Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIII, págs. 27-49 (artículo completo)
Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)
San Carlos de Bariloche, 23-26 Setiembre 2014

ENGINE SIMULATION USING LAYERING AND SLIDING INTERFACES MESH DYNAMICS TECHNOLOGIES

Horacio J. Aguerre[a,b], Santiago Márquez Damián[a], Juan M. Gimenez[a] and Norberto M.Nigro[a]

[a]*Research Center for Computational Mechanics, CIMEC-UNL/CONICET, Güemes 3450, Santa Fé, Argentina, aguerrehoracio@gmail.com*

[b]*FRCU, Universidad Tecnológica Nacional, Ing. Pereyra 676, 3260 C. del Uruguay (ER), Argentina*

Keywords: OpenFOAM, Opposite Piston Engine, Mesh Dynamics

Abstract. The optimization of internal combustion engines is an important strategy to reduce the global energy requirements and pollutant emissions. The design of new engines needs to use efficient engineering tools to reduce development time and economic resources. Regarding to this, computational fluid dynamics in general and the OpenFOAM® suite in particular have shown to be adequate tools for the development of engines with the advantages for the latter of being a free code with a large users community.

Typical engine simulation cases involve multiple fluid regions with relative motion. The engine kinematics requires a mesh adaptation in order to preserve the connection between the subdomain interfaces and to manage large mesh deformations. In this context, this work presents an extension of the OpenFOAM®'s dynamic mesh library which performs automatic mesh topological actions for engine problems. The implementation combines the mesh topological modifiers like *layering* for the treatment of mesh large deformations and *sliding interfaces* to accomplish the connection of fixed and moving subdomains. Further, a set of new features are added on the layering mesh modifier to allow the configuration of multiple motion regions and the use of different mesh size zones.

The library is used within a compressible flow solver for the resolution of a real problem. A new design of an opposite piston engine is tested and the performance of the in-house code is evaluated.

1 INTRODUCTION

Computational fluid dynamics (CFD) is an efficient tool for optimizing and redesign internal combustion engines. The increasing demand of even more efficient engines enforces new and more reliable numerical models for CFD codes. In this context, this work is within a project that pretend to contribute in a CFD study that evaluates a new design of an opposite piston engine with the aim of increasing its performance and energy efficiency.

Basically, the engine geometry can be divided in three different regions: the cylinder, the exhaust ports and the intake ports. The air mass gets into the cylinder through the intake ports and exits across the exhaust ports depending on the engine cycle phase. The physical communication between the cylinder and the ports and thus the inlet and outlet of cylinder air, is managed by the proper engine mechanism. In four strokes engines, the valves manage the entrance and the exit of air in the cylinder. On the other hand, in two strokes engines, the pistons themselves are the elements that execute the gas exchange control.

The opposite piston engine studied in this work is a two stroke engine which has two pistons per cylinder. In order to perform the engine simulation, the computational domain must be divided in three regions to represent the cylinder part and the intake and exhaust ports. The cylinder is a moving zone which has large deformations, as the maximum cylinder volume is approximately 10 times larger than the minimum volume. To accomplish the large deformation, an adequate mesh dynamic strategy should be adopted with the purpose of maintain a good mesh quality for all cylinder sizes. Additionally, the region connectivity varies depending on the piston positions. Therefore a region coupling and decoupling method is needed to link the separated regions.

This work uses the OpenFOAM®official version as computational code. The software is composed by a set of free object oriented C++ libraries for continuum mechanics computations. OpenFOAM®
was presented initially in the work of Weller et al. (1998) and nowadays is the open source CFD code with the largest user community.

Main advances in mesh dynamics for engine problems using OpenFOAM®were developed by Lucchini et al. (2007) using the OpenFOAM®extended version. In this work, the *sliding mesh interfaces* method is proposed to address the region coupling and decoupling process and the *layering technique* is used to solve the problem of large domain deformations. Other examples of use of both mesh dynamics tools on two stroke engines can be found in the master thesis of Vissers (2007) and Johnsson (2010).

The present development includes an extension of the combined *sliding mesh interfaces* and *layering* mesh dynamic technologies which includes the possibility of define multiple zones motion and the use of different location based mesh size zones to reduce the numerical error produced by inappropriate domain discretization (Jasak, 1996).

The new dynamic mesh tool is used for the simulation of a new configuration of an opposite piston engine. The variable mesh size zone capability of the dynamic mesh method is exploited to have an enhanced mesh resolution on the combustion chamber. The flow patterns in this zone are evaluated and a preliminary estimation of the fuel-air mixture is performed by approximating the fuel dynamics with the scalar transport of a marker function.

2 COMBINED SLIDING INTERFACES AND LAYERING MESH DYNAMICS TECHNOLOGIES

2.1 Sliding interfaces

The sliding mesh interfaces are necessary when two or more mesh regions have common interfaces and a relative tangential movement between themselves. There are basically two alternatives in order to solve the numerical difficulty that present the separated regions. The first alternative is to treat the regions separately as individual problems and use an adequate numerical strategy to couple the interfaces values. An example of this approach is found in Nilsson et al. (2008). Furthermore the recently released OpenFOAM® version 2.3 has included the periodic Arbitrary Coupled Mesh Interfaces (ACMI) boundary condition type which enables the numerical coupling between region interfaces (Farrell and Maddison, 2011). The second option is to merge the regions into a single one using mesh dynamics technology. In this approach, the interfaces between regions disappear and they become in a singular set of internal faces on the interfaces overlapping region and become in boundary faces where the interfaces do not overlap. The latter has the advantage of reducing the numerical error and saving in computational cost, when the regions involved do not vary considerably their relative positions avoiding to perform the topological changes.

2.1.1 Procedure

The sliding interfaces procedure consists in coupling topologically two or more separated mesh regions. The coupling method uses a master-slave interfaces approach. The target of the topological mesh modifier is to create an unique interface combining the master and slave ones. The resulting interface must be conformal with the two sets of adjacent cells of the master and slave sides. In other words, the new interface must be the set of internal and boundary faces that link topologically both groups of adjacent cells. The number of adjacent cells of both sides remain unchanged, only faces and points are added or removed.

At the first mesh update process, the mesh is composed by two regions or more, which are separated. This means that each region is bounded by its own closed surface. This state is named *decoupled state* and is the initial required condition for this mesh dynamic method. When the mesh is decoupled, the moving mesh action can be executed individually for each region. In this step whichever mesh dynamic technology can be applied to perform the boundary motion. It is only required that the interfaces of the regions do not separate each other in their normal direction. Figure 1 explains the decoupled state with a simple case that is composed by two regions where each one of them has two cells. The region A owns the master interface and the region B has the slave interface. The interfaces should be superimposed on the figure, but a gap between them is included for clarity purposes. The figure shows also the mesh motion action which correspond to the compressing of the right region by the upward movement of its lower boundary face. This motion produces that the slave interface slides along the master one.

After completing the mesh motion step, the points of each region are located in their final position according to the desired displacement function and therefore the region interfaces are ready to be merged into one through the coupling action.

The coupling method is divided in the following steps. First, the slave interface points are projected in their normal direction onto the master interface. This action produces an *enriched interface* as it has its set of original points and additionally, the new set of projected slave interface points. After projecting the slave points, the distance between all slave and master

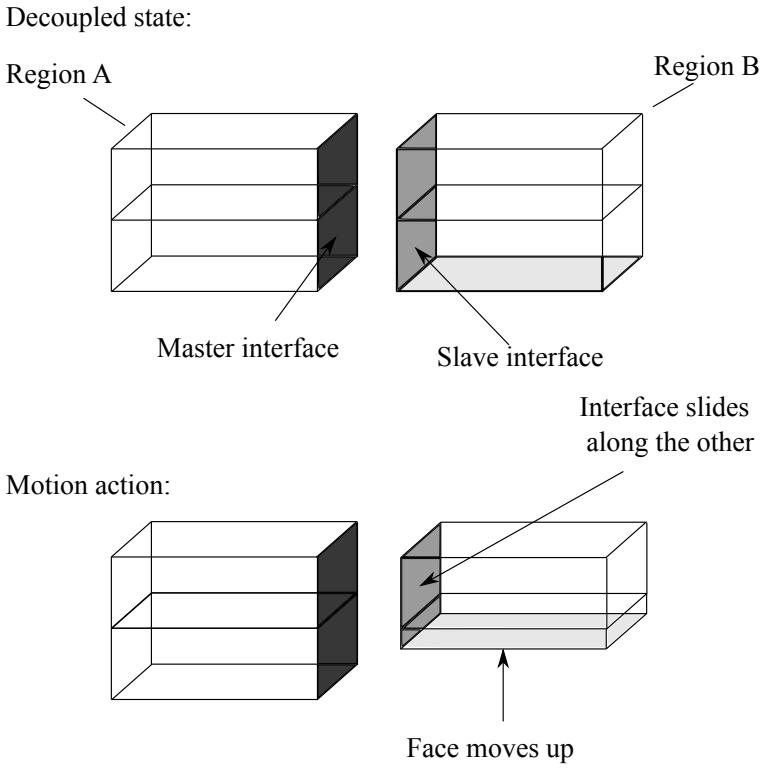

Figure 1: Decoupled regions and mesh motion action in an example case.

points is measured. If the distance between two points lies below a certain tolerance, the slave point is merged into the master point. Likewise, if a slave point projection lies near a master interface edge, the slave point is moved to the master edge on the enriched interface. These tolerances are set up as a fraction of the shortest master interface edge. The use of these merging tolerances avoid the generation of extreme small geometric entities, which can decrease the mesh quality.

The next action in the coupling procedure is to divide the *enriched interface* faces accordingly to both set of new points. In this topological modification new faces appear which can be either internal or boundary faces. Finally, the domains are joined with the new interface and the mesh is then composed by only one region. The mesh update procedure is finalized and the finite volume problem can be implemented on the new mesh configuration.

Figure 2 describes the actions explained in the last paragraphs on the proposed example case.

In the next time step, the mesh update procedure is performed newly. Unlike the first time step of the simulation, where the regions were separated as an imposed requisite for the method, the regions are at this instance coupled. Thus, they need to be decoupled in order to allow the mesh movement mechanism. The topological modifier gets back the decoupled state of the slave and master interfaces and the mesh recovers its separated state. In order to do this work, the data related to the mesh has to be saved before the mesh coupling. This information includes the points and faces that were retired from the mesh as a result of the previous coupling action. When the regions are decoupled, the mesh update procedure completes a cycle. The mesh movement mechanism can be executed and the regions are ready to be coupled again.

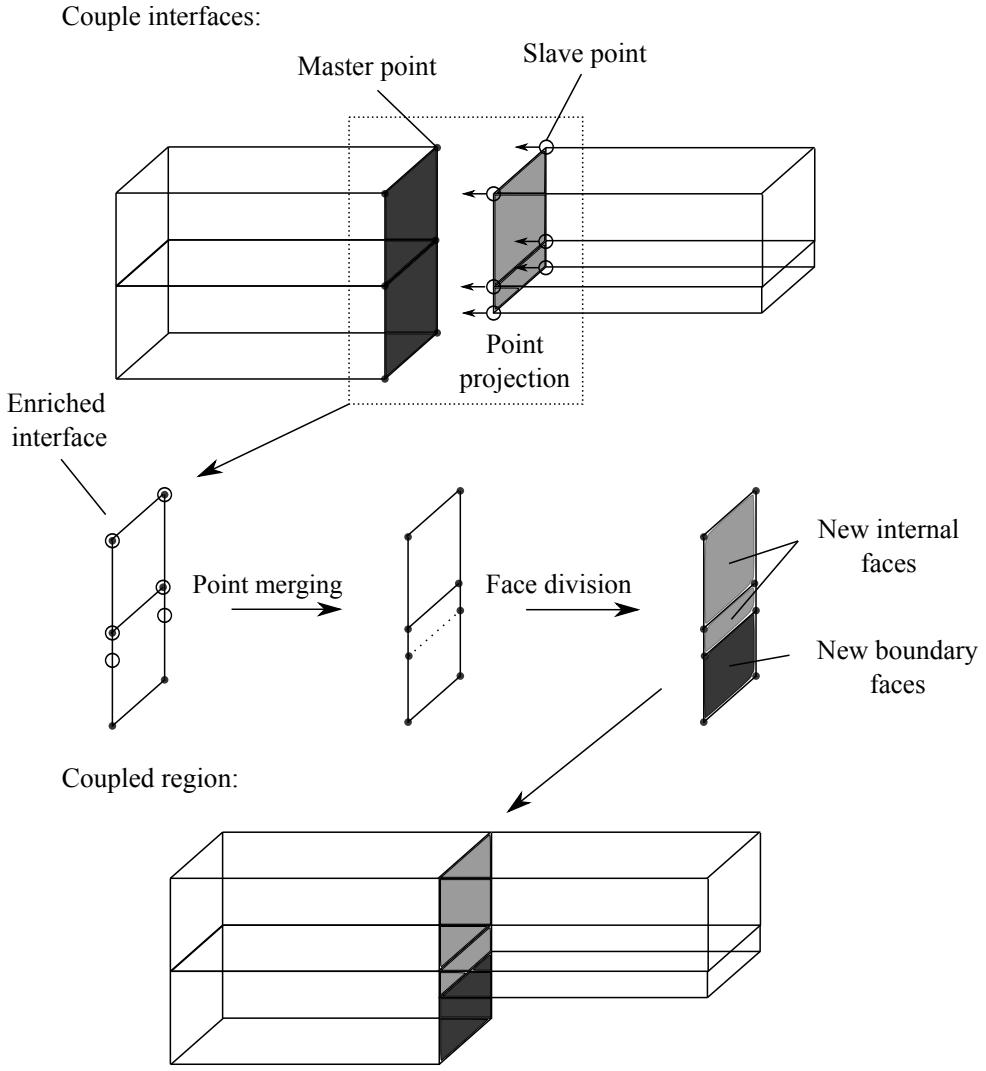

Figure 2: Different mesh actions for the coupling step.

2.2 Layering

The layering mesh dynamic technology is specially formulated to solve large domain deformations. The technique uses meshes that are extrusions of planar 2D mesh. The 3D mesh is finally formed by dividing the domain in the extrusion direction conforming a sequential set of layers. Due to the mesh boundary motion, the domain adds or removes layers depending on the motion direction and the mesh is adapted to the new boundary position. The layer height is defined by layer tolerances. A more detailed description of the layering procedure can be found in Aguerre et al. (2013).

2.2.1 Distinctive aspects between layering and mesh deformation methods

The layering technique has the characteristic that the mesh boundary motion does not affect the internal mesh. Topological actions are only performed on boundary adjacent cells. This feature avoids the inconvenient that is presented with the mesh deformation strategy. In the basic formulation of this method, a Laplacian equation is solved to determine the point displacements,

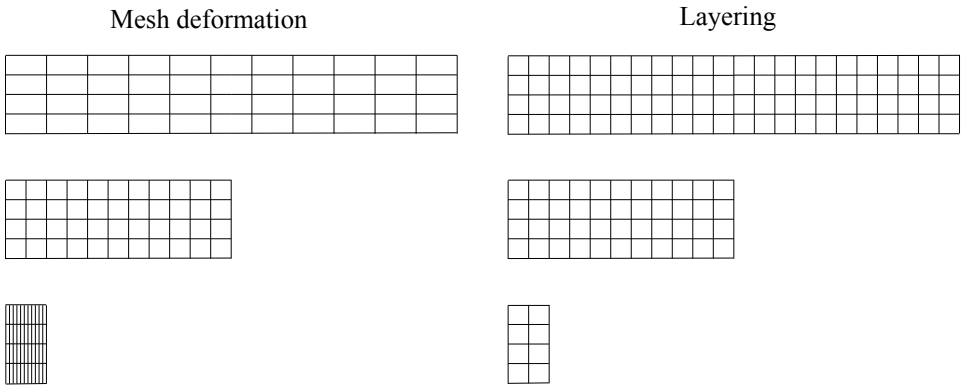

Figure 3: Comparison between mesh deformation and layering methods.

then the boundary motion influences all mesh points. The mesh deformation method does not change the number of cells since no topology modification is performed on the mesh. Then, when the domain has large deformations the mesh resolution varies considerably and it could cause a poor or excesive mesh resolution depending on the domain size.

On the other hand, the use of the standard layering technique, which uses constant layer height tolerances produces, a constant mesh resolution and an invariant aspect ratio which are desirable mesh qualities for many applications.

Figure 3 shows three different domain configurations using the mesh deformation and layering technique. It must be seen that it is not possible to maintain a constant mesh resolution and aspect ratio for all states when the mesh deformation method is used.

For some applications it is necessary to have a variable mesh resolution as the physical phenomena can have different gradients of the variables depending on the domain position. That is the case of internal combustion engines problems. In engine cylinders it is essential to have a good mesh resolution on the combustion chamber, specially for direct fuel injection engines, where air and fuel are mainly mixed. Furthermore, the mixture is ignited there and the combustion also takes place in this zone.

If a fine mesh resolution is defined for the combustion chamber, the standard layering technique with constant layer tolerances will produce an excessive amount of cells when the cylinder is completely expanded. On the contrary, the mesh deformation strategy will lack of sufficient mesh resolution. Moreover, with this method the mesh resolution on the combustion chamber is lost when the cylinder has its maximum size. Figure 4 describes the drawbacks of both mesh dynamics methods for accomplish local refinements,

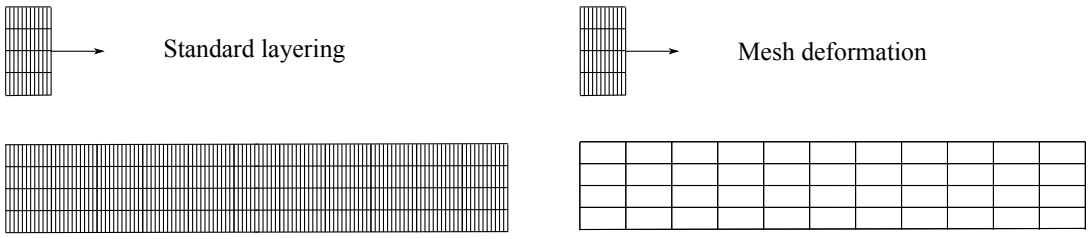

Figure 4: Layering and mesh deformaiton.

2.2.2 Spatial variable layer tolerances

This work presents a layering technique with spatial variable layer tolerances to allow two mesh resolutions on different regions of the domains. The method is specially designed for internal combustion engines, as it is necessary a fine resolution on the combustion chamber.
The new method is based on the definition of three types of layer size zones, a fine zone, a coarse zone and a transition zone. For the present work, the fine zone represents the combustion chamber and the rest of the cylinder is considered into the coarse zone. There is also a transition zone that is used to smooth the layer size variation in the region between the fine and coarse zones. It is important to remark that smooth mesh size transitions are very welcomed for minimizing numerical errors with a strong impact on the solution quality.
The set up of the problem consists in defining the spatial coordinates at the beginning of each layer size zone. The layer height is established only for the coarse zone, and additionally a size fraction factor must be provided to determine the layer height on the fine zone. The layer height for the transition zone are linearly interpolated from the fine and coarse values.

$$\begin{aligned} h_f &= s_f \times h_c \\ h_t &= (1-\alpha)\, h_f + \alpha h_c \end{aligned} \tag{1}$$

where h_c is the coarse zone layer height, s_f is the size fraction factor, h_f is the fine zone layer height, h_t is the transition zone layer height and α is the linear interpolate factor for the transition zone layer height.

Figure 5 presents the layers configuration as result of using spatial variable layer tolerances.

2.2.3 Multiple zones motion

The layering version developed in this work allows the treatment for multiple zones motion. This new feature is conceived specially for opposite piston engine cases where it is necessary to have independent definition of the pistons velocities. The motion zones have their own layering

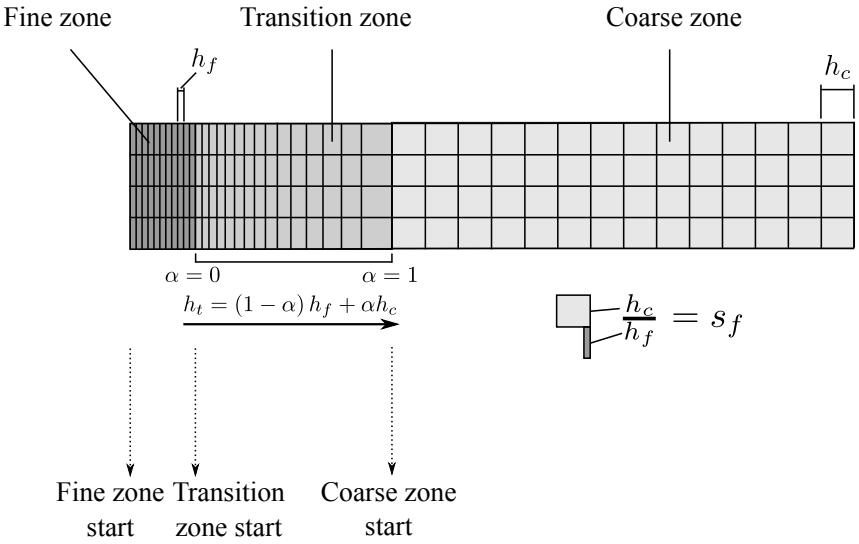

Figure 5: Layers configuration using spatial variable layer tolerances.

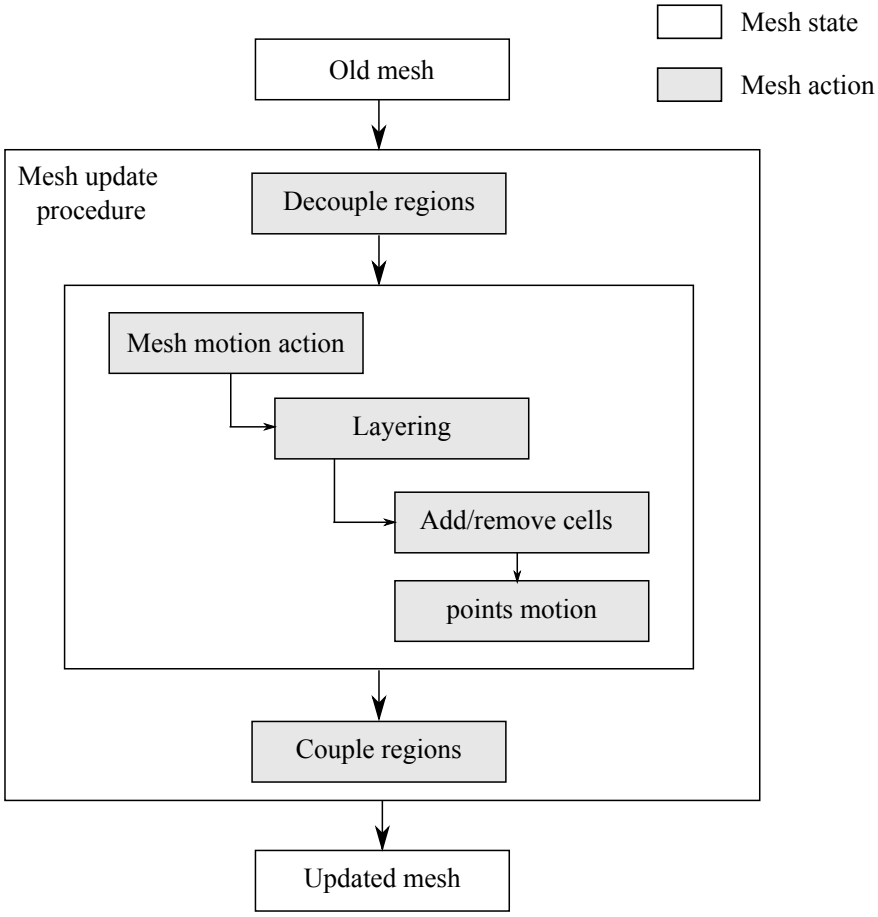

Figure 6: Scheme describing the mesh update procedure.

set up parameters and the layering procedure for each motion zone is called sequentially within the mesh update procedure. The method requires the topological changes to be performed on each motion zone not influencing the topology configuration of the others motion zones.

2.3 Mesh update procedure combining sliding interfaces and layering.

The mesh update procedure presented in this work is composed basically by a set of actions that produces the combination of *sliding interfaces* and *layering* mesh dynamic methods. The basic necessary actions are the same that were described in Section 2.1.1 where the mesh motion mechanism is achieved with the layering technique. The layering technique is composed by two actions. First, the topological action of removing or adding boundary cells is performed and then the point coordinates are moved to their new position following the displacement function. Finally, the global mesh update procedure concludes when the domains are coupled. Figure 6 presents a scheme with the mesh update actions described before.

3 FINITE VOLUME DISCRETIZATION

3.1 Arbitrary Lagrangian-Eulerian formulation

The combination of the *sliding interfaces* and *layering* mesh dynamics methods must be made according to the Arbitrary Lagrangian-Eulerian (ALE) formulation for finite volume problems. As described in Jasak and Tuković (2007), the finite volume discretization for each

cell of the domain is expressed as follows,

$$\frac{\rho^n \boldsymbol{\psi}^n V_P^n - \rho^o \boldsymbol{\psi}^o V_P^o}{\Delta t} + \sum_f \left(\phi_{f_{\text{rel}}} \boldsymbol{\psi}_f\right) = \sum_f \left(\rho \Gamma_\psi\right)_f \left(\boldsymbol{\nabla} \boldsymbol{\psi}_f \cdot \mathbf{S}_f\right) + \mathbf{S}_\psi V_P \tag{2}$$

where $\boldsymbol{\psi}$ is a tensorial transport variable, ρ is the fluid density, V is the cell volume, Γ_ψ is the diffusion coefficient, $\mathbf{S}_\psi$ is the volume source, $\mathbf{S}_f$ is the face area vector, $\phi_{f_{\text{rel}}}$ is the mass flux relative to the mesh motion, the subscript P represents the cell values, the subscript f represents the face values and superscripts n and o indicate the new and old time level values respectively. The relative mass flux $\phi_{f_{\text{rel}}}$ is the difference between the absolute mass flux $\phi_{f_{\text{abs}}}$ and mesh flux $\phi_{f_{\text{mesh}}}$,

$$\phi_{f_{\text{rel}}} = \phi_{f_{\text{abs}}} - \phi_{f_{\text{mesh}}} \tag{3}$$

The mesh flux is calculated as the sweep volume generated by mesh faces velocity $\mathbf{u}_{\text{mesh}_f}$ scaled with the density,

$$\phi_{f_{\text{mesh}}} = \rho \left(\mathbf{u}_{\text{mesh}_f} \cdot \mathbf{S}_f\right) \tag{4}$$

The mesh dynamics introduces in the standard Eulerian finite volume discretization the concepts of old volumes V^0 and the mesh flux $\phi_{f_{\text{mesh}}}$. Both terms must be calculated according to the spatial conservation law (Demirdžić and Perić, 1988),

$$\frac{V_P^n - V_P^0}{\Delta t} = \sum_f \phi_{f_{\text{mesh}}} \tag{5}$$

3.1.1 Old mesh volumes and mesh flux determination

It was described previously that the *sliding interfaces* strategy does not modify the adjacent cells in the decoupling and coupling action. Only interface faces and points are added or removed. However, the point merging actions that are performed in the slave interfaces points modify the slave adjacent cells volumes as the point coordinates are changed. This produces, after the decoupling action is performed, that the slave adjacent cells may have a different volume in comparison with the old mesh state. Then, the mesh motion action will generate on the mesh a total cell volume variation that is not correct, with the consequent bad definition of the mesh flux. This issue produces spurious values in the numerical solution when slave points are moved due to the merging action.

In order to avoid this issue, any geometrical calculation of cell volumes and mesh fluxes is performed on the mesh update procedure block that is shown in Figure 6. Using this strategy, the mesh state after coupling the regions is correct by itself: the regions are correctly coupled and the mesh points are updated to their new position giving by the desired motion function. Nevertheless, the mesh flux and old volumes are not yet defined.

The mesh cell volumes from the previous time-step, must be the old cell volumes for the new mesh state. The mesh update procedure generates intermediate mesh configurations which have different cell volumes values. Then, it is necessary to make a copy of the cell volumes before decoupling the regions to conserve the old mesh values. However, the updated mesh may have different number of cells and the old cells volumes must be mapped from the old mesh to the new mesh state taking into account the topology change. A similar problem takes place when

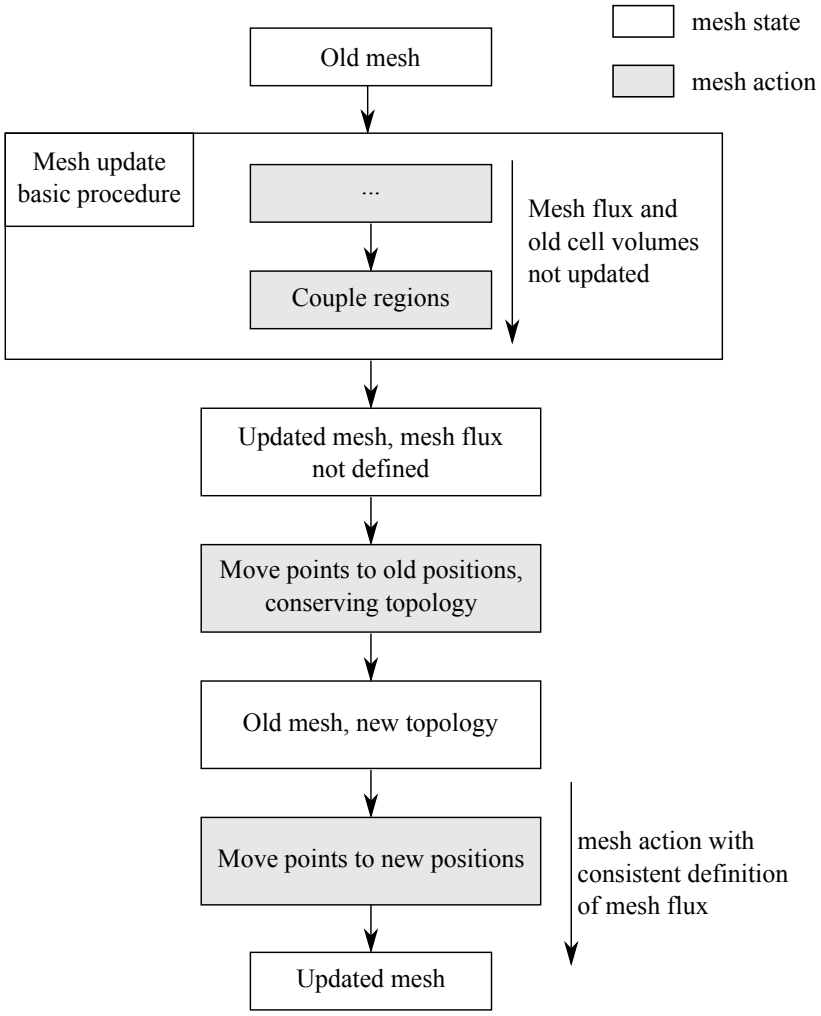

Figure 7: Scheme describing the new mesh update procedure including the two additionally mesh actions.

calculating the mesh flux of a face that is born in the new mesh topology. The old face position is undefined because the face did not exist on the previous mesh state. The definition of the face sweep volume is then unknown.

The previous problem is solved by condensing all mesh update procedures in an unique mesh dynamic action. In order to do this, two additionally mesh states are added to the scheme presented in Figure 6. After coupling the regions, the mesh points are moved to their old positions, but the new topology of the mesh is conserved. After this, the points are moved again to their new positions, but in this process no topology change is performed on the mesh as the previous state has already the new topology. The last mesh dynamic action changes the point coordinates, which enables a geometrical definition of mesh fluxes. This is consistent with the cell volume change and then the spatial conservation law described in Eq. (5) is satisfied. Finally the mesh update scheme is showed in Figure 7.

The mesh state with old points positions and new topology must have the same volume in comparison with the old mesh state. In order to achieve the volume conservation in two meshes

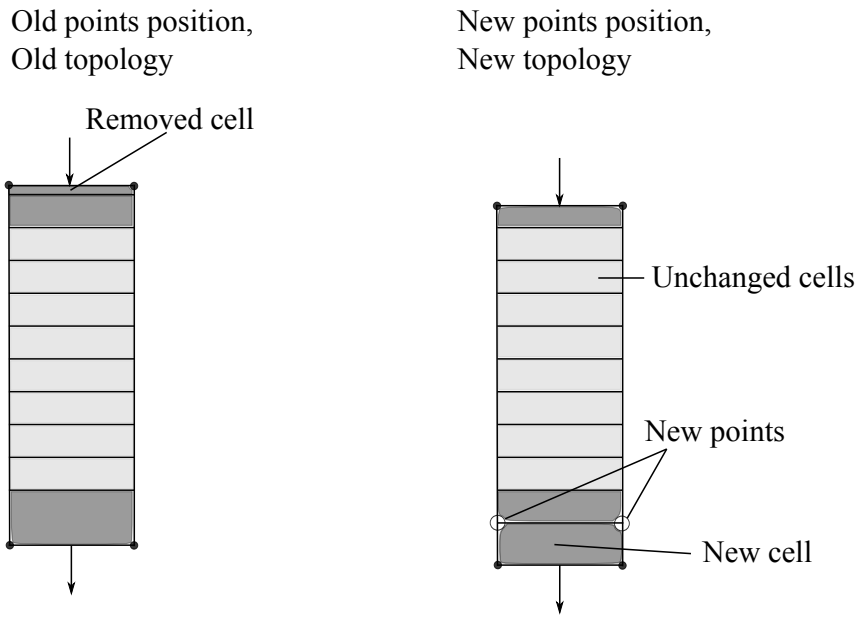

Figure 8: Mesh update example.

with different mesh topologies, next considerations are required: if a new cell is born in the new mesh state, its volume in the old state must be zero. On the other hand, if an old cell disappears in the new mesh state, this cell must be merged into a neighbouring cell in order to represent the old mesh state with the new topology.
Figures 8 and 9 explain a 2D mesh update example which includes the two considerations described before. The example case has two boundary faces which moves towards down. The new mesh state has a new topology and new point positions where a cell is removed in the upper part and a cell is added in the lower part. The rest of the cells remain unchanged. Figure 8 shows the initial and final mesh states of the example case.

The definition of the old points position for the mesh with the new topology configuration is shown in Figure 9. The volume of the removed cell is merged with its adjacent cell, and the new added cell arises from a zero volume cell.

Because the last mesh update action does not involve any topology change, it is not necessary any field mapping and the definition of mesh fluxes and old volumes is done correctly through performing geometrical computations.

3.2 Compressible fluid solver

The simulations of the opposite piston engine are done using the OpenFOAM®compressible fluid solver `rhoPimpleDyMFoam`. This solver uses the PIMPLE algorithm for the velocity-energy-pressure coupling. The PIMPLE algorithm uses the SIMPLE procedure (Patankar, 1980) which includes for each SIMPLE iteration a defined number of PISO correctors (Issa, 1986; Márquez Damián, 2013).
The flow on the engine simulation suffers of high pressure gradients that can induce supersonic flows. This flow condition may compromise the stability of the solver if the standard Poisson equation for the pressure is used. Therefore the convective terms of the pressure equation are treated implicit replacing the density values as a function of pressure. This form of the pressure equation increases the velocity-energy-pressure coupling and therefore the solver stability.

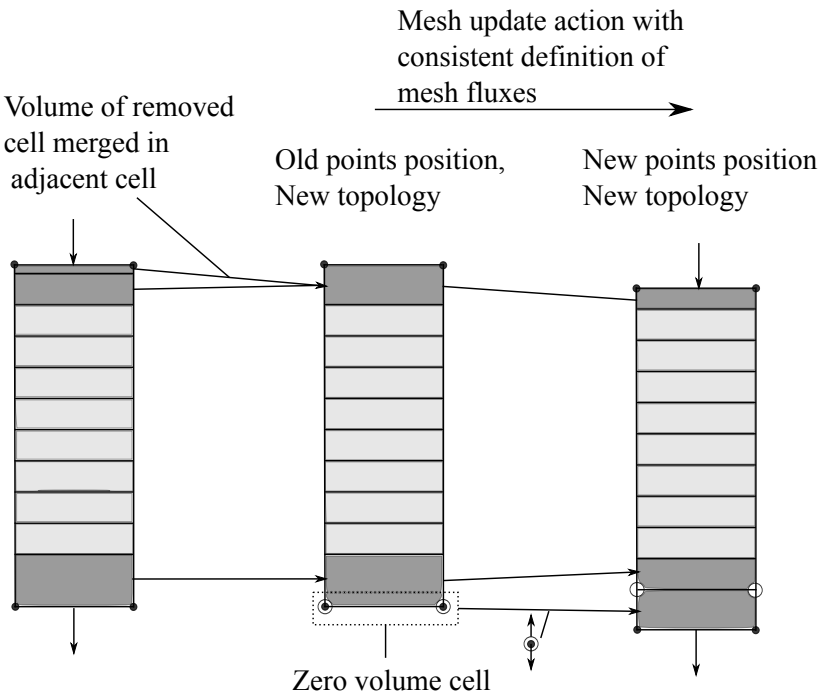

Figure 9: Mesh update example. Consistent definition of mesh fluxes through updating the mesh without topology change.

For the turbulence model, the standard k-epsilon model is used (Launder and Spalding, 1974). A marker function is incorporated in the solver by solving scalar transport equation. The diffusion coefficient for the marker function is calculated as the turbulent component of the air thermal conductivity.

4 SIMULATION OF AN OPPOSITE PISTON ENGINE

In this section an engineering problem is solved using the new developed mesh dynamics tools for CFD. The problem consists in the evaluation of a new design of an opposite piston engine which has a new configuration of its combustion chamber. The aim of the simulation is to determine the qualitative aspects of the flow inside the engine and principally to study the flow patterns on the combustion chamber. Additionally, a preliminary simulation of the combustible injection is achieved by using a marker function to represent the combustible dynamics inside the flow.

The engine geometry is schematic represented in Figure 10 where the main components of the cylinder are described. The motor is a two stroke engine, where the pistons manage the air inlet and outlet by covering and uncovering the ports as they move across the cylinder.

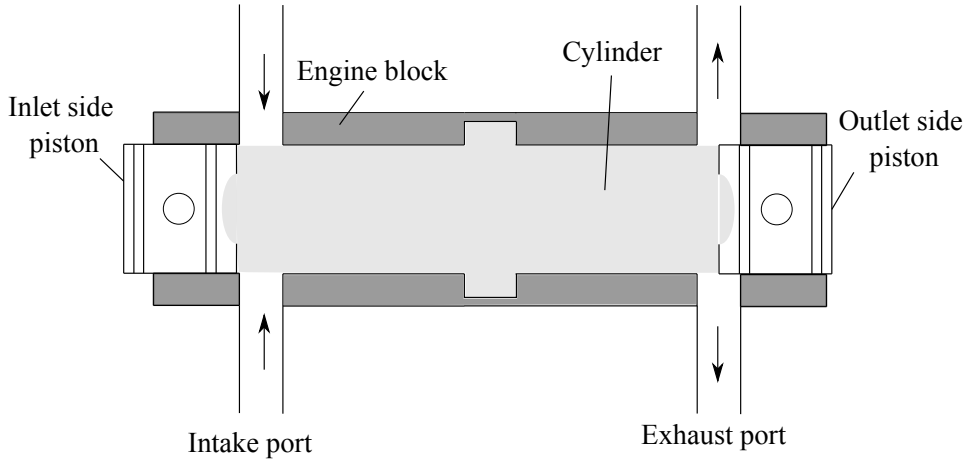

Figure 10: Main parts of the engine cylinder.

Due to a new design of the engine block configuration, the block is thinner on the cylinder middle part. This produces a higher diameter for this part of the cylinder and thus the combustion chamber geometry is modified. The place where the engine block is thinner defines a new zone in the combustion chamber which is defined as the *ring* zone.
The combustion chamber is conformed when the pistons determine the minimum volume for the cylinder. This point is illustrated in Figure 11 where the *ring* zones is also detailed.

It should be noted in Figure 11 that the outlet side piston overlaps a fraction of the ring zone boundary. Therefore an additionall *sliding interface* is necessary for the ring zone.

4.1 Computational geometry

The computational geometry of the engine is decomposed in four separate regions: the intake port, the exhaust port, the cylinder and the *ring* part. The different regions of the engine are shown in Figure 12.

The communication between the decoupled regions is done by the interfaces. The cylinder is a moving domain as its size varies due to the pistons movement. Therefore sliding interfaces are used to accomplish the relative tangential movement between the cylinder and the rest of

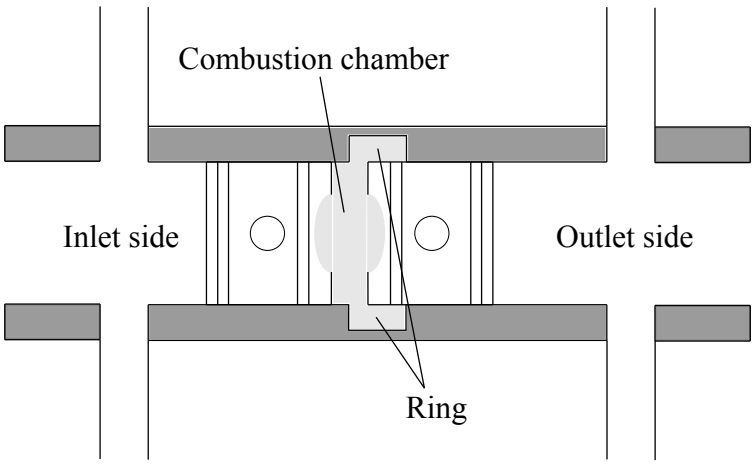

Figure 11: Combustion chamber.

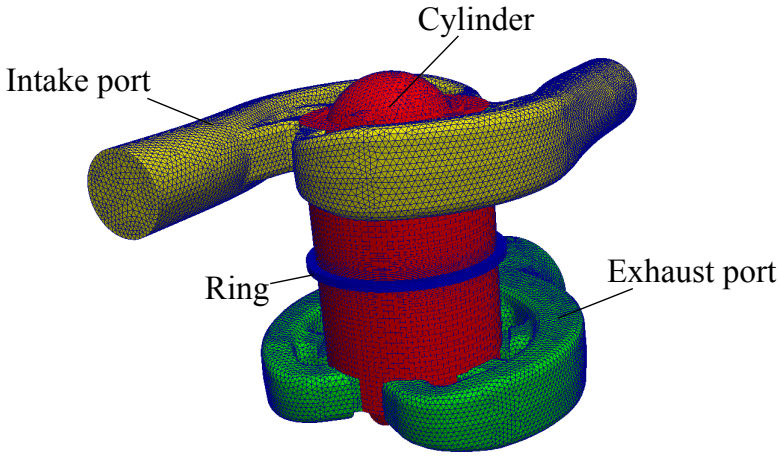

Figure 12: Computational domains.

the regions.
The interfaces are divided in two groups in order to set up the sliding interfaces. On group is formed by the cylinder interface which is the master interface and the rest form the second group which correspond to the slave interfaces.

The mesh adaptation of the cylinder is performed by the layering technique. The cylinder has two moving zones that correspond to both pistons respectively. Each zone has it own velocity expression because the pistons are driven with independent crankshafts.
One of the simulation targets is to evaluate the flow patterns in the combustion chamber. Therefore it is necessary to have a good mesh resolution on this zone. For this requirement, the spatial varying layering tolerances presented in this work are used. The different layer sizes for the cylinder mesh are distributed in three zones. The coarse zone is located on the major part of the cylinder where the combustion chamber is not included in this zone. In the coarse layer zone the layer size is 2 mm The fine zone includes the combustion chamber and has an

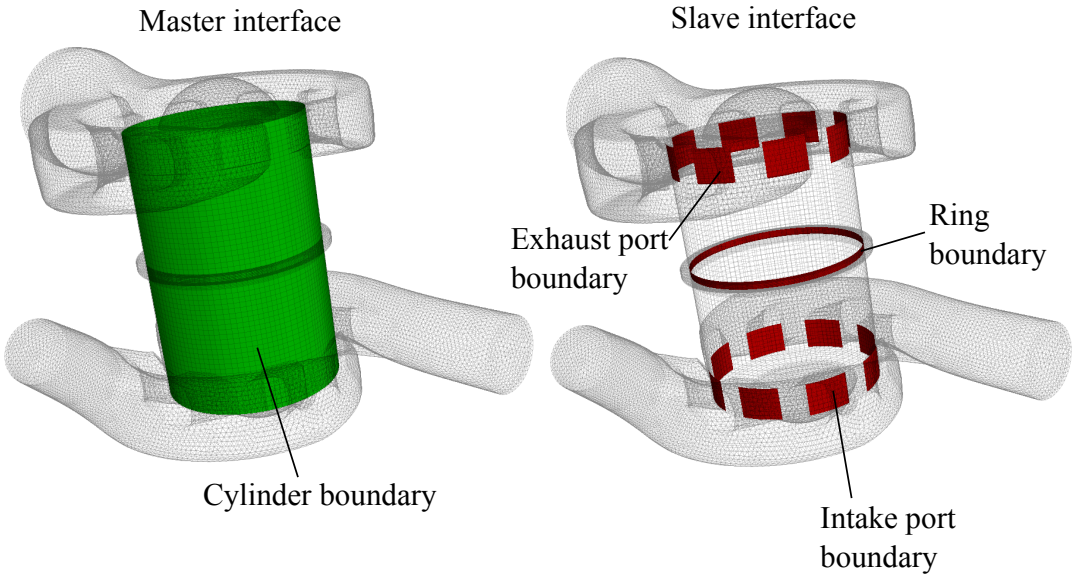

Figure 13: Sliding interfaces set up. Master and slave interfaces .

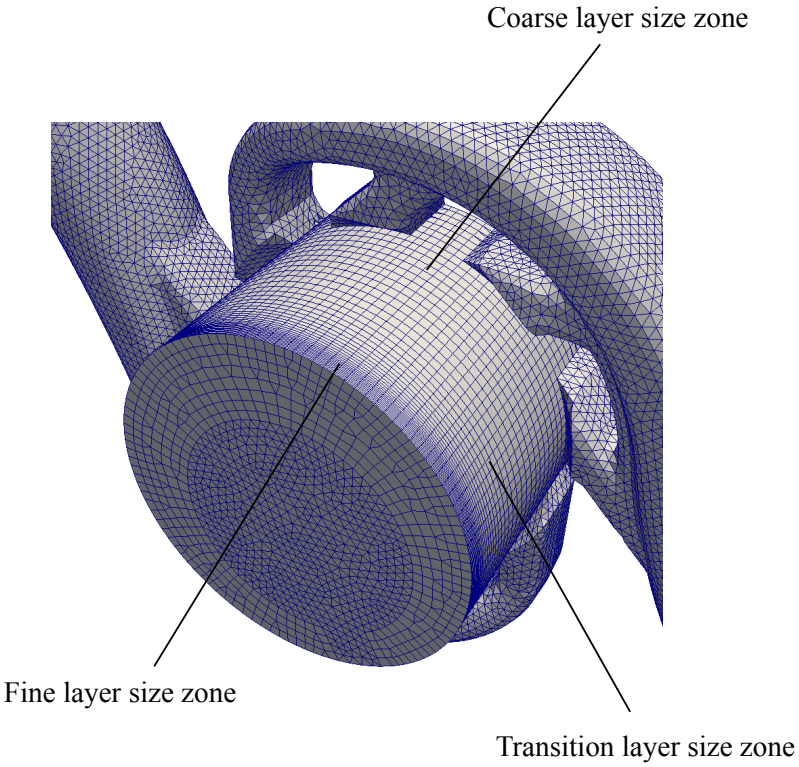

Figure 14: Three different layer size zones in one side of the cylinder.

axial length of $6\,\text{mm}$. The layer size in the fine zone is $0.3\,\text{mm}$. The transition zone, which is between the coarse and fine zone, has a layer size which has a linear variation between the coarse layer size and fine layer size. The three layer size zones of the cylinder mesh are shown in Figure 14.

4.2 Case Configuration

The engine is simulated without considering the fuel-air mixture ignition. Regarding to this, the engine motion can be considered as generated by an external motion source.
The fluid is considered as ideal gas with the physical properties of air. The temperature dependence of the fluid dynamic viscosity and fluid thermal conductivity is determined applying the Sutherland transport model. Table 4.2 lists the fluid physical parameters used in the simulation.

Parameter	Value
Specific ideal gas constant (R)	$287\,\frac{\text{J}}{\text{Kg K}}$
Heat capacity at constant pressure (C_p)	$1005\,\frac{\text{J}}{\text{Kg K}}$
Sutherland coefficient (A_s)	1.45×10^{-6}
Sutherland temperature (T_s)	$110.4\,\text{K}$

Table 1: Fluid physical parameters.

4.2.1 Boundary conditions

The boundary conditions for the problem are total pressure at the inlet boundaries and static pressure at the outlet boundary. The value of the inlet total pressure is 190000 Pa which corresponds to the nominal value of the engine turbo compressor. The outlet static pressure is set up as atmosphere pressure with a value of 101300 Pa.
For the temperature conditions, the inlet boundaries have a value of 373 K, which stands for the turbo compressor air temperature. At the outlet and at the walls a zero gradient condition of temperature is set up. The velocity has a zero gradient condition in the inlet and outlet boundaries. At the walls the velocity is zero.
The epsilon treatment at the inlet uses a fixed value of 200, where the inlet adjacent cell values are calculated using a mixing length model with a value of $2.5\,\text{mm}$. At the walls, an epsilon value of 200 is defined using the standard wall function for wall adjacent cells. For outlet boundaries, a zero gradient condition is configured.
The turbulent kinetic energy values at the inlet are fixed in 1 where a turbulent intensity value of 5% is determined for calculating inlet adjacent cell values. At the walls, the standard wall function is used with a value of 1. The outlet boundaries are configured with the zero gradient constraint.

4.2.2 Numerical set up

The problem is solved with the OpenFOAM®application `rhoPimpleDyMFoam`. An upwind difference scheme is used for the convective terms. The diffusion terms are solved with linear cell values interpolation and the temporal discretization is done with the implicit Euler scheme. For the outer SIMPLE iterations a residual relative tolerance of 0.01 is configured. The number of PISO iterations is set up in 3 with 1 non orthogonal correctors iteration.

4.2.3 Temporal set up

The engine is simulated with a crankshaft speed of 3000 revolutions per minute. The case is solved until a steady state solution is reached. This state is achieved after 7 engine cycles, where each cycle last $0.02\,\text{s}$ at the proposed engine velocity. Hence the simulation has a total duration of $0.14\,\text{s}$.
The time-step size is limited by the piston Courant number. It is defined as follows,

$$Co_p = \frac{v_{p_{\text{max}}} \Delta t}{h_{\text{min}}} \tag{6}$$

where Co_p is the piston Courant number, $v_{p_{\text{max}}}$ is the maximum piston velocity, Δt is the simulation time step and h_{min} is the minimum layer size tolerance.
The piston Courant number must be less than one to avoid negative cell volumes. For the present dynamic mesh set up, the minimum layer tolerance is $h_{\text{min}} = 1.5 \times 10^{-4}\,\text{m}$ and the maximum piston velocity is $v_{p_{\text{max}}} \cong 8.5 \frac{\text{m}}{\text{s}}$. With this parameter values, the time step must be less than $\Delta t < 1.76 \times 10^{-5}$ s. The time-step of the simulation is configured in 10^{-5}s.

4.3 Results

The simulation presents a steady state periodic solution after 7 engine cycles. The steady state solution can be seen on Figure 15 which exhibits the evolution of the in-cylinder pressure along the whole simulation time.

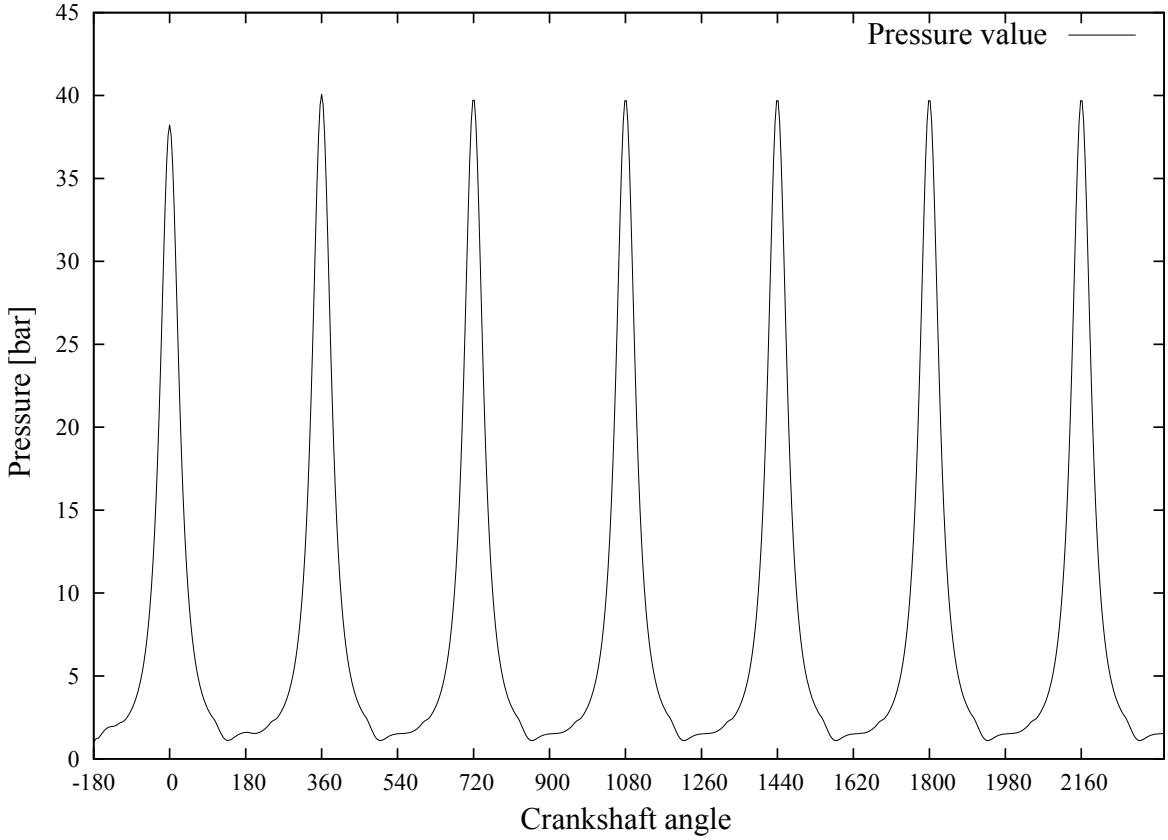

Figure 15: In-cylinder pressure during 7 engine cycles.

The new mesh dynamic tool works successfully for the proposed case. Figure 16 exhibits multiple instants of the simulation for the compression phase of the engine cycle. The figures are composed by velocity vectors located on a slice plane which is coloured representing the velocity magnitude values.

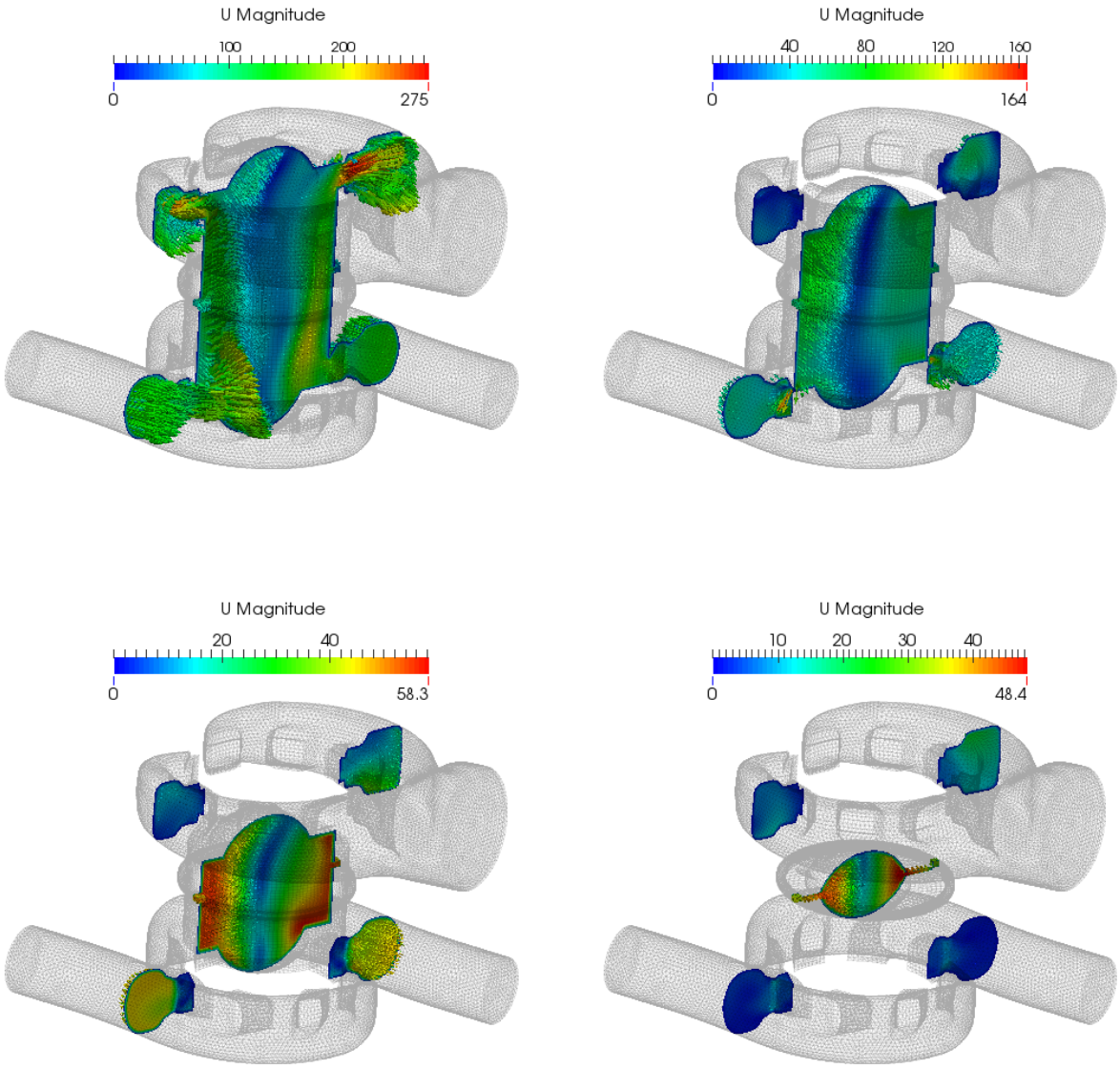

Figure 16: Different mesh states on the compression phase.

4.3.1 Combustion chamber flow

The flow patterns on the combustion chamber presents a swirl flow where the rotation centre is slightly displaced from the cylinder center due to the pistons cavities. The angular velocity of the flow depends on the pistons position. The maximal velocities are found when the engine cycle is in the bottom dead center (BDC) with velocities over 150 metres per second. These high velocities values are induced by the incoming air which flows from the opened intake ports. The velocity magnitudes decrease after the intake ports are closed and they reach the minimum values on the cycle top dead centre (TDC) with maximum values of 52 metres per second. In all cases, the flow inside the ring has the minimum relative velocities. Velocity vectors in a transversal plane that slices the combustion chamber including the ring zone are, shown in Figure 17.

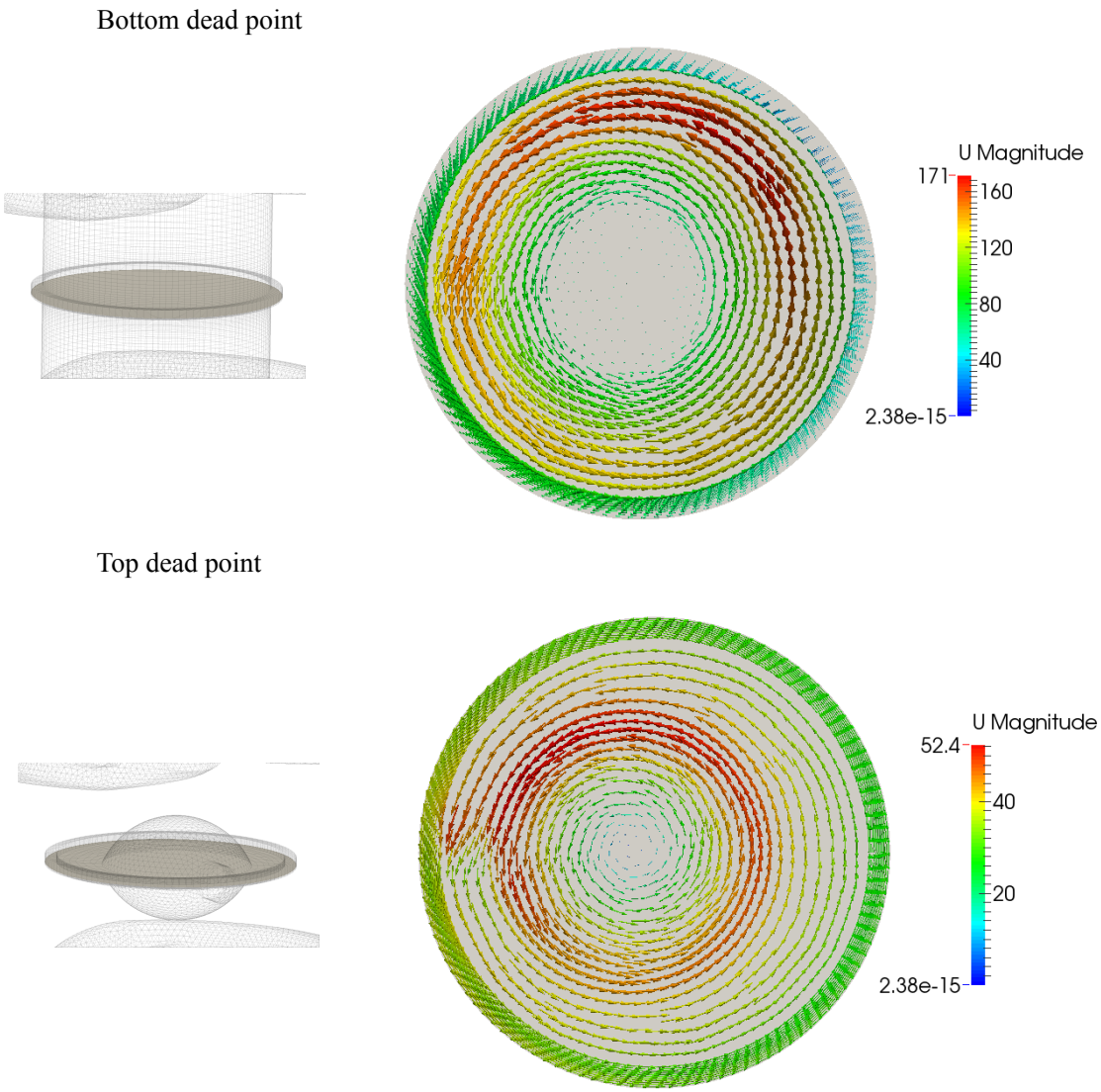

Figure 17: Velocity vectors on a slice plane for the both extreme positions of the cylinder.

4.3.2 Marker function

For a better evaluation of the flow characteristics a marker function is used. The marker function emulates the concentration of fuel in the fuel-air mixture. The calculated transport of the marker function is a preliminary approximation of the fuel dynamics. It is important to remark that the basic model underestimates the fuel air mixing since the combustible volumetric expansion due to vaporization is not considered.
The initialization of the marker function is done with the aim of reproducing the direct combustible injection. The marker is introduced in the cylinder in the form of a cone. The marker injection velocity is determined in 50 meters per second and the cone angle is defined in 65 degrees. The marker injection starts 123 crankshaft angle degrees (CAD) before TDC and has a total duration of 10 CAD. Figure 18 shows the evolution of the marker during the injection time. It can be seen that the cone shape of the emulated spray is deformed due to the flow induced advection.

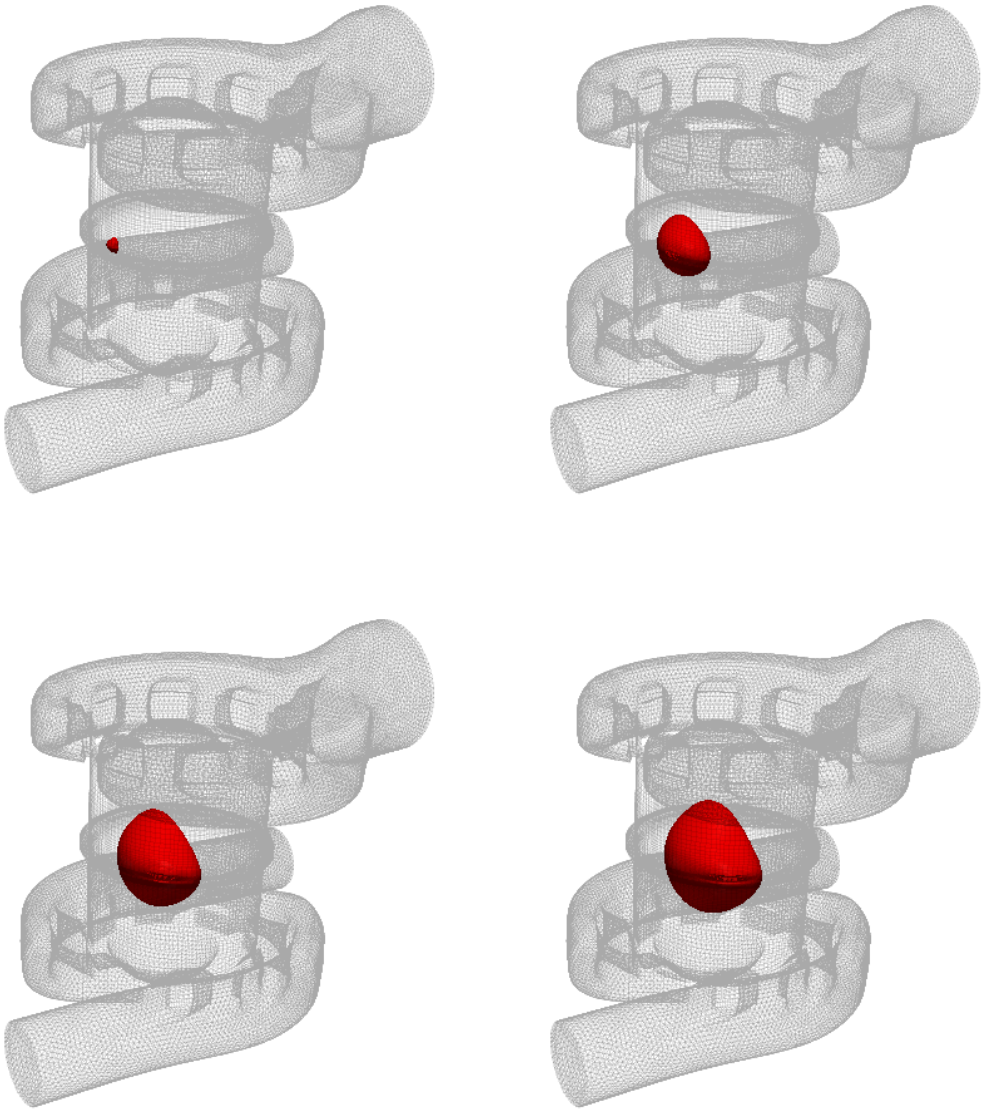

Figure 18: Injection phase of the marker function.

A determined quantity of the marker is injected in order to represent the stoichiometric fuel-air mixture with an unity value of the marker scalar function. When the marker value is above one, the fuel air mixture is said to be rich. On the contrary, if the marker value is below one, there is a deficiency of fuel and then the mixture is lean.
The root mean square error (RMSE) of the marker is used to evaluate the mixture quality. Figure 19 indicates the RMSE and mean value of the marker inside the cylinder.

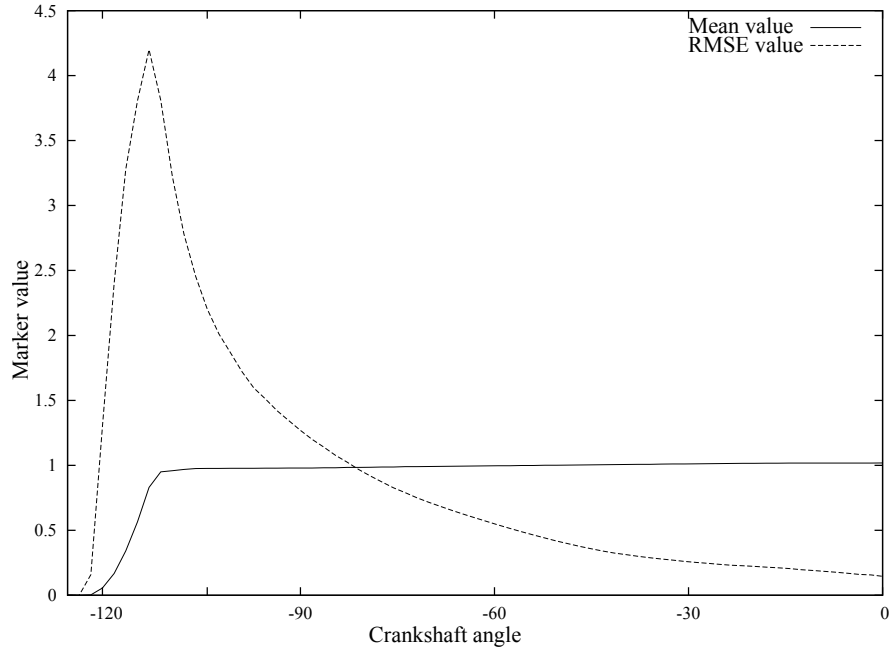

Figure 19: RMSE and mean value of the marker inside the cylinder.

The marker value evolution shows a good mixing procedure because the RMSE decreases along the compression phase. The same analysis is done only for the *ring* zone as can be seen in Figure 20.

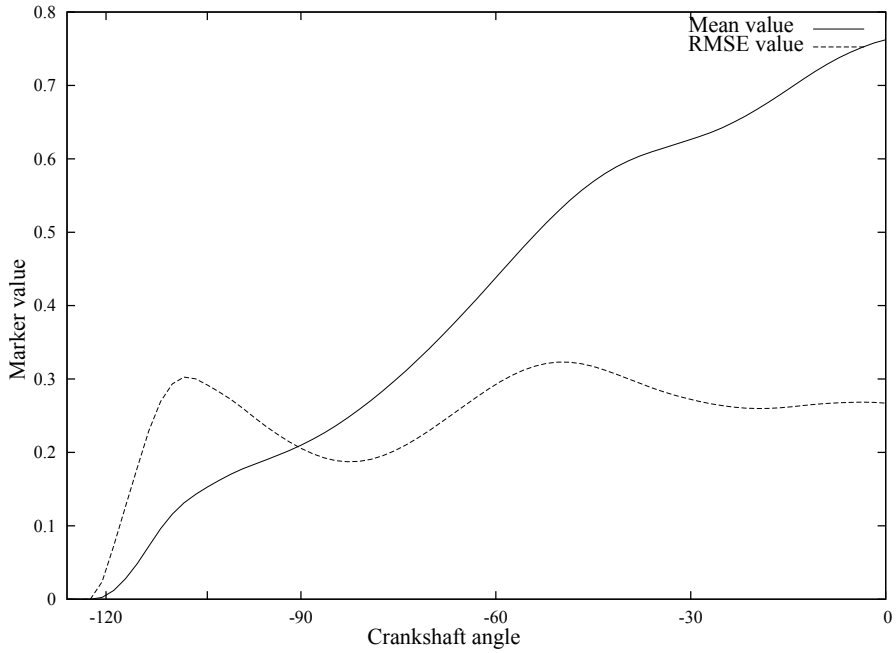

Figure 20: RMSE and mean value of the marker inside the *ring* zone.

The marker values in the *ring* zone indicates lean fuel-air mixture. The RMSE values in this zone are higher than the values that were obtained in the cylinder zone.

The marker values in the combustion chamber are shown in Figure 21 at the moment of ignition start. The combustion chamber is divided in three different regions as function of the marker values. The blue zone represent the locations where the marker has values below 0.8. The green zone has marker values between 0.8 and 1.2 and the red zone represents the zone with marker values above 1.2. In the figure is detailed the ignition point. The marker values indicates a lean mixture in this zone. This situation may produce a potentially ignition problem in the engine.

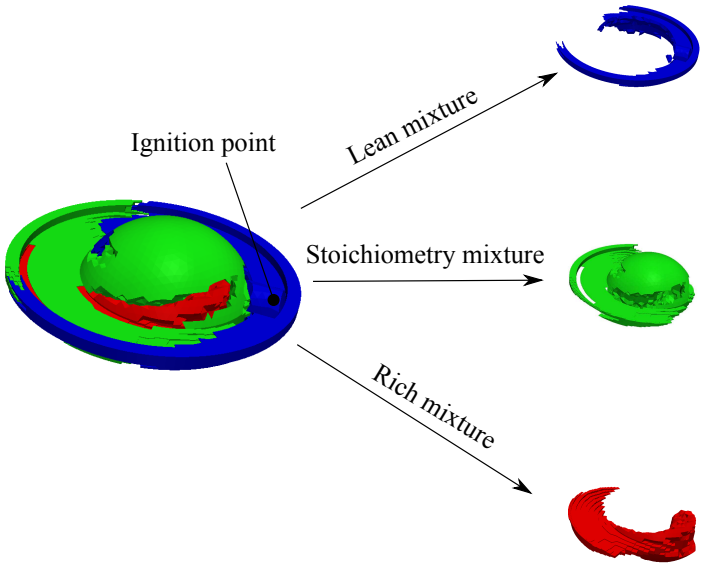

Figure 21: Different zones in combustion chamber as function of the function marker values.

5 FINAL NOTES

A numerical simulation of an opposite piston engine was performed using the OpenFOAM® suite. The engine kinematics that is presented in the problem was addressed with a full automated mesh dynamic method. The method is based on the combination of *sliding interfaces* and *layering* mesh dynamics technologies.

The mesh update method was extended for the treatment of multiple motion zones and for the use of spatial variable layering tolerances which enables different layer size zones. The implemented mesh update strategy worked successfully for accomplish the pistons motion and the cylinder-ports and cylinder-ring communication.

For this version of the mesh dynamic method, a consistent calculation of the mesh flux is defined. This is done by condensing the set of mesh actions in an unique process that moves only the point coordinates without doing any topology modification on the mesh. This procedure eliminates spurious solution values due to wrong definition of the mesh face fluxes and old cell volumes.

The flow patterns in the combustion chamber were studied. The simulation results indicate the presence of a swirl flow in the combustion chamber. The air velocity magnitude in the ring zone has lower values than in the rest of the combustion chamber.

A preliminary evaluation of the direct fuel injection and combustible dynamics was performed.

The results showed that there is a potential problem at the ignition moment, because the spark location point has a lean fuel-air mixture.

6 ACKNOWLEDGMENTS

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grant PIP 112 201101 00331), Universidad Nacional del Litoral (UNL, Argentina, grant CAI+D 2011 PI 501 201101 00435 and PJ 500 201101 00015) and Agencia Nacional de Promoción Científica y Técnica (ANPCyT, Argentina, PICT 2013-0830)

REFERENCES

Aguerre H., Marquez Damian D., Jimenez J., and Nigro N. Modeling compressible fluid problems with openfoam. XX Congreso sobre Métodos Numéricos y sus Aplicaciones, 2013.

Demirdžić I. and Perić M. Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids*, 8(9):1037–1050, 1988.

Farrell P. and Maddison J. Conservative interpolation between volume meshes by local galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, 200:89–100, 2011.

Issa R.I. Solution of implicitly discretized fluid flow equations by operator-splitting. *J. Comput. Phys.*, 62:40–65, 1986.

Jasak H. *Error analysis and estimation in the Finite Volume method with applications to fluid flows*. Ph.D. thesis, Imperial College, University of London, 1996.

Jasak H. and Tuković Z. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30(2):1–18, 2007.

Johnsson M. *Stratified scavenging computations in two-stroke engines using OpenFOAM*. Master's Thesis, Chalmers University of Technology, Goteborg, Sweden, 2010.

Launder B. and Spalding D. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269 – 289, 1974.

Lucchini T., D'Errico G., Jasak H., and Tuković v. Automatic mesh motion with topological changes for engine simulation. SAE Technical Paper 2007-01-0170, 2007.

Márquez Damián S. *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces*. Ph.D. thesis, Facultad de Ingenieria y Ciencias Hidricas, Universidad Nacional del Litoral, 2013.

Nilsson H., Page M., Beaudoin M., Gschaider B., and Jasak H. The openfoam turbomachinery working group, and conclusions from the turbomachinery session of the third openfoam workshop. 24th Symposion on Hydraulic Machinery and Systems, 2008.

Patankar S. *Numerical Heat Transfer and Fluid Flow*. Series in computational and physical processes in mechanics and thermal sciences. Hemisphere Publishing Company, 1980. ISBN 9780891165224.

Vissers A. *Port size, geometry and layout design in a Two-Stroke JaqEngine*. Master's Thesis, Eindhoven University of Technology, Eindhoven, the Netherlands, 2007.

Weller H., Tabor G., Jasak H., and Fureby C. A tensorial approach to computational continuum mechanics using object orientated techniques. *Computers in Physics*, 12(6):620 – 631, 1998.