A MESHFREE APPROACH FOR SIMULATION OF FLUID FLOW

Fernanda Olegario dos Santos
Escola de Engenharia de São Carlos, Departamento de Engenharia Mecânica, Universidade de São Paulo, Av. Trabalhador São-Carlense 400, Cxp 668, 13560-970, São Carlos-SP, Brasil
feroleg@sc.usp.br

Antonio Castelo Filho
Instituto de Ciências Matemáticas e de Computação, Departamento de Matemática Aplicada e Estatística, Universidade de São Paulo, Av. Trabalhador São-Carlense 400, Cxp 668, 13560-970, São Carlos-SP, Brasil
castelo@icmc.usp.br

Luis Gustavo Nonato
Instituto de Ciências Matemáticas e de Computação, Departamento de Matemática Aplicada e Estatística, Universidade de São Paulo, Av. Trabalhador São-Carlense 400, Cxp 668, 13560-970, São Carlos-SP, Brasil
gnonato@icmc.usp.br

Abstract. *Meshfree fluid flow simulation has achieved large popularity in the last years. Meshfree Galerkin Methods and Smooth Particle Hydrodynamics are typical examples of meshfree techniques, whose ability to handle complex problems has motivated the CFD community. In this work we present a new meshfree approach that uses moving least square (MLS) to discretize the model equations. A mesh is only employed to manage the neighborhood relationship of points spread within the domain, avoiding thus the problem of keeping a good quality mesh. This strategy was implemented in the integrated simulation system, called UmFlow-2D, whose purpose is to simulate incompressible fluid flow on unstructed mesh. The system is divided in three modules: modeling, simulation and visualization modules. The simulation module implements the Navier-Stokes equation, which are discretized by a Generalized Finite Difference Method. In particular convective terms are discretized by a semi-lagrangean technique. A projection method is employed to uncouple the velocity componentes and pressure. Results of numerical simulations proving the effectiveness of our approach in two-dimensional fluid flow simulations are presented and discussed.*

Keywords: *Moving Least Square, Numerical Simulation, Generalized Finite Difference Method, Meshfree Discretization.*

1. INTRODUCTION

The numerical treatment of partial differential equations with meshfree discretization techniques has been a very active research area in recent years. The need for new techniques for the solution of problems where the classical numerical methods fail or are prohibitively expensive has motivated the development of this new approaches. Aiming at avoiding difficulties as the generation of good quality meshes and mesh distortions in large deformation problems, the meshfree methods try to construct approximation functions in terms of a set of nodes.

The literature has presented a set of different meshfree methods, according to computational modeling, the meshfree methods may be put into two different classes, [Li and Liu, 2004] those that approximate the strong form of a partial differential equation (PDE), this techniques, in general, discretize the PDE by a collocation technique. Examples of such methods are generalized finite difference method (GFDM) [Liszka and Orkisz, 1980] and smoothed particle hydrodynamics (SPH) [Monaghan, 1988] . The methods in the second class, i.e., serving as approximations of the weak from of a PDE, are often Galerkin weak formulations (meshfree Galerkin methods). Examples of such an approach are element-free Galerkin method (EFGM) [Belytschko, 1994], diffuse element method (DEM)[Nayroles et al. (1992)], reproducing kernel particle methods (RKPM)[Liu et al. (1995)], and partition of unit method (PUM) [Melenk and Babuska, 1996].

In this work we present a new meshfree method that approximates the strong form of a PDE. Our approach estimates the derivatives involved in a PDE from a polynomial approximation conducted in each discretized node. Different from GFDM methods, which use the classical Taylor series expansion to calculate the polynomial from which the derivatives are extracted, our strategy adopts a more flexible scheme to compute the polynomial approximation, namely the moving least square (MLS)[Levin, 1998]. The moving least square presents some advantages over Taylor series expansion. For example, the weight assignment, usually employed to control the contribution of neighbor nodes to the polynomial approximation, can be accomplished in a more straight way by MLS. Furthermore, MLS can be combined with partition of unity in order to tackling the problem of the number of neighbor nodes properly. In order to show the effectiveness of the proposed technique, we present an integrated simulation system, called UmFlow-2D, wich aims at simulating two-dimensional incompressible fluid flow using unstructed mesh. The system is divided in three modules: modeling, simulation and visualization modules. The modeler module is used to define a fluid, boundary conditions, and others simulation's datas. The simulation module implements the Navier-Stokes equation and boundary bonditions. The visualization module is a system that permit the visualization of result and implements several visualization method.

In this paper we presents the simulation module. This module implements the Navier-Stokes equation in a computacional code. The governing equations are discretized by a generalized finite difference method with the convective terms discretized by a semi-Lagrangian scheme. A projection method is employed to uncouple the velocity componentes and pressure. The strategy employed to solve the Poisson's equation generated from our discretization strategy is another novelty of this work. The details of such a modeling is also presented.

The work is organized as follows: Section 2 presents the least square discretization method proposed in this work. A description of how to employ such a discretization method in Navie-Stokes equations is discussed in section 3. The scheme adopted to define the boundary conditions is presented in section 3.1. Section 4 presents some results obtained from the proposed approach. Conclusions and future work are in section 5.

2. Least Square Approximation

In this section we present some basic definitions and notation employed in the remaining of the text and the Least Square Approximation.

2.1 Computional Node Arrangement

Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of discrete nodes representing a domain $D \subset \mathbb{R}^2$. For each node $v_i \in V$ we define the *local coordinate system of* v_i by writing any point $\mathbf{r} = (x, y) \in D$ as $\bar{\mathbf{r}}_i = \mathbf{r} - \mathbf{r}_i$, where $\mathbf{r}_i = (x_i, y_i)$ are the coordinates of v_i. We denote by $\bar{\mathbf{r}}_{k,i} = \mathbf{r}_k - \mathbf{r}_i$ the coordinates of a node $v_k \in V$ written in the local coordinate system of v_i.

Let $S_i \subset V$ be a non-empty subset of nodes, S_i is a *star of* v_i, if the two conditions bellow are satisfied:

1. if $\|\bar{\mathbf{r}}_{s,i}\| \leq \|\bar{\mathbf{r}}_{k,i}\|, \forall v_k \in V,\ k \neq s$ then $v_s \in S_i$
2. if v_s is in the convex hull of S then $v_s \in S_i$

The *local minimum length* of a star S_i is defined as:

$$h_i = \min_{v_s \in S_i} \|\bar{\mathbf{r}}_{s,i}\| \tag{1}$$

Notice that the local minimum length is the same for all stars of v_i. From the definition of local minimum length we can define the *global minimum length* with respect to V:

$$h = \min_{v_i \in V} h_i \tag{2}$$

in another words, the global minimum length h is the shortest distance of the nodes representing D.

2.2 Least Square Approximation

Let $v_i \in V$ be a node in the domain D and S_i be a star of v_i. Suppose that $f : D \to \mathbb{R}$ is a real function defined in D. We aim at approximating f in a neighborhood of v_i by a function $\bar{f}$ of the form:

$$\bar{f}_i(\bar{\mathbf{r}}) = f(\mathbf{r}_i) + W_i(\bar{\mathbf{r}}) \tag{3}$$

where W_i is a polynomial of degree d that can be written as:

$$W_i(\bar{\mathbf{r}}) = \sum_{j=1}^{N} c_j P^{(j)}. \tag{4}$$

The terms $P^{(j)}$ in Eq. (4) forms a basis of monomial $\{x, y, x^2, xy, y^2, \ldots\}$, which can be numbered as in Fig. (1). Notice that the constant monomial is not considered, as the polynomial will be employed to approximate derivatives, thus the constant term can be neglected.

Given the values of f in each node $v_k \in S_i$, we can compute the coefficients c_j of W_i by solving the linear system $Ac = B$:

$$\begin{bmatrix} a_{11} & \cdots & a_{1N} \\ & \vdots & \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} \tag{5}$$

Monomial Basis	
monomial	Degree
$P^{(1)} = x$	1
$P^{(2)} = y$	1
$P^{(3)} = x^2$	2
$P^{(4)} = xy$	2
$P^{(5)} = y^2$	2
$P^{(6)} = x^3$	3
$P^{(7)} = x^2y$	3
$P^{(8)} = xy^2$	3
$P^{(9)} = y^3$	3

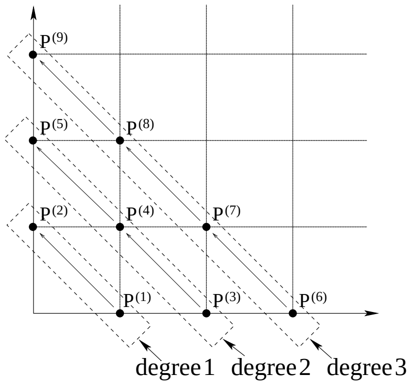

Figure 1. Monomial basis and numbering scheme.

where the elements a_{ij} of the matrix A and the elements b_i of vector b are given by:

$$a_{ij} = \sum_{v_k \in S_i} P^{(i)}(\bar{\mathbf{r}}_k)P^{(j)}(\bar{\mathbf{r}}_k)w_k; \quad i,j = 1,\ldots,N \tag{6}$$

$$b_i = \sum_{v_k \in S_i} (f(\mathbf{r}_k) - f(\mathbf{r}_i))P^{(i)}(\bar{\mathbf{r}}_k)w_k \tag{7}$$

As can be seen from Eq. (6) and (7), using the MLS we are assigning weights w_k for the node $v_k \in S_i$. Such weights can depend on the distance between v_k and v_i or they can be a Gaussian in v_i. It is important to point out that the rank of A depends on the number of elements in S_i. For example, for a quadratic polynomial approximation there will be needed at least five nodes in S_i. The higher the degree of W_i the more nodes are needed.

Once the coefficients c_j have been computed, the derivatives of f can be approximated in v_i by the derivatives of $\bar{f}_i$. Furthermore, if $\bar{f}_i$ is a quadratic polynomial then the second order derivatives are given directly from the coefficients c_j, i.e.,

$$\begin{aligned} \frac{\partial^2 \bar{f}_i}{\partial \bar{x}^2} &= \frac{\partial^2 W_i}{\partial \bar{x}^2} = 2c_3 \\ \frac{\partial^2 \bar{f}_i}{\partial \bar{x}\partial \bar{y}} &= \frac{\partial^2 W_i}{\partial \bar{x}\partial \bar{y}} = c_4 \\ \frac{\partial^2 \bar{f}_i}{\partial \bar{y}^2} &= \frac{\partial^2 W_i}{\partial \bar{y}^2} = 2c_5 \end{aligned} \tag{8}$$

It can be shown that the discretization strategy presented above is consistent if the nodes in S_i are distributed properly. Details about this theoretical result can be found in Peña's master dissertation [Peña, 2003]. In order to verify the effectiveness of the scheme above in numerical simulations, we apply the proposed strategy in an incompressible fluid flow simulation problem. How to conduct the discretization of the Navier-Stokes equations from our approach is the subject of the next section.

3. Discretizing Navier-Stokes Equations

Although the discretization technique presented in the last section has been developed for meshfree domain decompositions, we prefer using a mesh to make the access to the neighborhood of a node easier. To this end, the set of nodes representing a domain D has been input in a Delaunay mesh generator. It is not difficult to show that Delaunay meshes guarantee the first condition of the definition of a star. Without any post-processing a Delaunay mesh satisfy the second condition in almost every node. Steiner points can be inserted if it is strongly necessary to respect condition 2 of the definition of a star.

Pressure discretization will also be making use of the mesh, as we are storing the pressure on the triangular cells. It is worth mentioning that the velocity field is stored on the nodes as in Fig (2). Such a scheme has been adopted in order to make velocity and pressure decoupling easier.

Consider the Navier-Stokes and continuity equations:

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{Fr^2}\mathbf{g}, \tag{9}$$

$$\nabla \cdot \mathbf{u} = 0 \; . \tag{10}$$

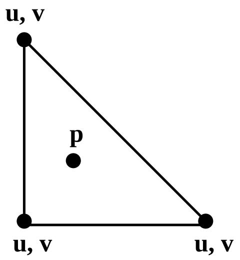

Figure 2. Computational cell.

where $\mathbf{u}$ is the velocity and p is the pressure of the fluid, $Re = LU/\nu$ is the Reynolds number and $Fr = U/\sqrt{gL}$ is the Froude number.

The material derivative $\frac{D\mathbf{u}}{Dt}$ is discretized by the semi-Lagrangian method:

$$\frac{D\mathbf{u}}{Dt} = \frac{\mathbf{u}(\mathbf{x}, t+\delta t) - \mathbf{u}(\mathbf{x}-\delta\mathbf{x}, t)}{\delta t}. \tag{11}$$

where $\delta\mathbf{x} = \mathbf{u}\delta t$.

Using the fractionary step method (projection method), we obtain the set of equations:

$$\frac{\tilde{\mathbf{u}}(\mathbf{x}, t+\delta t) - \mathbf{u}(\mathbf{x}-\delta\mathbf{x}, t)}{\delta t} = \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{Fr^2}\mathbf{g}, \tag{12}$$

$$\frac{\mathbf{u}(\mathbf{x}, t+\delta t) - \tilde{\mathbf{u}}(\mathbf{x}, t+\delta t)}{\delta t} = -\nabla p^{n+1}, \tag{13}$$

$$\nabla^2 p^{n+1} = \frac{1}{\delta t}\nabla\cdot\tilde{\mathbf{u}}(\mathbf{x}, t+\delta t), \tag{14}$$

From the above equations, the velocity and pressure fields can be computed, for each time step, as follows:

1. Intermediate velocity

$$\tilde{\mathbf{u}} = \mathbf{u}(\mathbf{x}-\delta\mathbf{x}, t) + \delta t\left(\frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{Fr^2}\mathbf{g}\right) \tag{15}$$

2. Intermediate pressure

$$\nabla^2 p^{n+1} = \frac{1}{\delta t}\nabla\cdot\tilde{\mathbf{u}}, \tag{16}$$

using the following boundary conditions

- Homogeneous Neumann condition for rigid contours, given by

$$\frac{\partial p^{n+1}}{\partial n} = 0.$$

This condition is used also on the inflows.

- Homogeneous Dirichlet condition for outflow, i.e., $p^{n+1} = 0$.

3. New velocity

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \delta t\nabla p \tag{17}$$

The term $\mathbf{u}(\mathbf{x}-\delta\mathbf{x}, t)$ in Eq. (15) is computed by linear interpolation of the velocity $\mathbf{u}$ on the nodes v_i, v_j and v_k closest to $\mathbf{x}-\delta\mathbf{x}$. The Laplacian term $\nabla^2\mathbf{u}$ is computed from a least square approximation as described in (9).

After estimating $\tilde{\mathbf{u}}$, we must solve Poisson's equation (16). In fact, this is the hardest step of the scheme. Using a quadratic polynomial for the least square approximation, a 5×5 linear system is obtained:

$$\begin{bmatrix} a_{11} & \cdots & a_{15} \\ \vdots & & \\ a_{51} & \cdots & a_{55} \end{bmatrix}\begin{bmatrix} c_1 \\ \vdots \\ c_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_5 \end{bmatrix} \tag{18}$$

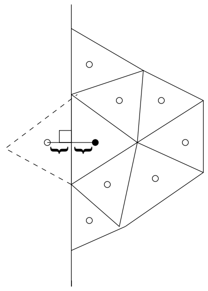

Figure 3. Boundary condition for pressure.

where the elements a_{ij} and b_i are given by Eq. (6) and Eq. (7) respectively.

Using Gaussian elimination we can re-write the system (18) as:

$$\begin{bmatrix} \hat{a}_{11} & \cdots & \hat{a}_{15} \\ & \ddots & \\ & & \hat{a}_{55} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_5 \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_5 \end{bmatrix}. \tag{19}$$

By backward substitution one can obtain the coefficients c_5 and c_3 that are involved in the discretization of $\nabla^2 p$, and they can be written as:

$$c_3 = \sum_{v_k \in S_i} \alpha_k p(r_k) + \alpha_i p(r_i) \tag{20}$$

$$c_5 = \sum_{v_k \in S_i} \beta_k p(r_k) + \beta_i p(r_i) \tag{21}$$

where α_k and β_k are constants obtained from the Gaussian elimination process.

In that way, the Poisson matrix is sparse and non-symmetric. In our implementation we employ the bi-conjugate gradient method [Saad, 2003] to solve the resulting linear system.

Once p has been calculated, moving least square can be employed to approximate ∇p, making it possible to solve Eq. (17).

3.1 Boundary Conditions

The boundary conditions employed in our discretization scheme have been discussed. In fact, we must handle three different types of boundary: rigid contours, inflow and outflow.

For rigid contours two different boundary conditions have been implemented in our code: no slip and free slip. In the first case the velocity is set to zero in all nodes defining the rigid contours. The free slip condition imposes that the velocity in the normal direction be zero and the derivative of the tangential velocity with respect to the normal direction is also zero.

On the inflows, the velocity is given in the normal direction, being zero in the tangential direction.

On the outflows, the derivative of the normal component of the velocity with respect to the normal direction is zero and the derivative of the tangential velocity with respect to the normal direction is also zero.

The boundary conditions for pressure is introducted for each triangular element at contour an "imaginary" triangular element, increasing a neighborhood in the cell of approach of the derivatives as in Fig. (3).

- Homogeneous Neumann condition for rigid contours, given by

  $$\frac{p^{(n+1)}(r_f) - p^{(n+1)}(r_0)}{\mathit{dist}(r_f \rightarrow r_0)} = 0.$$

  The value of p in the triangular element imaginary is equal to the value of p in the triangular element of contour.

  $$b_i = \sum_{v_k \in C_0} (p^{(n+1)}(\mathbf{r}_k) - p^{(n+1)}(\mathbf{r}_0)) P_k^{(j)} \omega_k \tag{22}$$

  The independent term b_i is not modified.

- Homogeneous Dirichlet condition for outflow, i.e.,

  $$p = 0.$$

  In this case the value of p in the element imaginary is

  $$p(r_f) = -p(r_i).$$

  and the independent term is modified

  $$b_i = \sum_{v_k \in C_0} (p^{(n+1)}(\mathbf{r}_k) - p^{(n+1)}(\mathbf{r}_0))P_k^{(j)}\omega_k$$

  Getting $2p_0P_f^i$ where P_f^i is the monomial i of the triangular element imaginary.

4. Results

In order to illustrate the effectiveness of our discretization technique, we present two examples of simulations. The first example shows the classical fluid flow simulation in a channel. The second example aims at illustrating the behavior of our approach in a flow over a circular cylinder.

4.1 Flow in a Channel

The well known *Hagen-Poiseuille* flow has been chosen to validate our numerical method, as an analytical solution is available. This simulation consists of a flow between two parallel plates, as illustrated in Fig. (4).

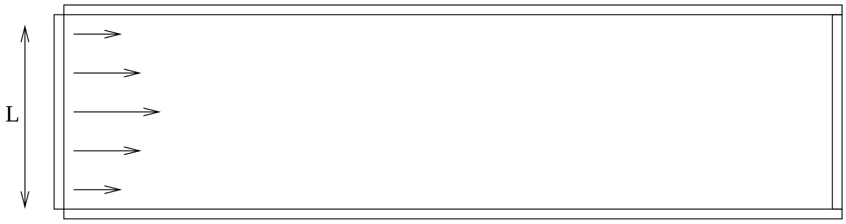

Figure 4. *Hagen-Poiseuille* flow.

The analytical solution for Hagen-Poiseuille flow, which can be found in Batchelor [Batchelor 1970], is given by:

$$u(y) = -\frac{1}{2\mu}\frac{\partial p}{\partial x}(yL - y^2), \tag{23}$$

where μ is the viscosity and the velocity u is a function of the distance y to the wall. Considering L to be the width of the channel, the pressure gradient can be written as:

$$\frac{\partial p}{\partial x} = -12\frac{\mu Q}{L^3}, \tag{24}$$

where Q is defined by:

$$Q = \int_0^L u(y)dy. \tag{25}$$

Considering $u(y) = U\left(1 - \frac{y^2}{L^2}\right)$ on the inflow, where U is the reference velocity, and choosing $L = U = 1$, the analytical solution for the parabolic profile, is:

$$u(y) = -4y(y - 1), \tag{26}$$

Three different meshes have been employed to show the convergence of our method: a course mesh with 193 cells (M1), an intermediate mesh containing 728 cells (M2), and a refined mesh with 2853 cells (M3). The parameters of the simulation have been set as: domain: $3m \times 1m$; Viscosity: $0.10Ns/m^2$; Density: $0.10Kg/m^3$; Reynolds: $Re = 1$; Froude: $Fr = 0.319275$, with $g = 9.81$ and $\mathbf{g} = (0, 0)$. Figure (5) shows the intermediate mesh and Fig. (6) presents a qualitative map of the velocity in x.

Figure (7) shows a comparison between the analytical and numerical solution on a line in the middle of the channel. One can observe that in the refined mesh it is difficult to distinguish the analytical from the numerical solution.

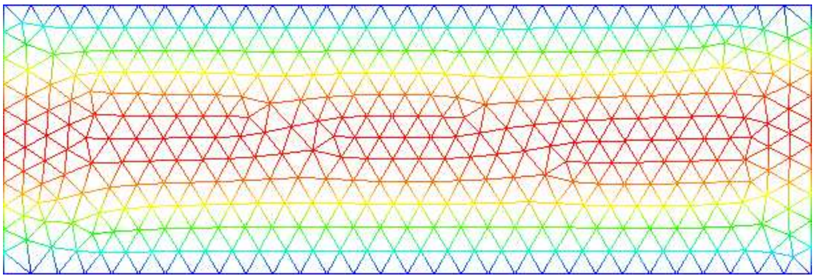

Figure 5. Intermediate mesh with 728 cells.

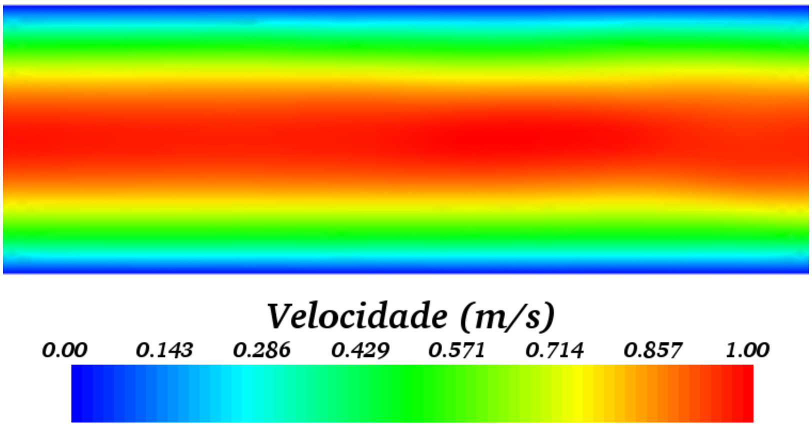

Figure 6. Numerical simulation for the Hagen-Poiseuille flow with profile of parabolic: velocity field in x direction, calculated in a refined mesh.

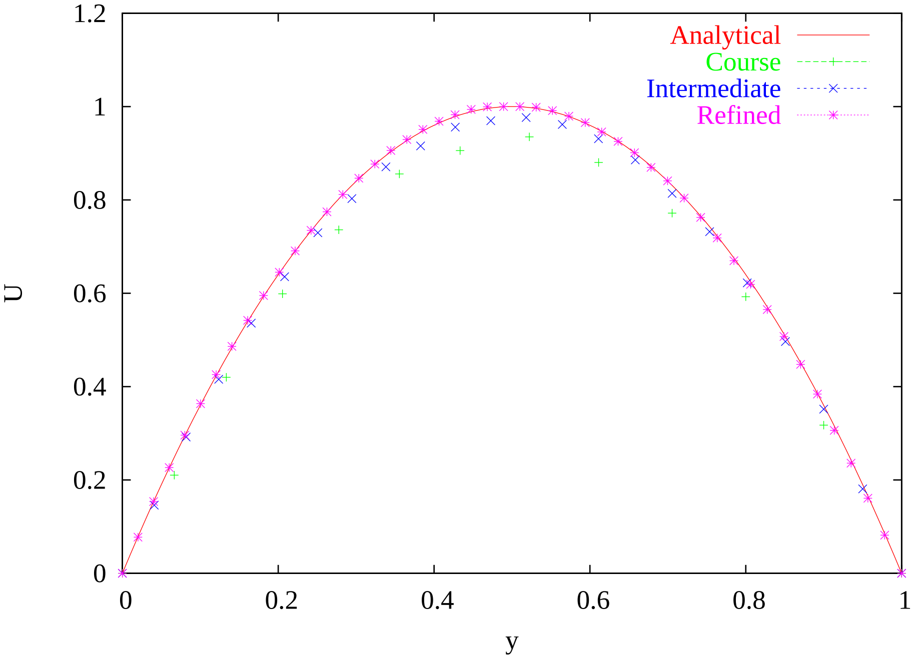

Figure 7. Comparing analytical and numerical results.

4.2 Simulation in the Plate with Contraction

We finish this section with an example illustrating the simulation of bidimensional. The problem of contraction it characterizes for the draining of a fluid through of a channel of diameter L that starts to flow in a channel of diameter minor L/4. The domain of the draining for the problem is presented in the Fig. (8).

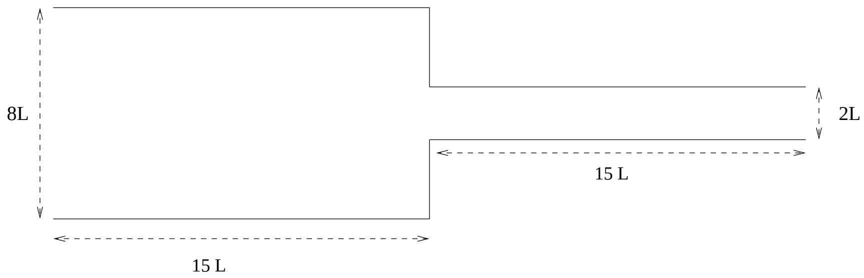

Figure 8. Rectangular plate with contraction.

At the beginning of simulation the domain is completely full with a constant velocity equal $u = 1,0\ ms^{-1}$ and fluid is injected in the entrance of the channel with a profile of velocity of the parabolic type. For a simulation of the problem the following model was used: domain $30m$; diameters: greater $8.0m$ and minor $2.0m$; Parameters of scale: $Uscale = 1.0$ ms^{-1} $Lscale = 1.0\ m$; item Reynolds number (Re): 1 and 100.

Simulations had used a mesh with 5.681 triangular elements ($\delta h \simeq 0.17$ m). The Fig. (9) shows the unstructed mesh where the calculations of solution of the problem had been made.

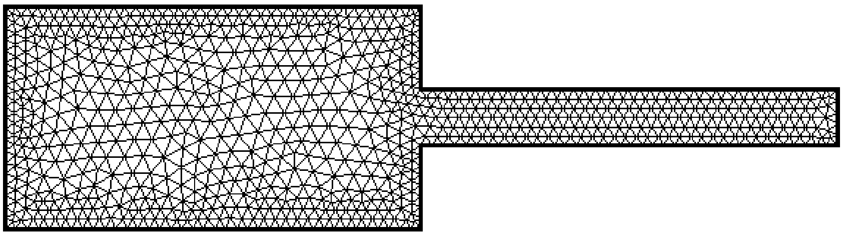

Figure 9. Mesh with 1562 elements.

The Fig. (10) - (13) they show simulations carried through by the UmFlow-2D codes, presenting the velocity fields in directions x and y.

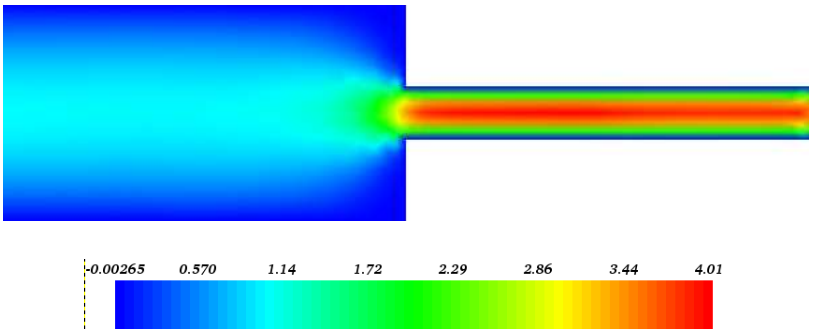

Figure 10. Rectangular plate with contraction: velocity field in x direction, Re = 1.

Notice from Fig. (10) - (13) velocity of fluid is in accordance with what we expected.

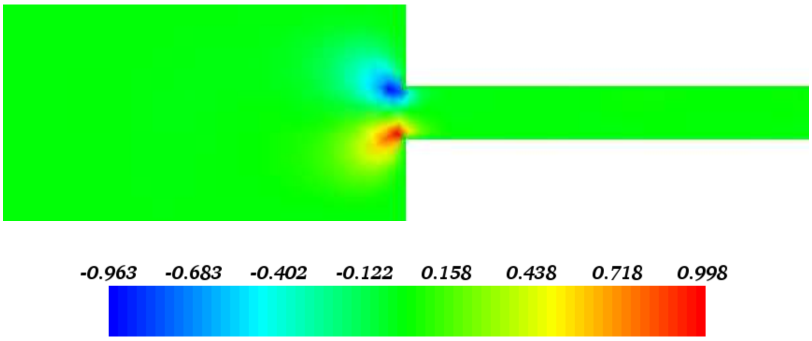

Figure 11. Rectangular plate with contraction: velocity field in y direction, Re = 1.

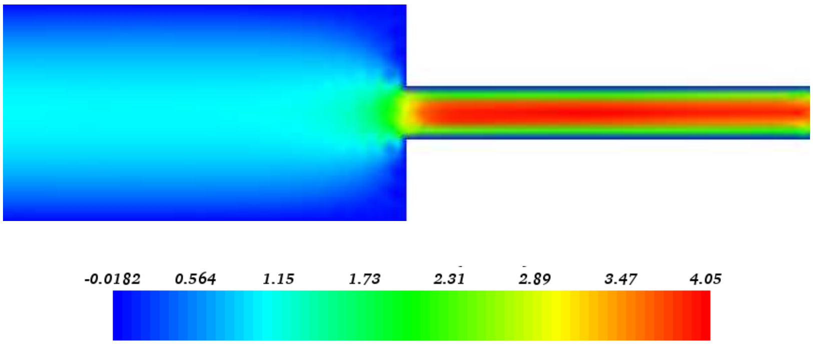

Figure 12. Rectangular plate with contraction: velocity field in x direction, Re = 100.

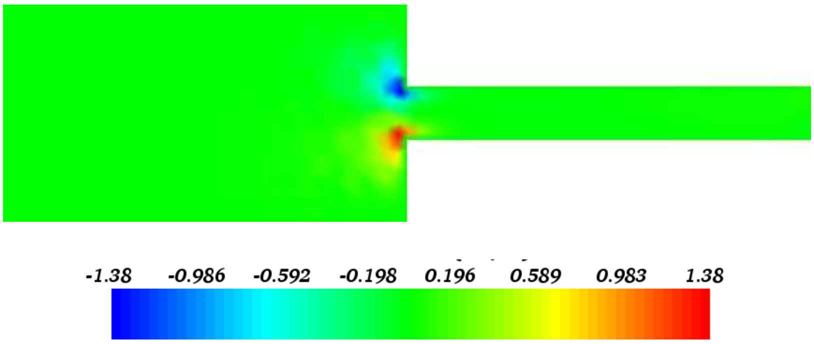

Figure 13. Rectangular plate with contraction: velocity field in y direction, Re = 100.

5. Conclusions and Future Work

In this work we present a new discretization technique that makes use of least square approximation to estimate derivatives. Such an approach has turned out to be very robust in fluid flow simulation, being thus a new alternative for handling these kind of problems. The strategy adopted to build the Poisson's matrix by Gaussian decomposition of the least square matrix is another contribution of this work.

The results of applying the proposed approach in the well known Hagen-Poiseuille flow and in a fluid flow simulation are very consistent, confirming thus the effectiveness of our method.

Although this new methodology has been developed envisioning a complete meshfree discretization scheme, we make use of a triangular mesh to improve the access to nodes neighborhood. In order to get rid of the mesh we are developing a set of data structures devoted to access neighborhood of nodes. A new scheme for discretizing the pressure on the nodes has also been investigated.

Another aspect we are considering is to employ high order semi-Lagrangian schemes, making it possible to deal with higher Reynolds number.

Acknowledgments

We acknowledge the financial support of FAPESP - the State of São Paulo Research Funding Agency (Grant# 02/11686-6) and (Grant# 2004/16064-5)

6. REFERENCES

Batchelor, G.K., 1970, "An Introduction to Fluid Dynamics", Cambridge University Press, Cambridge.

Belytschko, T.,1994, "Element-Free Galerkin Method", Int. J. Numer. Meth. Engrg., Vol.37, pp. 229-256.

Levin, D., 1998, "The approximation power of moving least-squares", J. Math. Comp., Vol 67, No. 224.

Li, S. and Liu, W.K., 2004, "Meshfree Particle Methods", Ed. Springer.

Liszka, T. and Orkisz, J.,1980, "The Finite Difference Method at Arbitrary Irregular Grids ans Its Application in Applied Mechanics", J. Computers and Structures, Vol.11, pp. 83-95.

Liu, W.K and Jun, S. and Li, S. and Adee, J. and Belytschko, T.,1995, "Reproducing kernel particle methods for sctructural dynamics", J. Int. J. Numer. Meth. Engrg., Vol.38, pp. 1655-1679.

Lizier, M. A. S. and Nonato, L. G. and Castelo, A. and Oliveira, M. C. F.,2003, "A Topological Aproach for Detecting and Extracting 2D Object Models from Images", J. Int. J. Numer. Meth. Engrg., São Carlos - São Paulo, pp. 10-17.

Melenk, J.M. and Babuska, I.,1996, "The partition of unity finite element method", Comp. Meth. Appl. Mech. Engrg., Vol. 139, pp. 289-314.

Monaghan, J.J.,1988, "An introduction to SPH", Computer Physics Communications, Vol. 48, pp. 89-96.

Nayroles, B. and Touzot, G. and Villon, P.,1992, "Generalizing the finite element method: diffuse approximation and diffuse elements",Computational Mechanics, Vol. 10, pp. 307-318.

Oliveira, J.,1999, "Desenvolvimento de um Sistema de Simulação de Escoamentos de Fluidos com Superfícies livres Bidimensionais", Dissertação de mestrado, ICMC-USP, Biblioteca, ICMC-USP-São Carlos, Brasil.

Oliveira, J.,2000, "Manual do FreeFlow2D", Relatório Técnico, ICMC-USP-São Carlos, Brasil.

Peña,D.R.I.,2003, "Método de diferenças finitas generalizadas por mínimos quadrados", Dissertação de mestrado, ICMC-USP, Biblioteca, ICMC-USP-São Carlos, Brasil.

Pressman,R.S.,1994, "Software Engineering: a practitioner's approach", European Edition, Rio de Janeiro, 801 p.

Saad,Y.,2003, "Iterative methods for sparse linear systems", Philadelphia:SIAM.

Santos, F.O.,2005, "Simulação numérica de escoamento de fluido utilizando diferenças finitas generalizadas", Dissertação de mestrado, ICMC-USP, Biblioteca, ICMC-USP-São Carlos, Brasil.