

A NEW MESHFREE APPROACH FOR FLUID FLOW SIMULATION WITH FREE SURFACE

F.O. dos Santos*, L.G. Nonato*, A. Castelo*, K.C. Estacio* and N. Mangiavacchi†

*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, So Carlos-SP, Brasil
email: {feroleg,gnonato,castelo,kemelli}@icmc.usp.br

†Faculdade de Engenharia
Departamento de Engenharia Mecânica,
Universidade do Estado do Rio de Janeiro,
Rua So Francisco Xavier, 524, 20550-013, Rio de Janeiro-RJ, Brasil
email: norberto.mangiavacchi@gmail.com

Key Words: Moving Least Square, Meshfree Discretization.

Abstract. *Meshfree fluid flow simulation has achieved large popularity in the last few years. Meshfree Galerkin Methods and Smooth Particle Hydrodynamics are typical examples of mesh-free techniques, whose ability to handle complex problems has motivated the interest in the field. In this work we present a new meshfree strategy that makes use of moving least square (MLS) to discretize the equations. A mesh is only employed to manage the neighborhood relation of points spread within the domain, avoiding thus the problem of keeping a good quality mesh. The modeling of the free surface is based on the volume of fluid (VOF) technique. Distinct from mesh dependent discretization approaches, which estimate the fraction of fluid from the mesh cells, our approach employ the neighborhood relation and a semi-Lagrangian scheme to compute the free surface. Results of numerical simulations proving the effectiveness of our approach in two-dimensional fluid flow simulations are presented and discussed.*

1 INTRODUCTION

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 new approaches, such as meshfree methods. 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, such as generalized finite difference method (GFDM),¹ smoothed particle hydrodynamics (SPH),² element-free Galerkin method (EFGM),³ diffuse element method (DEM),⁴ reproducing kernel particle methods (RKPM),⁵ and partition of unit method (PUM).⁶ According to computational modeling, the meshfree methods may be put into two different classes:⁷ those that approximate the strong form of a partial differential equation (PDE) and those that approximate the weak form of a PDE.

The techniques in the first class, in general, discretize the PDE by a collocation technique. Examples of such methods are SPH and GFDM. The methods in the second class, i.e., serving as approximations of the weak form of a PDE, are often Galerkin weak formulations (meshfree Galerkin methods). Examples of such an approach are EFGM, DEM, RKPM, and PUM.

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).⁸ 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 a free surface fluid flow simulation whose governing equations have been discretized by our approach jointly with a semi-Lagrangian scheme. The strategy employed to solve the Poisson's equation generated from our discretization strategy is another novelty of this work. The free surface is modeled by a scheme similar to VOF.⁹ 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 free surface is presented in section 4. Section 5 presents some results obtained from the proposed approach. Conclusions and future work are in section 6.

2 LEAST SQUARE APPROXIMATION

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

2.1 Star and Node Arrangement

Let $V = \{v_1, v_2, \dots, v_n\}$ 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 $S \subset V$ be a non-empty subset of nodes and $v_i \notin S$ a node of V . The set S is a *star of v_i* , denoted by S_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 $v_s \in S$
2. if v_s is in the convex hull of S then $v_s \in S$

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

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

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 :

$$h = \min_{v_i \in V} h_i \quad (2)$$

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 $f : D \rightarrow \mathbb{R}$ 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:

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

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

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

The terms $P^{(j)}$ in expression (4) forms a basis of monomial $\{x, y, x^2, xy, y^2, \dots\}$, which can be numbered as in figure 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.

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

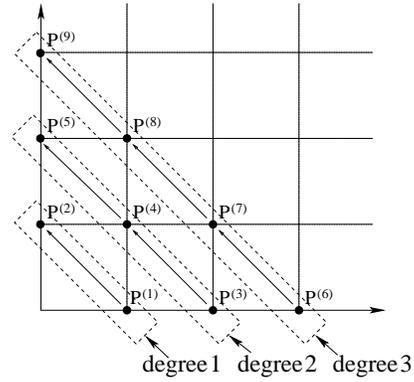


Figure 1: Monomial basis and numbering scheme.

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 $Ac = B$:

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

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

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

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

As can be seen from equations (6) and (7), 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.,

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

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.¹⁰

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 mesh-free 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. Such a scheme has been adopted in order to make velocity and pressure decoupling easier.

Consider the Navier-Stokes equations:

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

$$\nabla \cdot \mathbf{u} = 0. \quad (10)$$

where Re is the Reynolds number and Fr is the Froude number.

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

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

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

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

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

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

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

1. Intermediate velocity

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

2. Intermediate pressure

$$\nabla^2 p = \frac{1}{\delta t} \nabla \cdot \tilde{\mathbf{u}}. \quad (16)$$

3. New velocity

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

The term $\mathbf{u}(\mathbf{x} - \delta\mathbf{x}, t)$ in equation (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:

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

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

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

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

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:

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

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

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 method¹¹ 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 equations (17).

4 BOUNDARY CONDITIONS AND FREE SURFACE MODELING

Up to now, the boundary conditions employed in our discretization scheme have not been discussed. In fact, we must handle four different types of boundary: rigid contours, inflow, outflow and free surface.

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 pressure is set to zero and the derivative of the normal component of the velocity with respect to the normal direction is zero.

The free surface model is based on the volume of fluid (VOF) method,⁹ with some special features. The volume of a cell is represented by a scalar obtained from a function $\varphi : T \rightarrow [0, 1]$, where T is the set of triangles (cells) decomposing the domain. Intuitively, the function φ represents the volume of fluid in each cell.

The function φ is computed from the transport equation given by:

$$\frac{D\varphi(\sigma)}{Dt} = 0 \quad (22)$$

where σ is a cell.

Equation (22) is also discretized by a semi-Lagrangian scheme, as described in (11).

The boundary conditions for pressure and velocity at the free surface are given by setting the pressure equal to zero and setting $(\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{m} = 0$ for the velocity, where \mathbf{n} and \mathbf{m} are the unit normal and tangential vectors to the free surface. Here \mathbf{T} is the stress tensor defined by:

$$\mathbf{T} = -p\mathbf{I} + \frac{1}{Re}(\nabla\mathbf{u} + \nabla\mathbf{u}^t)$$

As we are imposing $p = 0$ on the surface cells we have $\mathbf{T} = \frac{1}{Re}(\nabla\mathbf{u} + \nabla\mathbf{u}^t)$

5 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 mold filling simulation.

5.1 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 figure 2.



Figure 2: *Hagen-Poiseuille flow.*

The analytical solution for Hagen-Poiseuille flow, which can be found in Batchelor,¹² is given by:

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

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:

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

where Q is defined by:

$$Q = \int_0^L u(y) dy. \quad (25)$$

Considering $u(y) = U$ on the inflow, where U is the reference velocity, and choosing $L = U = 1$, the analytical solution is:

$$u(y) = -6y(y - 1), \quad (26)$$

Three different meshes have been employed to show the convergence of our method: a course mesh with 193 cells, an intermediate mesh containing 728 cells, and a refined mesh with 2853 cells. 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: $Fr = 0.319275$. Figure 3 shows the intermediate mesh and figure 4 presents a qualitative map of the velocity in x .

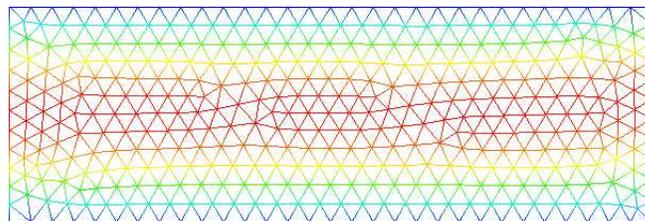


Figure 3: Intermediate mesh with 728 cells.

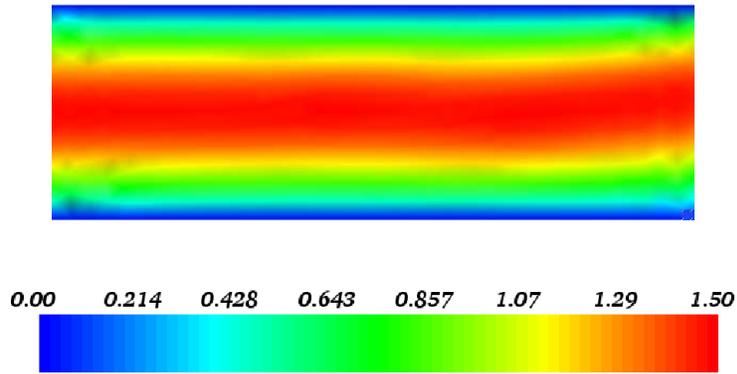


Figure 4: Velocity field in x direction.

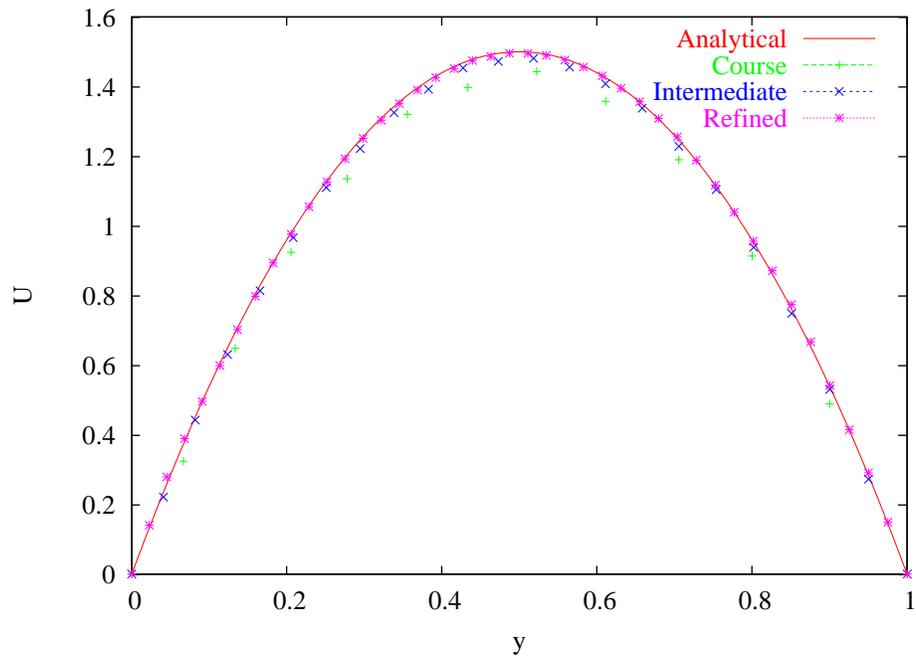


Figure 5: Comparing analytical and numerical results.

Figure 5 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.

5.2 Mold Filling

We finish this section with an example illustrating the behavior of the method when a free surface boundary condition is present. In this simulation a free slip boundary condition has been imposed on the rigid contours. A linear profile has been adopted on the three inflows, which have been defined on the right-most and left-most vertical lines and also on the horizontal bottom line.

Figure 6a), 6b), and 6c) show the velocity field in the x and y directions at three different times respectively. The colors from blue to red represent the velocities from $-10m/s$ to $10m/s$. The bounding box of the domain is a rectangle with base $11m$ and height $7m$.

Figure 7 illustrates the free surface propagation at the same times as in figure 6.

Notice from figure 7 that the free surface propagation is in accordance with what we expected.

6 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 with free surface, 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 with free surface 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# 03/02815-0), and CNPq, the Brazilian National Research Council (Grants # 300531/99-0).

REFERENCES

- [1] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, **11**, 83–95 (1980).
- [2] J.J. Monaghan. An introduction to sph. *Computer Physics Communications*, **48**, 89–96 (1988).

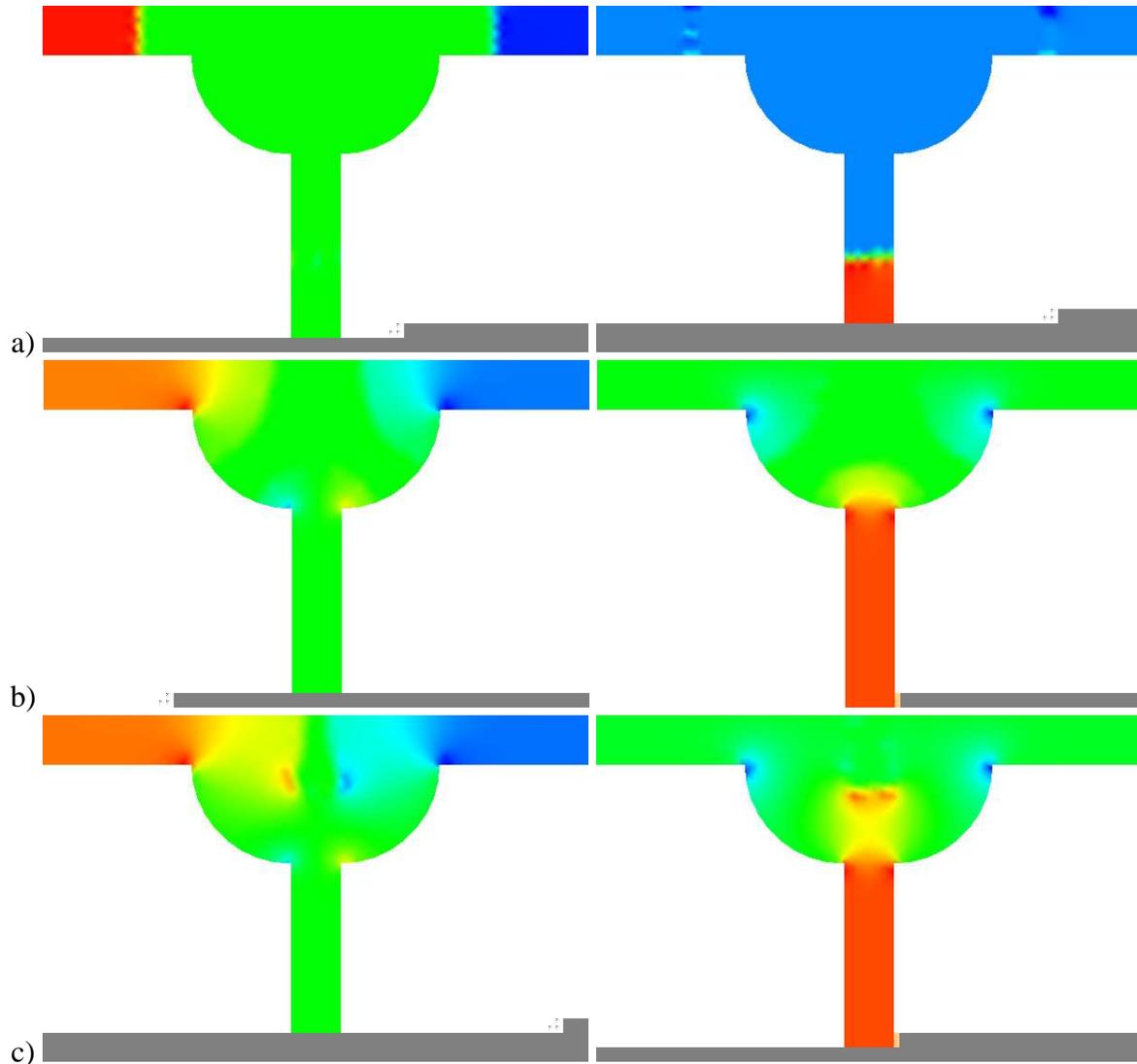


Figure 6: Mold Filling: Velocity in the x and y directions at: a) $t = 0.186339s$, b) $t = 0.555319s$, c) $t = 0.737863s$

- [3] T. Belytschko. Element-free galerkin method. *Int. J. Numer. Meth. Engrg.*, **37**, 229–256 (1994).
- [4] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, **10**, 307–318 (1992).
- [5] W.K Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *Int. J. Numer. Meth. Engrg.*, **38**, 1655–1679 (1995).
- [6] J.M. Melenk and I. Babuska. The partition of unity finite element method. *Comp. Meth. Appl. Mech. Engrg.*, **139**, 289–314 (1996).
- [7] S. Li and W.K. Liu. *Meshfree Particle Methods*. Springer, (2004).

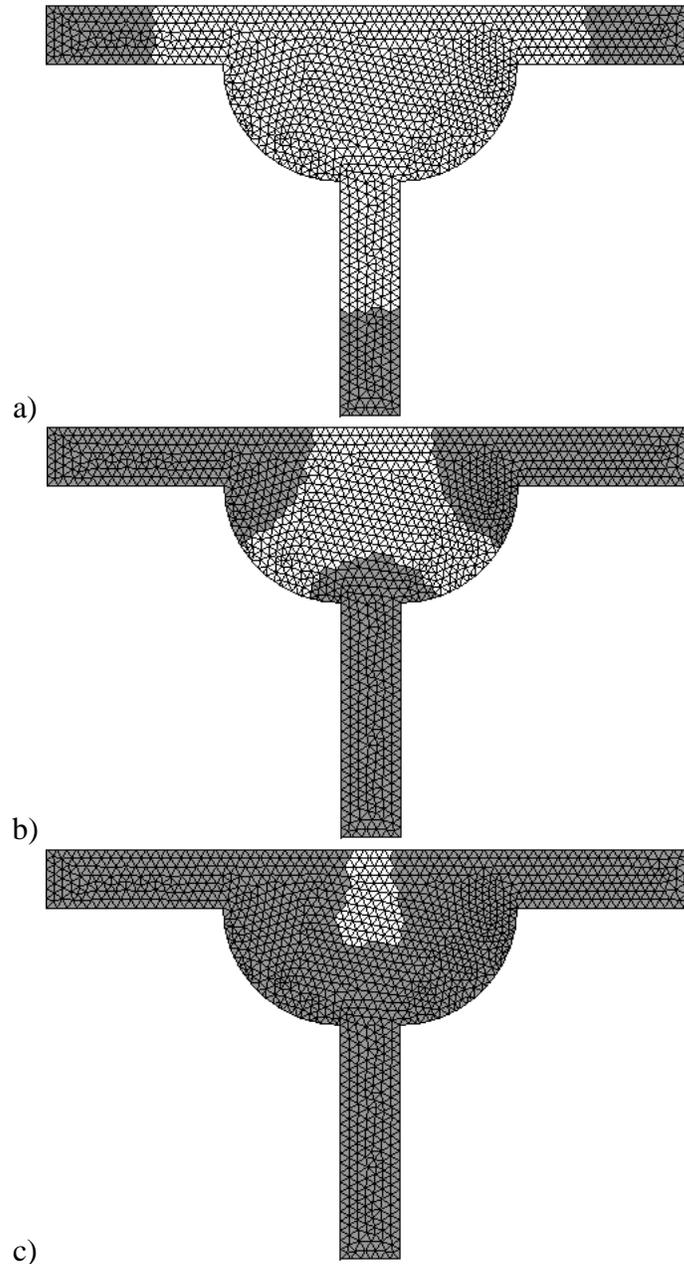


Figure 7: Free Surface Propagation at: a) $t = 0.186339s$, b) $t = 0.555319s$, c) $t = 0.737863s$

- [8] D. Levin. The approximation power of moving least-squares. *Math. Comp.*, **67**(224) (1998).
- [9] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**(201) (1981).
- [10] D.R.I Pe a. Método de diferenças finitas generalizadas por mínimos quadrados. Master

dissertation, ICMC-USP, (2003).

- [11] Y. Saad. *Iterative methods for sparse linear systems*. Philadelphia : SIAM, (2003).
- [12] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, (1970).