_i\right)^n + \frac{\partial \tau_{ij}^n}{\partial x_j} - \left(\rho g_i\right)^n + \left(\frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial}{\partial x_j}\left(u_j U_i\right)_i + \rho g_i\right)\right)^n\right] \tag{14}$$

The above equation can be solved subsequently using an explicit time step applied to the discretized form. Once the pressure increment is calculated, the following 'correction' term can be applied [12],

$$\Delta U_i = U_i^{n+1} - U_i^n = \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} - \frac{\Delta t^2}{2} u_k \frac{\partial Q_i^n}{\partial x_k}. \tag{15}$$

From Eq. 1 we have

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

Substituting U^{n+1} by the known intermediate, auxiliary variable *U** and rearranging after neglecting higher-order terms ($\Delta t^3, \Delta t^4, \ldots$) the above equation can be rewritten as follows [12]:

$$\Delta\rho = \left(\frac{1}{c^2}\right)^n \Delta p = -\Delta t\left[\frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t\theta_1\left(\frac{\partial^2 p^n}{\partial x_i \partial x_i} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i \partial x_i}\right)\right]. \quad (17)$$

It is clear that the above equation is fully self-adjoint. Therefore, the standard Galerkin-type procedure can be optimally used for spatial descretization. After evaluation of ΔU_i and Δp (or $\Delta\rho$), the energy equation can be solved independently and the value of $(\rho E)^{n+1}$ can be obtained by the characteristic-Galerkin process applied to Eq. (6). This work employs full form of the characteristic based split algorithm which is discussed above, for numerical simulation of incompressible flow. Therefore, the conservation of energy equation and universal gas law will not be solved.

Choosing different values of θ_1, θ_2, the above equations can be solved in the form of explicit, semi-implicit and nearly implicit [12]. Table 1 shows the values of θ_1, θ_2 in different forms of solution.

Table 1. Values of θ_1, θ_2 in different forms of solution

Fully explicit	$\frac{1}{2} \le \theta_1 \le 1$ and $\theta_2 = 0.$
Semi-implicit	$\frac{1}{2} \le \theta_1 \le 1$ *and* $\frac{1}{2} \le \theta_2 \le 1.$
Nearly implicit	$\theta_1 = \theta_2 = 1.$

After temporal discretization of the governing partial differential equations of fluid dynamics, these equations should be discretize spatially in order to transform partial differential equations to a system of linear algebraic equations.

2.3. SPATIAL DISCRETIZATION USING MESHLESS METHOD

Characteristic based split algorithm was used in previous section for temporal discretization of the governing partial differential equations of fluid dynamics. The next step is spatial discretization of these equations. Spatial derivatives of the partial differential equations can be approximated at any point by using a simple linear function of the nearby unknown discretized field variables. This process can transform partial differential equations to a system of linear algebraic equations. There are many techniques for spatial discretization of partial differential equations such as finite difference method, finite volume method and finite element method. Zienkiewicz and Codina used finite element method for spatial discretization in CBS algorithm [25]. In this work, meshless method is used for spatial discretization of the governing equations of fluid dynamics. Similar to finite element method, meshless method employs shape functions to approximate field variables in the problem domain. The major difference between finite element method and meshless method is that meshless method does not use elements for creating shape functions. Meshless method uses only nodes scattered arbitrarily in a domain without using any predefined mesh to provide connectivity of the nodes. Construction of Meshless shape functions is the most important topic in Meshless methods. To minimize CPU time, 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 in two dimensional problems and sphere or cube in three dimensional problems. The size of support domains can be constant or can be adaptively modified with respect to the density of the nodes in different parts of the problem domain. Figure 1 shows three support domains that their sizes have been modified adaptively.

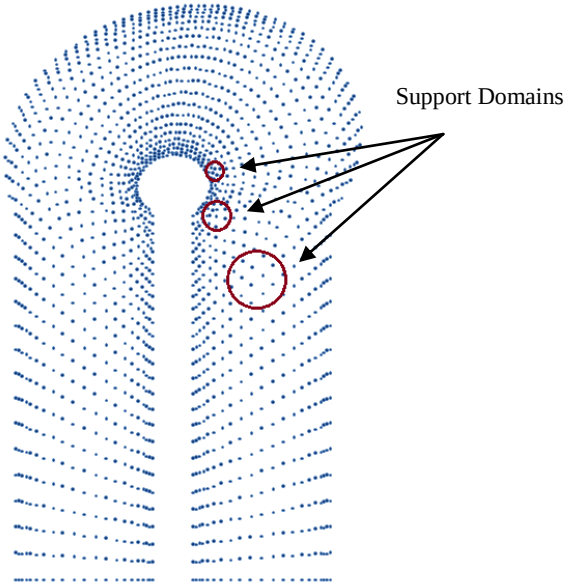

Figure 1. Adaptive support domain

There are several techniques to construct shape functions such as SPH method developed by Lucy, Gingold and Monaghan [2,3] and RKPM by Liu *et al.* [26]. These methods are classified in finite integral representation category. In finite series representation form, there are some procedures such as MLS method, which was developed by Lancaster and Salkauskas [14], PIM by Liu and Gu [15] and RPIM by Wang and Liu [16]. In this paper, we describe MLS procedure in brief. For more information about shape function constructions in meshless method reader is referred to the literature [1].

2.3.1. MOVING LEAST SQUARES APPROXIMATION

Moving least squares (MLS) was initially introduced for data fitting and surface construction in 1981 by Lancaster [14]. MLS was used for shape function construction in diffuse element method by Nayroles et al. in 1992 [27]. The MLS approximation has two major advantages: 1) continuousness and smoothness of the approximated field function in the whole problem domain; and 2) capability of MLS method to produce an approximation with the desired order of consistency [1].

Consider *u(x)* to be the function of the field variable defined in the problem domain. The approximation of *u(x)* at point *x* is denoted by u^h(x). MLS approximation first writes the field function in the form [1]:

$$u^h(X) = \sum_j^m p_j(X)a_j(X) \equiv p^T(X)a(X), \tag{18}$$

where $u^h(X)$ is the approximation of *u(X)*, *X* is the 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 2D space, is given by [1]

$$p^T(X) = \{1, x, y, xy, x^2, y^2, ..., x^m, y^m\}. \tag{19}$$

For a given set of *n* nodal values of the field function, u_1 u_2,...,u_n, at *n* nodes x_1 x_2,...,x_n that are located in the support domain, Eq. 18 can be employed to calculate the approximated values of the field function at these nodes [1]

$$u^h(X, X_I) = p^T(X_I)a(X), \qquad I = 1,2,...,n. \tag{20}$$

The weighted residual is defined as follows [1]

$$J = \sum_I^n \hat{W}(X - X_I)\left[u^h(X, X_I) - u(X_I)\right]^2 = \sum_I^n \hat{W}(X - X_I)\left[p^T(X_I)a(X) - u_I\right]^2, \tag{21}$$

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]

$$\frac{\partial J}{\partial a} = 0, \tag{22}$$

which results in the following linear equation system

$$A(X)a(X) = B(X)U_s, \tag{23}$$

where *A(X)* and *B(X)* can easily be obtained from Eq. 21 and Eq. 22. U_s is the vector that collects the nodal parameters of the field variables for all of the nodes in the support domain

$$U_s = \{u_1, u_2, ..., u_n\}^T. \tag{24}$$

Solving Eq. 23 for *a(X)*, we obtain

$$a(X) = A^{-1}(X)B(X)U_s. \tag{25}$$

If the above equation is substituted into the Eq. 18, the approximation of the field function can be determined as follows [1]:

$$u^h(X) = p^T A^{-1}(X)B(X)U_s \tag{26}$$

or

$$\mathrm{u^h(X) = N(X)U_s}, \tag{27}$$

where $N(X) = p^T A^{-1}(X)B(X)$ is the matrix of MLS shape functions corresponding to *n* nodes that are located in the support domain. Using these shape functions, the field variables can be easily written as follows:

$$U_i = N_u \hat{U}_i, \quad \Delta U_i = N_u \Delta \hat{U}_i, \quad \Delta U_i^* = N_u \Delta \hat{U}_i^*,$$
$$u_i = N_u \hat{u}_i, \quad p = N_p \hat{p} \quad \text{and} \quad \rho = N_\rho \hat{\rho}. \tag{28}$$

where,

$$\hat{U}_i = \left[U_i^1, U_i^2, ...U_i^k ...U_i^m\right]^T$$
$$\text{and} \quad N = \left[N^1, N^2, ...N^k ...N^m\right], \tag{29}$$

In the above equations, *k* is the node (or variable) identifying number which, varies between *1* and *m*. Using above meshless shape functions, partial derivetives of the governing equation of fluid dynamics can be represented with simple algebraic equations. Using Galerkin procedure, these shape functions can be used in Equations 13-16 to transform temporally discretized partial differential equations to a system of linear algebraic equations. Employing standard Galerkin approximation, the weak form of Eq. 14 can be written as follows [12]:

$$\int_\Omega N_u^k \Delta U_i^* d\Omega = \Delta t\left[-\int_\Omega N_u^k \frac{\partial}{\partial x_j}\left(u_j U_i\right)d\Omega - \int_\Omega \frac{\partial N_u^k}{\partial x_j}\tau_{ij} d_\Omega - \int_\Omega N_u^k \left(\rho g_i\right)d\Omega \right]^n$$
$$+ \frac{\Delta t^2}{2}\left[\int_\Omega \frac{\partial}{\partial x_l}\left(u_l N_u^k\right)\left(-\frac{\partial}{\partial x_j}\left(u_j U_i\right) + \rho g_i\right)d\Omega\right]^n + \Delta t\left[\int_\Gamma N_u^k \tau_{ij} n_j d\Gamma\right]^n. \tag{30}$$

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

Using integration by parts, the weak formulation of the density-pressure equation (Eq. 17) is as follows [12]:

$$\int_\Omega N_p^k \Delta\rho = \int_\Omega N_p^k \frac{1}{c^2}\Delta p d\Omega = \Delta t \int_\Omega \frac{\partial N_p^k}{\partial x_i}\left[U_i^n + \theta_1\left(\Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i}\right)\right]d\Omega$$
$$- \Delta t\theta_1 \int_\Gamma N_p^k \left(U_i^n + \Delta U_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i}\right)n_i d\Gamma. \tag{31}$$

Similarly, the weak form of the correction equation (Eq. 15) can be written as follows [12]:

$$\int_\Omega N_u^k \Delta U_i^{n+1} d\Omega = \int_\Omega N_u^k \Delta U_i^* d\Omega - \Delta t \int_\Omega N_u^k \left(\frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i}\right)d\Omega - \frac{\Delta t^2}{2}\int_\Omega \frac{\partial}{\partial x_j}\left(u_j N_u^k\right)\frac{\partial p^n}{\partial x_i}d\Omega. \tag{32}$$

Finally, the weak form of the energy equation can be written as follows [12]:

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

$$+\frac{\Delta t^2}{2}\left[\int_{\Omega}\frac{\partial}{\partial x_j}\left(u_j N_E^k\right)\left[\frac{\partial}{\partial x_i}\left(-u_i\left(\rho E+p\right)\right)\right]d\Omega\right]^n \tag{33}$$

$$+\Delta t\left[\int_{\Gamma} N_E^k\left(\tau_{ij}u_j+k\frac{\partial T}{\partial x_i}\right)n_i d\Gamma\right]^n .$$

Using above standard Galerkin procedure, the partial differential equations can be transformed into a system of linear algebraic equations where, the field variable in all nodes are unknowns in this system of linear equations. This system of linear equation can be easily solved using common techniques.

3. OPTIMIZATION METHOD

Differential evolution (DE) algorithm has been used in this work for shape optimization of U-turn channel. Differential evolution was developed by Price and Storn [17] in 1996. Like genetic algorithm (GA), this method is able to optimize objective functions which are function of discrete variables. The most advantage of DE algorithm compared to GA is that this method does not require the transformation of the variables into binary strings.

3.1 SINGLE-OBJECTIVE ADAPTIVE DIFFERENTIAL EVOLUTION

In this work single-objective adaptive DE has been used for shape optimization of U-turn passage. Assume *t*-th generation of designs contains *N* individuals. Each individual $\vec{x}_t$ has *n* design parameters as follows [19]:

$$\vec{x}_t=\left(x_1,x_2,x_3,...,x_n\right) \tag{34}$$

To evolve the parameter vector $\vec{x}_t$, three other parameter vectors $\vec{a}_t$, $\vec{b}_t$ and $\vec{c}_t$ are randomly selected such that $\vec{a}_t\neq\vec{b}_t\neq\vec{c}_t\neq\vec{x}_t$. Then the vector $\vec{y}_t$ is defined as follows [19]:

$$\vec{y}_t=\vec{a}_t+F.(\vec{b}_t-\vec{c}_t) \tag{35}$$

where *F* is mutation scaling factor which is a positive constant. *F* can be set a real value between 0 and 2. However Storn and Price suggest $F\in[0.5,1]$. After that, another vector $\vec{z}=\left(z_1,z_2,...,z_i,...,z_n\right)$ as a new individual is obtained by the following function [19]:

$$z_i=\begin{Bmatrix} y_i & if & r_i\le C_r \\ x_i & if & r_i>C_r \end{Bmatrix} \quad i=1..n \tag{36}$$

where r_i is a uniformly distributed random variable ($0\le r_i<1$) and C_r is crossover constant ($C_r\in[0,1]$). The final step in DE algorithm is the selection of the better individual for the minimization of the objective function $f(\vec{x})$. This process can be defined as follows [19]:

$$x_{t+1}=\begin{Bmatrix} \vec{z} & if & f(\vec{z})\le f(\vec{x}_t) \\ \vec{x}_t & if & f(\vec{z})>f(\vec{x}_t) \end{Bmatrix} \tag{37}$$

The selection process involves a simple replacement of the original individual with the obtained new individual if it has a better fitness.

There are three control parameters in the DE algorithm: the mutation scale factor *F*, the crossover constant C_r, and the population size *NP*. In the standard differential evolution, which was proposed by Price and Storn, these parameters were assumed to be constant. Some researchers tried to improve the performance of differential evolution algorithm by adaptively tuning of these control parameters. Ali and Törn [28] proposed that the mutation scaling factor, *F* should be diversified at early stages and intensified at latter stages. They modified the mutation scale factor by following equation [28]:

$$F=\begin{Bmatrix} \max\left\{l_{\min},1-\left|\frac{f_{\max}}{f_{\min}}\right|\right\} & if & \left|\frac{f_{\max}}{f_{\min}}\right|<1 \\ \max\left\{l_{\min},1-\left|\frac{f_{\min}}{f_{\max}}\right|\right\} & otherwise & \end{Bmatrix} \tag{38}$$

where $l_{\min}=0.4$, f_{min} and f_{max} are the minimum and maximum objective function value over the individual of the populations, which are obtained in a generation.

4. RESULTS AND DISCUSSION

Meshless method has been used for numerical simulation of flow inside a U-turn passage. Meshless characteristic based split (CBS) algorithm together with MLS technique have been employed for this purpose. Differential evolution method has been used for shape optimization of U-turn passage. A new computational code has been implemented using C++ language for this purpose. Figure 2 shows the flow chart of optimization algorithm. Meshless flow solver analyzes fluid flow inside the passage and determine the pressure loss inside U-turn passage. Differential evolution algorithm changes the shape of U-turn using setting parameters. These two modules are performed together as parts of unique computational software. In each iteration, Meshless solver module solves full N-S equations inside U-turn passage and consequently, determine the pressure loss which is the fitness function and gives feedback to the optimization module which modifies the shape of the passage in order to improve fitness function. This process is repeated until the satisfactory results are obtained.

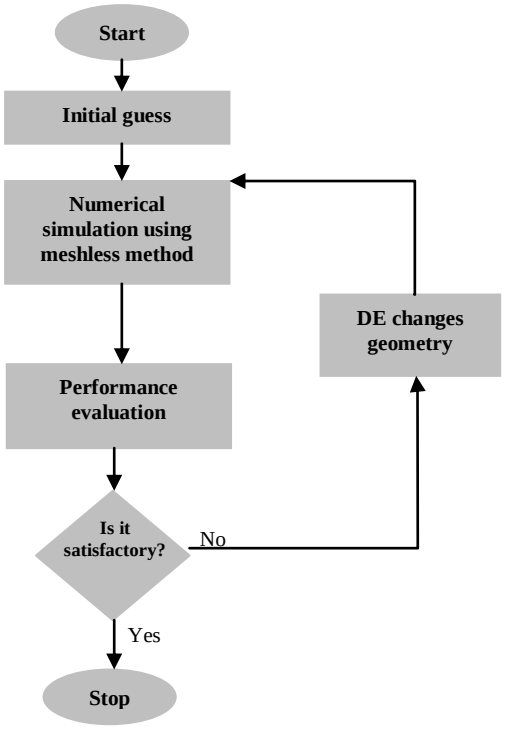

Figure 2 Flow chart of optimization algorithm

4.1. U-TURN GEOMETRY AND BOUNDARY CONDITION DEFINITION

Figure 3 shows the geometry of the U-turn passage. To have more realistic solution, the inlet and outlet lengths of the passage are considered long enough to avoid interference of created vortexes on inlet and outlet boundary conditions. Differential evolution is allowed to change the geometry of bending part of the passage. This part is defined using two curves Γ_1, Γ_2 which, are evaluated in polar coordinate.

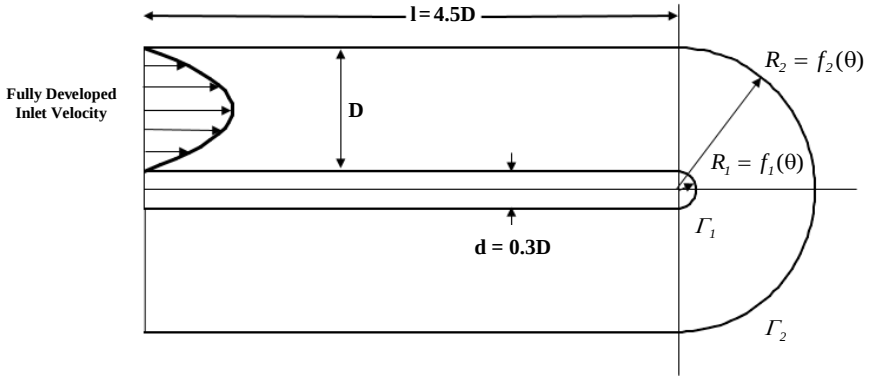

Figure 3 The geometry of U-turn passage

Γ_1, Γ_2 are defined using Fourier series as follows:

$$R_1 = \frac{d}{2} + \sum_{i=1}^{7} A_i Sin(i\theta) \tag{39}$$

$$R_2 = D + \frac{d}{2} + \sum_{i=1}^{7} B_i Sin(i\theta) \tag{40}$$

where A_i and B_i are control parameters that DE algorithm tunes them to change the geometry of the U-turn channel. Fourier based functions are orthogonal functions that are capable of approximating any bounded continuous function with the desired order of accuracy. Figure 4 shows the first six Fourier based functions.

The flow has been assumed to be laminar and two dimensional in this work. The boundary conditions of the flow are fully developed parabolic velocity at the inlet, no slip at the walls and constant pressure at the outlet.

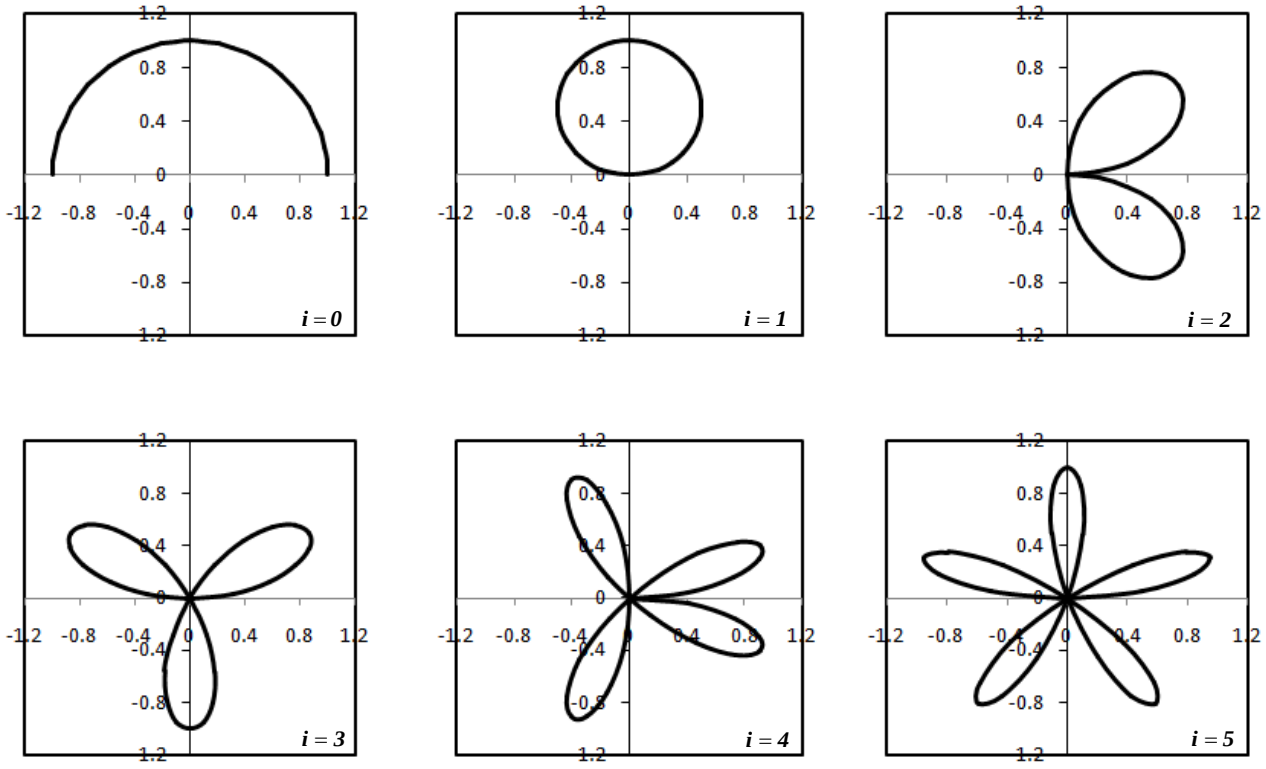

Figure 4 The first six Fourier based function

4.2. CONVERGENCY OF THE SOLUTION

Before using Meshless CBS algorithm for shape optimization of U-turn passage it is necessary to investigate the accuracy and convergency of the method. The results of Meshless algorithm has been validated with several other experimental, theoretical and numerical results in the literature. The reader is encouraged to read the references [9] to [11].

In U-turn flow problem the governing partial differential equations of fluid flow have been solved using Meshless CBS algorithm with different number of nodes. Figure 5 shows the l_2 pressure norm of the solution in different number of nodes.

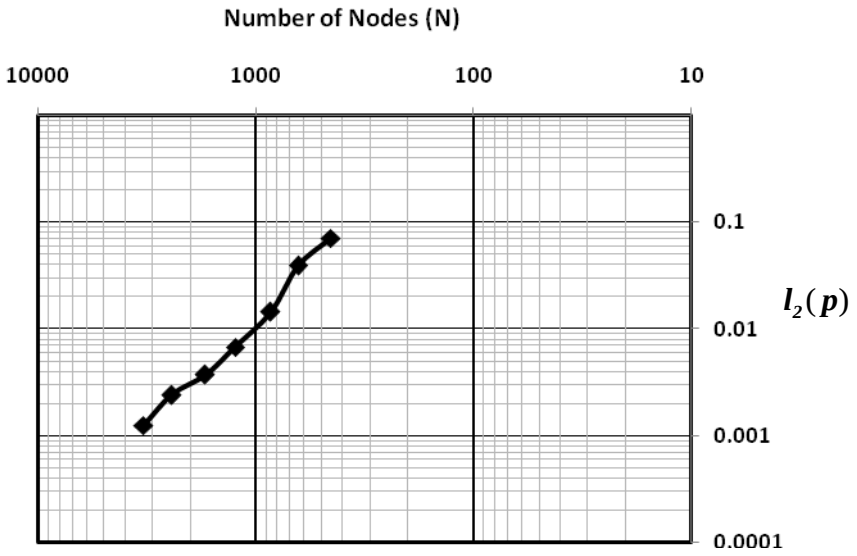

Figure 5 l_2 pressure norm of the solution in different number of nodes

In the above graph, $l_2(p)$ has been determined using following formula:

$$l_2(p) = \left(\frac{\int_{\Omega} (p - \hat{p})^2 d\Omega}{\int \hat{p}^2 d\Omega} \right)^{\frac{1}{2}} \tag{41}$$

where p is the approximated pressure of an arbitrary point inside problem domain and $\hat{p}$ is accurate amount of pressure in the same point in the problem domain.

4.3. SHAPE OPTIMIZATION RESULTS AND DISCUSSION

To start the shape optimization of the channel, it is necessary to randomly create the first generation. After that Meshless flow solver determines the fitness function of each individuals and deferential evolution algorithm creates new generations. Since 14 design variables are considered for optimization of the channel, the population size is considered to be fixed and equal to 40 individuals. The flow is assumed to be laminar with the Reynolds number equal to 133.33. The Reynolds number is defined as follows:

$$\mathrm{Re} = \frac{\rho \mathrm{u}_{\mathrm{ave}} \mathrm{d}}{\eta} \tag{42}$$

Figure 6 shows 3775 nodes which have been used in meshless method for numerical simulation of the flow inside U-turn passage. Using adaptive differential evolution algorithm pressure loss has been minimized inside U-turn passage. Results obtained show 24 percent improvement of pressure loss using optimized U-turn shape compared with initial guess. Pressure loss has been determined from the following formula:

$$\mathrm{fitness} = \frac{\frac{1}{\mathrm{A}_{\mathrm{inlet}}} \iint_{\mathrm{inlet}} \mathrm{PdA} - \frac{1}{\mathrm{A}_{\mathrm{outlet}}} \iint_{\mathrm{outlet}} \mathrm{PdA}}{\frac{1}{2} \rho \mathrm{u}_{\mathrm{ave}}^2} \tag{43}$$

where A_{inlet} and A_{outlet} are cross section area of the channel in the inlet and the outlet of the channel.

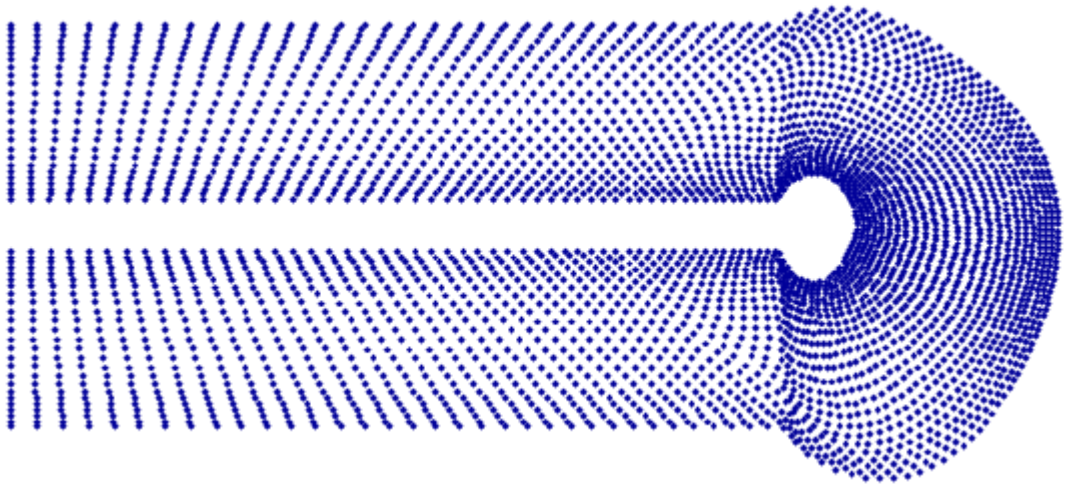

Figure 6 Nodes used in meshless method for numerical simulation of the flow inside U-turn passage

Figure 7 compares the amount of pressure along mid-line of the initial shape of channel with that of the optimal channel. This graph shows that the absolute amount of the maximum and the minimum pressure is smaller inside optimized shape compared to initial shape. Consequently the difference between two picks of the pressure curve of the optimal shape is milder compared with the initial shape. Figure 8 compares the amount of shear stress along inner wall of the initial shape of channel with that of the optimal shape. According to the results obtained, the direction of shear stress has been reversed after the turn. This is exactly where the flow separation occurs and a vortex has been created. In this graph, it is clear that the area where the shear stress has been reversed which is precisely the area where flow separation and low pressure vortex has been created is smaller in optimized shape compared with the initial shape. Figure 9 compares the amount of shear stress along outer wall of the initial shape of channel with that of the optimal shape. This graph shows that the amount of shear stress of the outer wall of the optimal shape is closer to zero in the U-turn section. Figure 10 shows normalized fitness of the best individual at each generation. Differential evolution algorithm has been able to optimize the geometry of the U-turn passage after 350 generations. Figure 11 shows pressure contours of the flow inside initial shape of channel. Figure 12 shows pressure contours inside optimized geometry. Figure 13 shows the streamlines inside the initial shape of the channel and the optimal channel. According to this figure, the size of the created vortex after the turn, which is the major reason of the pressure loss, is considerably smaller in the optimized channel compared with the initial shape of the channel.

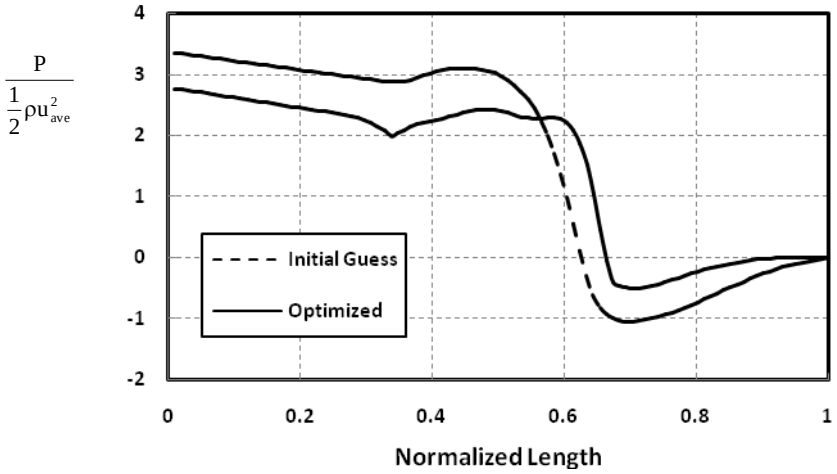

Figure 7 The amount of pressure along mid-line of the initial shape of the channel compared with that of the optimal channel

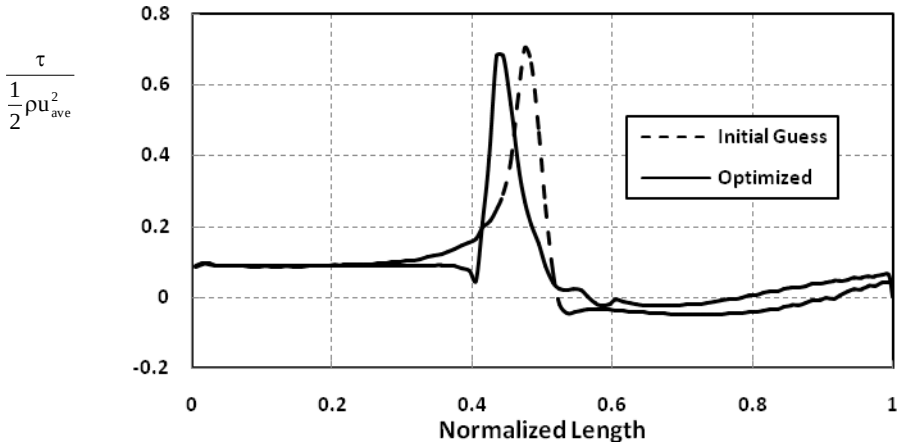

Figure 8 The amount of shear stress along inner wall of the initial shape of the channel compared with that of the optimal channel

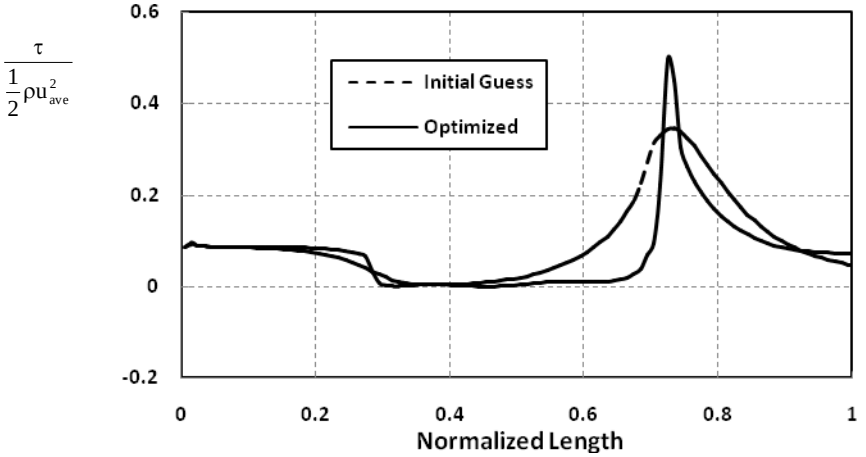

Figure 9 The amount of shear stress along outer wall of the initial shape of the channel compared with that of the optimal channel

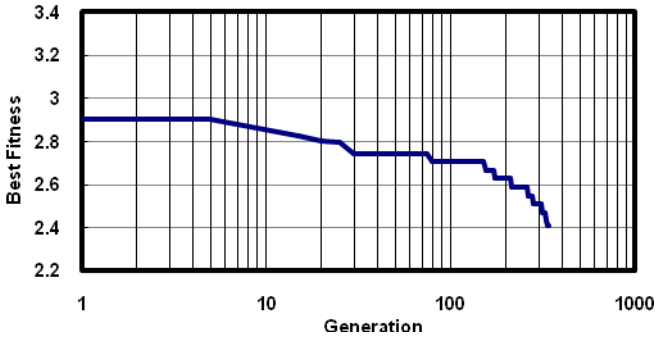

Figure 10 Fitness of the best individual at each generation

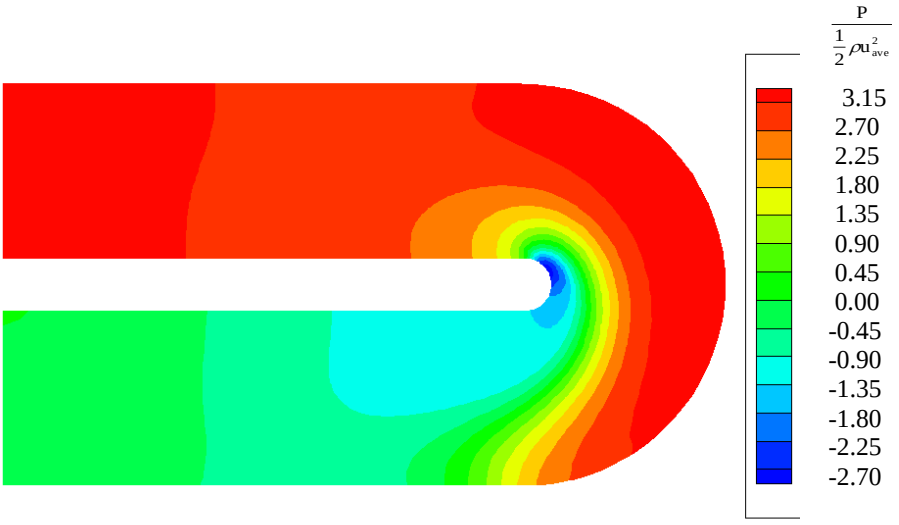

Figure 11 Pressure contours of the flow inside initial shape of the channel

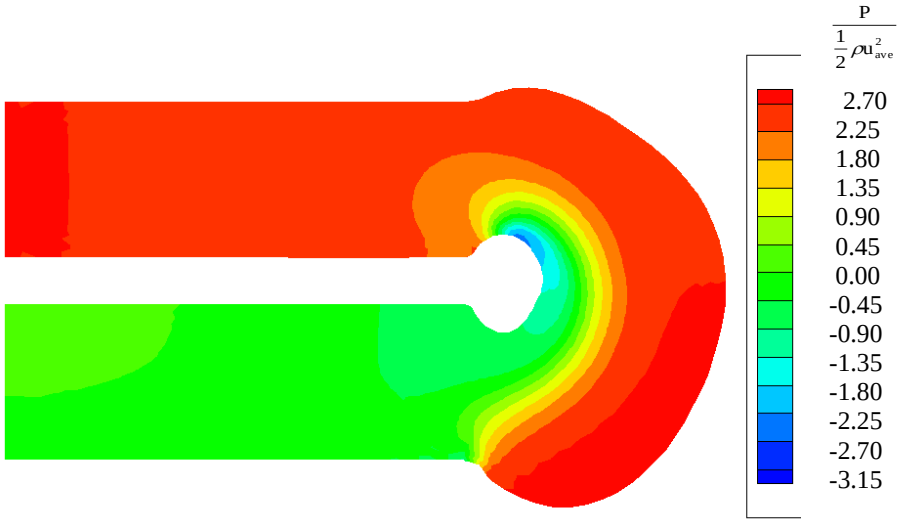

Figure 12 Pressure contours of the flow inside optimized shape of the channel

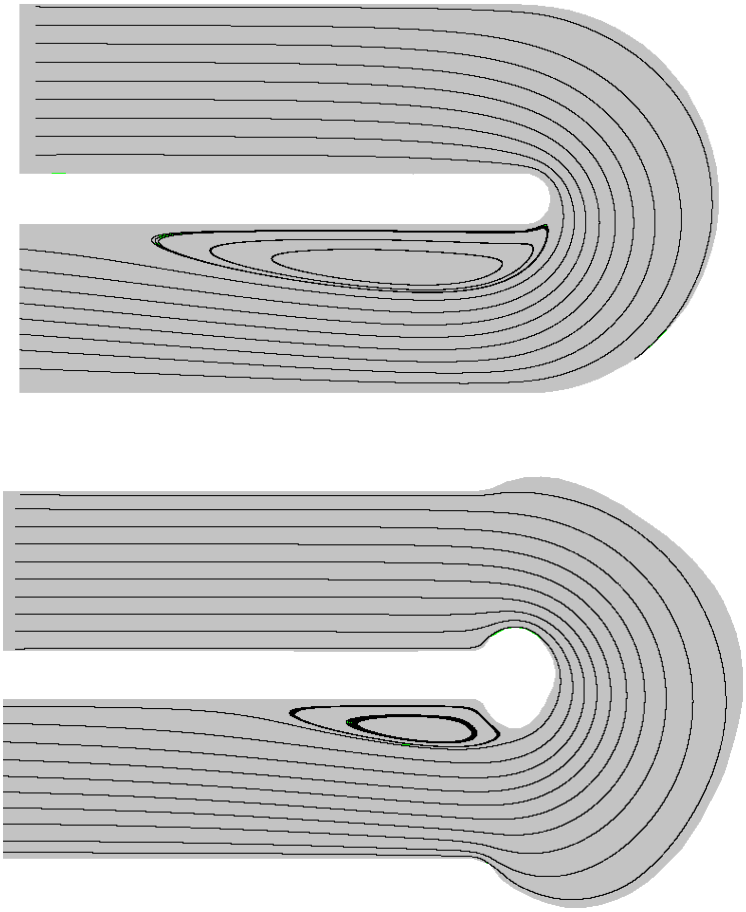

Figure 13 Streamlines inside the initial shape of channel and the optimal channel

5. CONCLUSION

Meshless method has been used for numerical simulation of flow inside a U-turn passage. Meshless characteristic based split (CBS) algorithm has been used for this purpose. Moving least square (MLS) technique has been employed for construction of shape functions. A new computational code has been implemented using C++ language for this purpose. Together with flow simulation code another module has been written for optimization of U-turn passage based on differential evolution (DE) algorithm. Using adaptive differential evolution algorithm together with a meshless computational simulation code, U-turn shape has been modified iteratively. The results show that the pressure loss of the U-turn passage of the optimal design has been decreased significantly compared with the initial guess. According to the results obtained, the size of the created vortex after the turn, which is the major reason of the pressure loss, is considerably smaller in the optimized channel compared with the initial shape of the channel. Differential evolution algorithm has been able to improve pressure loss inside U-turn passage by 24 percent after 350 generations.

REFERENCES

[1] Liu, G. R. , *Mesh Free Methods*, 1st edition., 2002, CRS Press LLC.
[2] Lucy, L., A Numerical Approach to Testing the Fission Hypotheses, *Astron J*., 1977; 82: 1013-1024.
[3] Gingold, R. A. and Monaghan, J. J., Smooth particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. R. Astron. Sc.*, 1977; 81: 375-389.
[4] Shamekhi, A. and Nai, M. H., Buckling Analysis of Circular FGM Plate Having Variable Thickness Under Uniform Compression by Mesh Free Method, Int. J. Compt. Methods, 2005; 2(3):327-340.
[5] Nik-khah Bahrami, M. and Shamekhi, A., Using Mesh Free Method for Free Vibration Analysis of Circular FGM Plates, *ICCES MM 2006 conference*, Dubrovnik, Croatia.
[6] Atluri, S. N. and Tulong Zhu, New Concepts in Meshless Method, Int. J. Numer. Meth. Engng., 2000; 47: 537-556.
[7] 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.

[8] Yagawa G, Shirazaki M. Parallel computing for incompressible flow using a nodal-based method. Computational Mechanics 1999; 23:209–217.
[9] Shamekhi, A. and K. Sadeghy, Cavity Flow Simulation of Carreau-Yasuda Non-Newtonian Fluids Using PIM Meshfree Method, Appl. Math. Modelling, Vol. 33, No. 11, pp 4131-4145, 2009.
[10] Shamekhi, A. et. al. , Using Mesh Free Method for Numerical Simulation of Non-Newtonian Fluid Flow Over a Step, Journal of Society of Rheology, Japan, Vol. 36, No. 1, pp 19-27, 2008.
[11] Shamekhi, A. and Sadeghy, K., On the Use of Characteristic Based Split Mesh Free Method for Solving Flow Problems, International Journal for Numerical Methods in Fluids, Vol. 56, pp 1885-1907, 2007.
[12] Zienkiewicz, O.C. and Taylor, R. L., The Finite Element Method, 5th edition, 2000; Vol. 3, Butterworth-Heinemann.
[13] Liu, W. K., Jun, S., and Zhang, Y., Reproducing kernel particle methods, Int. J. Numer. Methods Fluids, 1995; 20,1081-1106.
[14] Lancaster, P. and Salkauskas, K., Surfaces generated by moving least squares methods, Math. Comput., 1981; 37, 141-158.
[15] Liu, G. R. and Gu, Y. T., A point interpolation method, in Proc. 4th Asia-Pacific Conference on Computational Mechanics, December, Singapore, 1999,1009-1014.
[16] Wang, J. G. and Liu, G. R., Radial point interpolation method for elastoplastic problems, in Proc. 1st Int. Conf, on Structural Stability and Dynamics, December 7-9, Taipei, Taiwan, 2000, 703-708.
[17] Storn, R. and K. Price, Minimizing the real functions of the ICEC'96 contest by differential evolution, Proceedings of the IEEE Conference on Evolutionary Computation, pp 842-844, 1996.
[18] Price, K. and R. Storn, Genetic algorithms: Differential evolution, Dr. Dobb's Journal of Software Tools for Professional Programmer, v 22, n 4, pp 18, Apr 1997.
[19] Bin Xin, Jie Chen, Juan Zhang, Hao Fang, and Zhi-Hong Peng, Hybridizing Differential Evolution and Particle Swarm Optimization to Design Powerful Optimizers: A Review and Taxonomy, IEEE Transaction on Systems, Man, and Cybernetics—Part C: Application and Reviews, 2011.
[20] Montgomery, D. C., *Design and Analysis of Experiments*. JohnWiley ans Sons, New York, 1997.
[21] Chorin, A.J., Numerical solution of Navier-Stokes equations. Math. Comput., 1968; 22: 745-62.
[22] Comini, G. and S. Del Guidice. Finite element solution of incompressible Navier-Stokes equations. Num. Heat Transfer, Part A, 1972; 5, 463-78.
[23] Kawahara, M. and K. Ohmiya. Finite element analysis of density flow using the velocity correction method. Int. J. Num. Meth. Fluids, 1985; 5, 981-93.
[24] Ramaswamy, B., T.C. Jue and J.E. Akin. Semi-implicit and explicit finite element schemes for coupled fluid thermal problems. Int. J. Num. Meth. Eng., 1992; 34, 675-96.
[25] Zienkiewicz, O.C. and Codina, R., A General Algorithm for Compressible and Incompressible Flow, Part I. The Split Characteristic Based Scheme, Int. J. Num. Meth. Fluids, 1995; 20: 869-885.
[26] Liu, W. K., Jun, S., and Zhang, Y., Reproducing kernel particle methods, Int. J. Numer. Methods Fluids, 1995; 20,1081-1106.
[27] Nayroles, B., Touzot, G., and Villon, P., Generalizing the finite element method: diffuse approximation and diffuse elements, Comput. Mech., 1992; 10, 307-318.

[28] M. M. Ali and A. Törn, "Population set based global optimization algorithms: Some modifications and numerical studies," Comput. Oper. Res., vol. 31, no. 10, pp. 1703–1725, 2004.
[29] Holland, J. H., *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, 1975.
[30] Goldberg, D. E. and Samtani, M. P., Engineering Optimization Via Genetic Algorithm, ASCE, p 471-482, 1986.
[31] Grefenstette, J. J., Optimization of Control Parameters for Genetic Algorithm, IEEE Transactions on Systems, Man and Cybernetics, v SMC-16, n 1, p 122-128, 1986.
[32] Goldberg, D. E. and Chie, H. K., Genetic algorithm in Pipeline Optimization, Journal of Computing in Civil Engineering, v 1, n 2, p 128-141, 1987.
[33] R. Storn and K. V. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, 1995.
[34] Madavan, N. K., On improving efficiency of differential evolution for aerodynamic shape optimization applications, Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, v 6, p 3590-3609, 2004.
[35] Song, L. and Feng, Z. and Li, J., Shape optimization of turbine stage using adaptive range differential evolution and three-dimensional Navier-stokes solver, Proceedings of the ASME Turbo Expo, v 6 PART B, p 1033-1040, 2005.
[36] Chi, Y. C. and Fang, J. and Rao, D. L. and Cai, G. B., Self-adaptive center-mutation differential evolution and its application to shape optimization design of a turbine blade, Hangkong Dongli Xuebao/Journal of Aerospace Power, v 25, n 8, p 1849-1854, 2010.