PARALLEL COMPUTING SIMULATION OF MULTIPHASE FLOWS EMPLOYING THE VOF METHOD

Karime Louise Zenedin Glitz, karime@sinmec.ufsc.br
António Fábio Carvalho da Silva, afabio@sinmec.ufsc.br
Carlos Newmar Donatti, newmar@sinmec.ufsc.br
Clovis Raimundo Maliska, maliska@sinmec.ufsc.br
SINMEC – Computational Fluid Dynamics Lab. Mechanical Engineering Department, UFSC. CEP: 88040-970, Florianópolis-SC, Brazil. Phone: +55 48 3721 9562

Abstract. *In the last years numerical simulation became a very important tool in solving engineering problems, due to its versatility in dealing with problems of different engineering's areas. However, as problems grow in complexity, such a technique is limited by the current capability of the single-processor computers. This reason and the development of cluster of computers lead to a wide use of parallel computing for solving flow problems in very large domains or in detailed scales. The parallelization of a computer code requires, however, modifications in the serial algorithm in order to deal with the exchange of data among the processors, and such changes depend on the method used to solve the flow. In this paper a parallel algorithm employed to solve transient multiphase flows using the Volume-of-Fluid method (VOF) with Piecewise-Linear Interface Calculation (PLIC) reconstruction technique is presented. The solution of the Navier-Stokes equations is obtained by using the PRessure Implicit Momentum Explicit (PRIME) method to treat the pressure-velocity coupling in a staggered arrangement of variables, while the parallelization is carried out using the Multiblock method and the MPI library. Three two-dimensional tests are carried out in order to assess the performance of the new algorithm: the broken dam with and without an obstacle and the rising bubble in a resting fluid. The objective of the first case is to test the capability of the parallel algorithm to propagate the pressure gradient to all sub-domains when the water arm, which is formed after water flows against the obstacle, reaches the opposite wall of the domain. Whilst the second test aims in assessing the computational time spent – as well as the speed-up factor – as a function of the number of processors employed when this algorithm is used. This is also the objective of the third test, at which the effects due to the interfacial tension are considered. All the numerical simulations were carried out by the supercomputer SGI Altix ICE 8200 of the Computational Fluid Dynamics Lab (SINMEC), at the Federal University of Santa Catarina (UFSC). As shown by the results, the parallel algorithm proposed in this work is able to deal with complex flows such as the broken dam with an obstacle. It can also be verified that the employment of the proposed parallelization technique results in a significant economy of CPU time when more than one processor is used. As can be verified through the results, when the computational domain is divided into 16 sub-domains, the simulation is four times faster than the serial one for the rising bubble case. The results presented in this paper confirm that the proposed algorithm represents an alternative to expensive serial simulations of transient multiphase flows employing the VOF method.*

Keywords*: parallel computing, multiphase flow, VOF, Multiblock*

1. INTRODUCTION

In the last years numerical simulation became a very important tool in solving engineering problems, due to its versatility in dealing with problems of different engineering's areas. However, as problem grows in complexity, such a technique is limited by the current capability of the single-processor computers. This reason and the development of cluster of computers lead to a wide use of parallel computing for solving flow problems in very large domains or in detailed scales.

The parallelization of a problem means to divide it into several sub-problems and to solve each of them in separated processors. Each process runs simultaneously to the others and some information, which is employed as boundary condition, is exchange among the processors. This is done in order to achieve the solution in less CPU time, that is, than that required in a serial procedure.

There are several ways to parallelize a computer code. For instance, it can be done by parallelizing only the linear system solver or by subdividing the solution domain into sub-domains. Such strategy is called domain decomposition and it can be done in time or in space. Another possibility of parallelization is to employ the Multiblock method. This method is similar do the domain decomposition in space, however, in the Multiblock method, the geometry characteristics control the subdivision of the domain (Ferziger and Peric, 2002).

In this paper a parallel algorithm employed to solve transient multiphase flows using the Volume-of-Fluid (VOF) method (Hirt and Nichols, 1981) with Piecewise-Linear Interface Calculation (PLIC) reconstruction technique is presented (Malik and Bussmann, 2004; Kothe *et al.*, 1996). The solution of the Navier-Stokes equations is obtained by using the PRessure Implicit Momentum Explicit (PRIME) method to treat the pressure-velocity coupling in a staggered

arrangement of variables (Maliska, 2004), while the parallelization is carried out using the Multiblock method and the MPI library.

Three two-dimensional tests are carried out in order to assess the performance of the new algorithm: the broken dam with and without an obstacle and the rising bubble in a resting fluid. The objective of the first case is to test the capability of the parallel algorithm to propagate the pressure gradient to all sub-domains when the water arm, which is formed after water flows against the obstacle, reaches the opposite wall of the domain.

Whilst the second test aims in assessing the computational time spent – as well as the speed-up factor – as a function of the number of processors employed when this algorithm is used. This is also the objective of the third test, when the effects due to the interfacial tension are then considered.

2. NUMERICAL MODEL

The numerical simulations carried out considered two-dimensional transient flow of two fluids with different physical properties. Both the fluids are considered incompressible.

As the interface between the fluids is well-defined, the Navier-Stokes (NS) equations are solved for each phase. Since the fluids are different, the physical properties of the mixture (named the viscosity and the density of the fluid) are considered in the NS equations when assessing the flow in the interface region. Thus the governing equations of this flow are

$$\frac{\partial f_k}{\partial t} + \nabla \cdot \left(f_k \mathbf{u} \right) = 0 \tag{1}$$

which is the equation of continuity applied to both fluids,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \mathbf{u} \right) = 0 \tag{2}$$

which is the equation of continuity for the mixture, and

$$\frac{\partial \left(\rho \mathbf{u} \right)}{\partial t} + \nabla \cdot \left(\rho \mathbf{u} \mathbf{u} \right) = -\nabla p + \nabla \cdot \left\{ \mu \left[\nabla \mathbf{u} + \left(\nabla \mathbf{u} \right)^T \right] \right\} + \rho \mathbf{b} + \mathbf{F}_{sv} \tag{3}$$

which is the momentum equation.

In these equations f_k denotes the volume fraction of the fluid k, $\mathbf{u}$ denotes the velocity vector $\mathbf{u} = (u,v)$, t represents time and p denotes pressure. The last two terms in the right-hand side of Eq. (3) correspond to the body forces and to the force due to the interfacial tension, respectively. The density and the dynamic viscosity of the mixture are denoted by ρ and μ, respectively. These physical properties are estimated by averaging each fluid property weighted by its volume fraction. Therefore for a two-phase flow:

$$\rho = f_1 \rho_1 + f_2 \rho_2 \tag{4}$$

$$\mu = f_1 \mu_1 + f_2 \mu_2 \tag{5}$$

where the subindexes denote the fluid.

In order to numerically solve the flow, i.e., to determine the fields of u, v, p, ρ and μ, the Finite Volume method (FVM) was applied to the governing equations, by dividing the domain into discrete volumes and integrating these equations over these volumes and over time intervals. The PRessure Implicit Momentum Explicit (PRIME) method was used to treat the pressure-velocity coupling problem.

2.1. Interface tracking

Since a two-phase flow is simulated, an important task is to determine the position of the interface between the fluids. This is done by estimating the field of volume fractions through the solution of Eq. (1), which involves the assessment of the fluxes of f advected through the boundaries of the control volume. Among the several possibilities to evaluate these fluxes it can be mentioned the employment of an advection scheme or the application of an interface tracking technique such as the Volume-of-fluid (VOF) method, applied in this work. The main drawback of the advection scheme is that it provides a very diffuse interface, whose exact position cannot be determined. On the other hand the interface assessed by the VOF method is well-defined, when a good grid resolution is used, since a geometrical scheme is used to evaluate the advection fluxes of f. In this work this evaluation was carried out by employing the

Piecewise-Linear Interface Calculation (PLIC) technique, which approximates the interface by linear oriented-segments.

2.2. Interfacial tension effects

When a volume method – like the VOF method – is employed to track the position of the interface, the treatment of the interfacial tension effects becomes a complex task, as these effects must be considered only at such a narrow region as is the interface between the fluids.

In order to deal with this problem Brackbill and coworkers (1992) developed the Continuum Surface Force (CSF) model. In this model the force due to interfacial tension is included in the Navier-Stokes equations as a body force ($\mathbf{F}_{sv}$), which acts in a limited region around the interface. This force is a product among the interfacial tension coefficient (σ), the curvature of the interface (κ), the unit vector normal to the interface ($\hat{\mathbf{n}}$) and the Dirac Delta function (δ), given by:

$$\mathbf{F}_{sv} = \sigma\kappa\hat{\mathbf{n}}\delta \tag{6}$$

As the vector normal to the interface is defined as the gradient of the volume fraction, it was evaluated by employing a finite difference approach in a 3x3-volumes stencil. The Dirac delta function was taken to be equal to $|\nabla f_1|$. Therefore this force is estimated only at the volumes in the region around the interface and vanishes in the remaining volumes.

In order to estimate the curvature of the interface the Height Function (HF) method was applied (Francois *et al.*, 2006). According to this method, for curvature estimation's purposes only, the interface is approximated by a Height Function. And since the curvature is defined as the divergence of the unit vector normal to the interface, the curvature is expressed as

$$\kappa = -\frac{H_{xx}}{\left(1 + H_x^2\right)^{3/2}} \tag{7}$$

where H is the height function, and the subindexes denote differentiation.

2.3. Parallelization

In this work the parallelization is done by applying the Multiblock method. In this method the computational domain is divided into sub-domains (called partitions) and each of these partitions is assigned to a different processor. The solution of the flow is obtained simultaneously in each partition and since the flow in a sub-domain depends on the flow in the other ones, the values of the variables u, v and p are exchanged among them and act as boundary conditions in the neighbour partition.

Therefore, for a domain which was divided into two partitions, as illustrated in Fig. 1, the values of u_W, v_W, and p_W calculated in the partition 1 are informed to partition 2 and act as a Dirichlet boundary condition in this sub-domain. On the other side the values of u_E, v_E, and p_E estimated by partition 2 are communicated to partition 1, working as boundary conditions in this sub-domain.

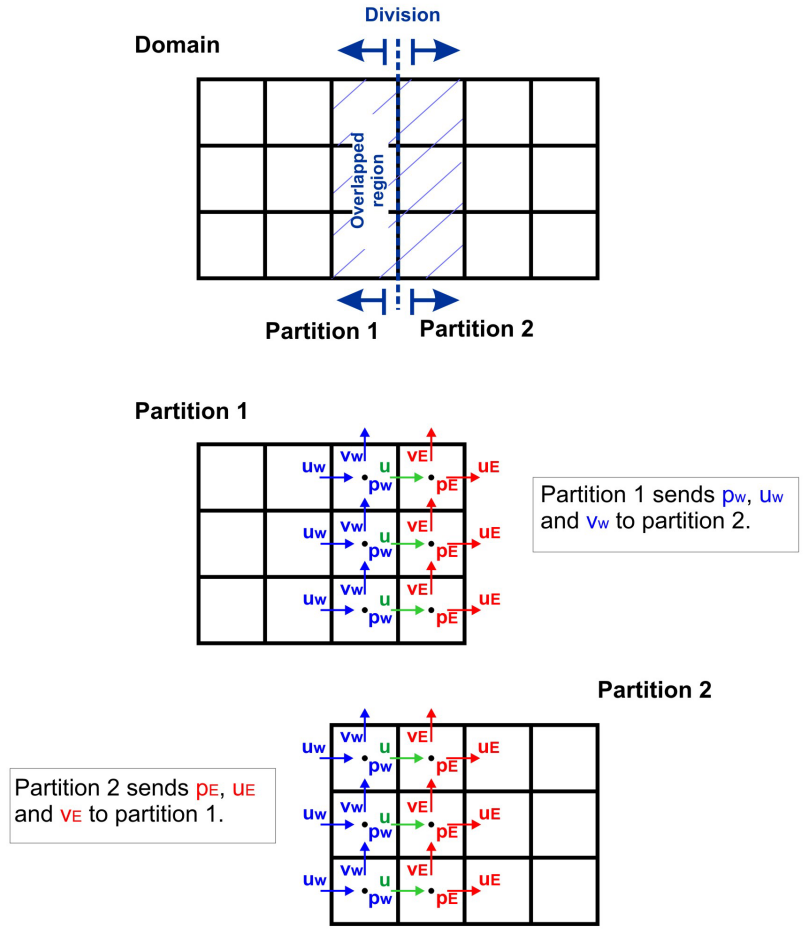


Figure 1. Example of the partition of a domain and data exchanged between the partitions.

As can be seen is this figure there is a hatched area called *Overlapped Region*, where the boundary conditions of both partitions lie. The size of this area determines the amount of information exchanged by the sub-domains. In this work it was employed the minimal possible size of the overlapped region, that is two rows/columns. It was also decided to apply the parallelization process only to the solution of the fields of u, v and p. Since there is not a substantial reduction of the computational costs when the determination of the field of f is parallelized, this task is done in serial by the master processor.

The algorithm of parallelized solution of the two-phase flow employing the PRIME method is shown in Fig. 2.

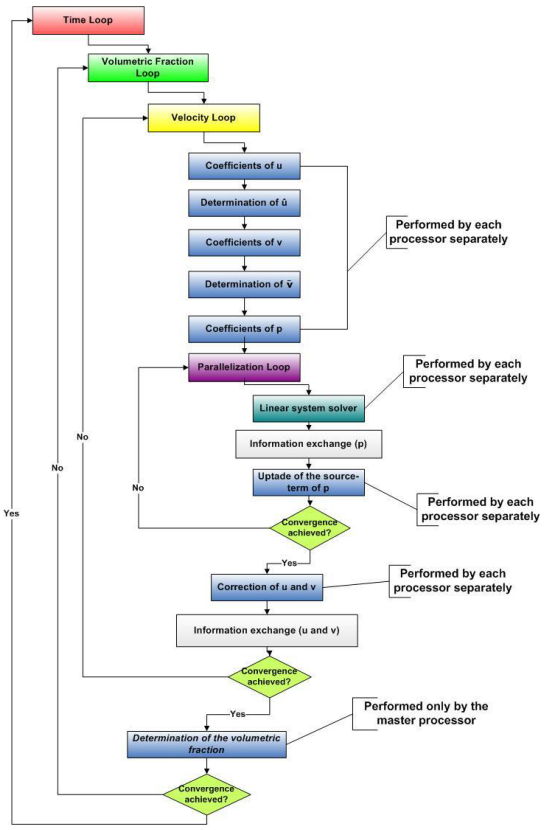
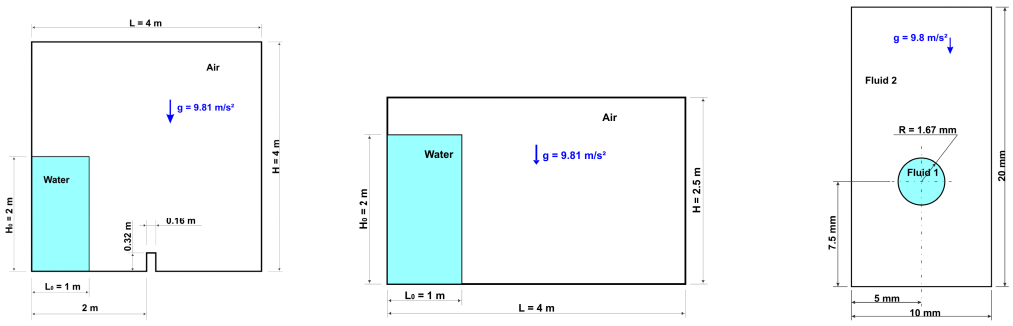


Figure 2. Algorithm for parallel simulations employing PRIME.

All the numerical simulations were carried out by the supercomputer SGI Altix ICE 8200 of the Computational Fluid Dynamics Lab (SINMEC), at the Federal University of Santa Catarina (UFSC). They were also performed by employing the MPI library (ANL, 2010; Barney, 2010).

3. TESTS

In order to analyze the performance of the parallel algorithm and to estimate the reduction in CPU time when a parallel simulation is carried out, three two-dimensional tests were performed: the broken dam case with and without an obstacle, and the rising bubble in a resting fluid. All the three cases are illustrated in Fig. 3.



(a) (b) (c)

Figure 3. (a) Broken dam with obstacle, (b) Broken dam without obstacle, and (c) Rising bubble.

Cartesian meshes were employed in all the numerical simulations.

3.1. Broken Dam with an obstacle

The objective of the this case is to test the capability of the parallel algorithm to propagate the pressure gradient to all sub-domains when the water arm, which is formed after water flows against the obstacle, reaches the opposite wall of the domain. Thus a coarse grid was employed.

Table 1 shows the values of the parameters employed in the numerical simulations. The interfacial tension effects were neglected in this case as well as in the simulations of the broken dam without an obstacle.

Table 1. Values of the parameters (broken dam with an obstacle).

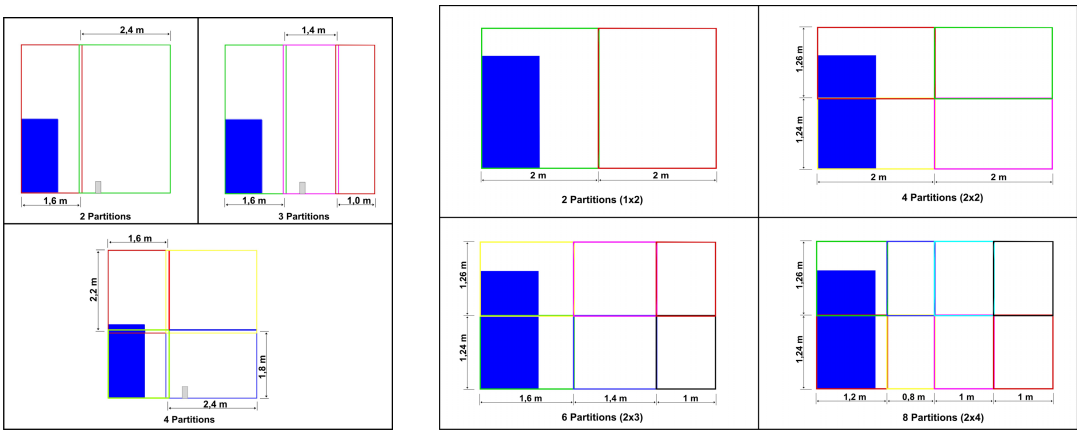
Water density (kg/m³)	1000
Water dynamic viscosity (Pa s)	8.55E-4
Air density (kg/m³)	1.16
Air dynamic viscosity (Pa s)	1.85E-5
Time step (s)	1E-4
Final simulation time (s)	1.25
Convergence criterium for p (Pa)	1
Convergence criterium for u and v (m/s)	1E-3
Convergence criterium for f	1E-5
Boundary Conditions	Impermeable and no-slip walls
Grid size	100 x 100 volumes
Number of partitions(1)	2, 3 and 4

(1) : shown in Fig. 4.

3.2. Broken Dam without an obstacle

The second test aims in assessing the computational time spent – as well as the speed-up factor – as a function of the number of processors employed when the algorithm illustrated in Fig. 2 is used. In this test with exception of the final simulation time, the grid size and the number of partitions, all the remaining parameters are equal to those of the previous case (Tab. 1).

All the simulations were performed up to 0.7 s in a 800 x 500 volumes grid. For the parallel simulations the domain was divided into 2, 4, 6 and 8 partitions, which are shown in Fig. 4. Besides the partitions for this test, Fig. 4 also shows the division of the domain for the previous case.



(a) (b)

Figure 4. Partitions of the broken dam (a) with an obstacle, and (b) without an obstacle.

3.3. Rising bubble in a resting fluid

As the previous case the objective of the third test is to estimate the CPU time and the speed-up factor of parallel simulations, in this case considering the interfacial tension effects.

In this test a bubble of a fluid 1 immersed in a resting fluid 2 raises toward the top of the domain. The geometrical and physical properties (Tab. 2) of the problem are such that the Eötvös and Morton numbers are respectively equal to 104 and 0.1.

Table 2. Values of the parameters (rising bubble).

Fluid 1 density (kg/m³)	25
Fluid 1 dynamic viscosity (Pa s)	1.22E-4
Fluid 2 density (kg/m³)	1000
Fluid 2 dynamic viscosity (Pa s)	1.04E-2
Interfacial tension coefficient (N/m)	1.047E-3
Time step (s)	5E-5
Final simulation time (s)	0.09
Convergence criterium for p (Pa)	1E-3
Convergence criterium for u and v (m/s)	1E-5
Convergence criterium for *f*	1E-5
Boundary Conditions	Impermeable and free-slip walls
Grid size	256 x 512 volumes
Number of partitions(2)	2, 4, 8, 16 and 32

(2) : The sub-domains have the same size in each simulation, and the partitions are illustrated in Fig. 5.

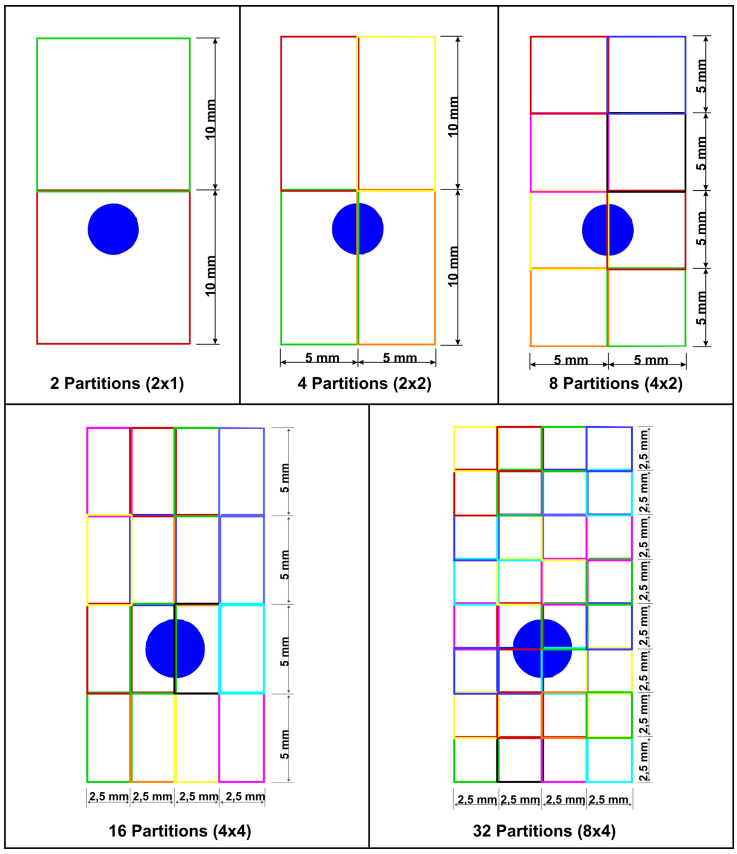


Figure 5. Partitions of the rising bubble problem.

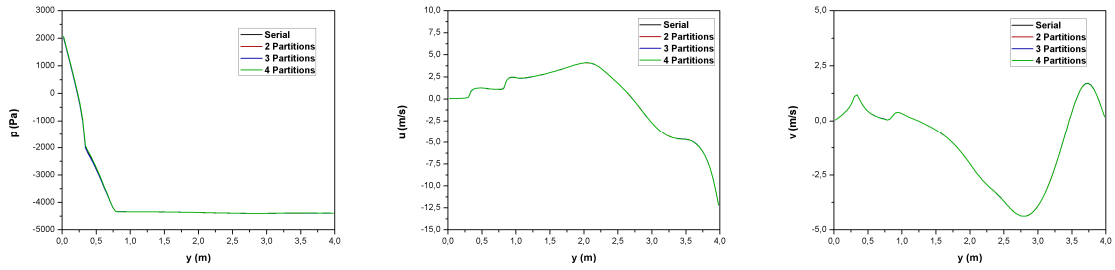
4. RESULTS

Before carrying out the parallel simulations, all the serial results obtained for the three cases mentioned in the previous section were validated with those published in the literature (Muzaferija and Peric, 1998; Anagnostopoulos and Bergeles, 1999; Ginzburg and Wittum, 2001).

4.1. Broken Dam with an obstacle

As the aim of this test is to validate the parallel algorithm the results obtained in the simulations employing 2, 3 and 4 sub-domains were compared to those of the serial simulation in three distinct instants: the first one before the water arm reaches the opposite wall (at 0.85 s), during the impact (at 0.92 s) and at the end of the simulation (at 1.25 s). At each of these instants the behaviour of the two components of **u** and of the pressure along a line of constant value of *x* or *y*, as well as the profiles of *f*, were compared.

The comparison among the serial and the parallel results shows excellent agreement. As can be seen in Fig. 6 for t=1.25 s and along a line of constant value of x (1.98 m), the curves of u, v and p obtained in the parallel simulations are indistinguishable from the serial ones.



(a) (b) (c)

Figure 6. Curves of (a) pressure, (b) horizontal component, and (c) vertical component of the velocity at 1.25 s and x=1.98 m *versus* y.

The volume fractions field obtained in the serial and in the 4-partitions simulations at this same instant are shown in Fig. 7, where the blue color indicates water (f=1). As can be observed there is no difference between them. This validates the algorithm conception and the implementing procedures.

(a) (b)

Figure 7. Volume fraction field at 1.25 s: (a) serial, and (b) parallel simulation with 4 partitions.

4.2. Broken Dam without an obstacle

In this case the behaviour of the CPU time spent – as well as of the speed-up factor – as the number of employed processors increases is assessed.

The speed-up factor is a parameter usually employed to analyze the performance of parallel simulations. It is expressed as

$$S_n = T_s / T_n \tag{8}$$

where T_s and T_n denote respectively the CPU time spent by the serial simulation and by a parallel simulation using n processors.

As the number of processors used in the simulations increases, it is expected that the expense of computational time decreases. On the other hand the speed-up factor should increase.

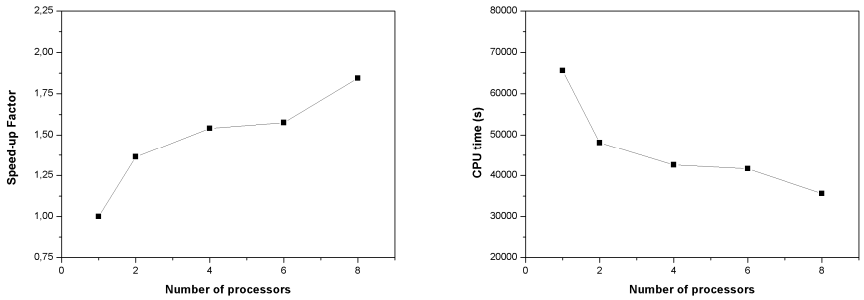
However there is a limit beyond which an increase in the number of partitions will not imply an economy of computational time, because the number of volumes in the overlapped regions will also increase and more time will be spent by the processors in exchanging data. In the broken dam test this boundary was not achieved as can be concluded from the results for the CPU time and speed-up factor shown in Tab. 3.

Table 3. CPU time and speed-up factor (broken dam without an obstacle).

Number of processors	CPU time (s)	Speed-up factor
1	65613.26	1
2	48112.02	1.36
4	42599.60	1.54
6	41640.81	1.57
8	35577.68	1.84

According to the values plotted in Tab. 3 it can be concluded that there was no expressive gain in CPU time when the calculation was shared by two, four or six processors. But when eight sub-domains were employed the CPU time dropped almost to a half of that spent in the serial simulation, resulting in a speed-up factor equal to 1.84.

As illustrated in Fig. 8 the speed-up factor presents an increasing tendency while the CPU time presents a decreasing tendency – both curves show exponential behaviour when only the four first data are considered – as more processors are employed.



(a) (b)

Figure 8. (a) Speed-up factor and, (b) CPU time for the simulations of the broken dam without an obstacle case.

4.3. Rising bubble in a resting fluid

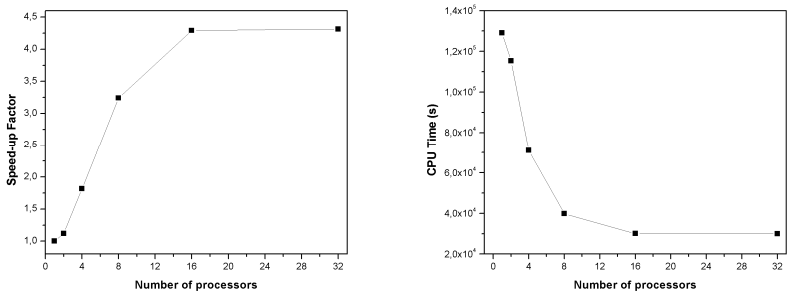
As in the previous case, in this test the total CPU time and the speed-up factor were estimated. However, in this case the effects of interfacial tension are considered, which means that some time will be spent in evaluating the curvature of the interface as well as more time will be spent in solving the flow since the force due to interfacial tension is treated explicitly.

Although the presence of the interfacial tension represents an increase in the computational time spent by the master processor – as this task is done by only one processor –, its consideration did not represent a significant loss in the performance of the parallel simulations. On the contrary: in this case an expressive economy in CPU time was achieved as more processors were used, as can be seen in Tab. 4: the numerical simulation performed by 16 processors spent approximately 23% of the time consumed by the serial simulation.

Table 4. CPU time and speed-up factor (rising bubble).

Number of processors	CPU time (s)	Speed-up factor
1	129055.13	1
2	115282.75	1.12
4	70993.52	1.82
8	39824.31	3.24
16	30075.52	4.29
32	29927.86	4.31

As clearly illustrated in Fig. 9 the computational time curve as well the one of the speed-up factor exhibit an asymptotical behaviour as the number of processors grows from 16 to 32. As previously mentioned, this behaviour indicates that an increase in the number of partitions will not enhance the performance of the simulations in terms of computational costs, since the balance between the computational time spent in the solution of the flow and that expended in exchanging data – and also the processors' idle time – was reached.



(a) (b)

Figure 9. (a) Speed-up factor and, (b) CPU time for the simulations of the rising bubble case.

5. CONCLUSIONS

As shown by the results presented in this paper, the parallel algorithm proposed in this work is able to deal with complex flows such as the broken dam with an obstacle. It can also be verified that the employment of the proposed parallelization technique results in a significant economy of CPU time when more than one processor is used, as illustrated in Fig. 9 for the rising bubble test: as the computational domain is divided into 16 sub-domains, the simulation is four times faster than the serial one.

The results presented in this work confirm that the proposed algorithm represents an alternative to expensive serial simulations of transient multiphase flows employing the VOF method.

6. ACKNOWLEDGEMENTS

The first author would like to express her gratitude to the National Council for Scientific and Technological Development (CNPq) for sponsoring her doctorate study.

7. REFERENCES

Anagnostopoulos, J. and Bergeles, G., 1999, "Three-dimensional Modeling of the Flow and the Interface Surface in a Continuous Casting Mold Model", Metallurgical and Materials Transactions B, Vol. 30, No. 6, pp. 1095-1105.

ANL – Argonne National Laboratory, 2010, MPICH2. 27 April 2010 <http://www.mcs.anl.gov/research/projects/mpich2>.

Barney, B., 2010, Message Passing Interface (MPI). 27 April 2010 <http://computing.llnl.gov/tutorials/mpi/>.

Brackbill, J., Kothe, D. and Zemach, C., 1992, "A continuum method for modeling surface tension", Journal of Computational Physics, Vol. 100, No. 2, pp. 335-354.

Ferziger, J.H. and Peric, M., 2002, "Computational Methods for Fluid Dynamics", Springer-Verlag, New York, USA, 423 p.

Francois, M., Cummins, S., Dendy, E., Kothe, D., Sicilian, J. and Williams, M., 2006, "A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework", Journal of Computational Physics, Vol. 213, No. 1, pp.141-173.

Ginzburg, I. and Wittum, G., 2001, "Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants", Journal of Computational Physics, Vol. 166, No. 2, pp. 302-335.

Hirt, C.W. and Nichols, B.D., 1981, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries", Journal of Computational Physics, Vol. 39, No. 1, pp. 201-225.

Kothe, D.B., Rider, W.J., Mosso, S.J., Brock, J.S. and Hochstein, J.I., 1996, "Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions", Aerospace Sciences Meeting and Exhibit, 34th, Reno: AIAA 96-0859.

Malik, M. and Bussmann, M., 2004, "Volume Tracking on Adaptively Refined Grids with Curvature Based Refiement", Proceedins of CSME Forum, London, UK, pp. 1-10.

Maliska, C.R., 2004, "Transferência de calor e mecânica dos fluidos computacional", LTC – Livros Técnicos e Científicos Editora S. A., Rio de Janeiro, Brazil, 453 p.

Muzaferija, S. and Peric, M., 1998, "Computation of free-surface flows using interface-tracking and interface-capturing methods", In: Nonlinear Water Wave Interaction, Computational Mechanics Publications, Southampton.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.