Automated moving mesh techniques and re-meshing strategies in CFD applications using morphing and rigid motions.
Technical Report

M. M. Profir

April 12, 2012

Abstract

The presence of moving meshes when motions of fluids and solids are being simulated requires a good control of the topological deformations that can take place both into the surface mesh and into the volume cells. The work done in getting the control of the mesh morphing techniques in CFD applications of fluid-structure interactions and rigid bodies motions is here presented. The effects of imposed rigid body motions on a fluid mesh surrounding or surrounded by a solid body are analysed. Surface boundary mesh extraction and CAD geometry updating strategies for the preservation of a good quality mesh during the calculation are investigated and developed. Java programming is employed in order to automate the process of re-meshing to avoid degeneration.

1 Introduction

In the Computational Fluid Dynamics (CFD) simulation codes the physical domain to be simulated is divided into a finite number of small control volumes ([3]) corresponding to the cells of a computational grid where discrete versions of the integral form of the continuum transport equations are applied.

In the applications involving moving fluids and solids it is important to keep under control the deformation of the meshes employed, by means of specific morphing techniques and re-meshing procedures. This is the reason why we focused on these techniques in the framework of a two-years individual research project.

The major objective of the project was to study the mesh morphing techniques in order to use them in applications that principally treat with fluid-structure interactions and rigid bodies motions. In such applications, when local deformations of surfaces occur, bad topological effects like warped, twisted or self-intersected faces can easily lead to the appearance of negative volume cells, determining an error in the numerical method and the simulation to stop running. We look hence at the CFD as an excellent source of applications allowing us to approach the mesh morphing if a strong continuum mechanics background was missing.

In the first year of research activities we acquired some theoretical knowledges about the physically-based deformable models (starting from the state of the art report [5]) and the recent advances in mesh morphing (following [1]). In parallel, we acquired some specific technical skills (like the use of an implementation code and a programming language). An overview of these theoretical and technical aspects was given in [7]. We proceeded then with a detailed analysis of the implementation methods in the updated and in continuous evolution (at the moment 6.06) version of the Star-ccm+ code, the major tool for our approach. The referential techniques concerning the morphing of the meshes together with their application in specific numerical simulations have been illustrated in [8].

For the second year of research activity, the foreseen objective was the application of the morphing techniques in CFD simulations that especially treat with structural and moving parts where local deformations occur. In particular, the objective was to apply them in cases where the deformations had to be considered in relation to specific physical requirements: rigid body motions, solid displacements, stresses, and also physical fields like fluid pressure, temperature or velocity.

The intention in this technical report (reflecting the last 12 months of research activity) is to give a demonstration of how the techniques that we have investigated and developed provided a suitable control of the mesh deformations in cases of interactions of fluids with rigid bodies.

In Section 2 we briefly explain the mesh morphing procedure, as it is employed in the Star-ccm+ code ([6]), giving also some illustrative examples.

In Section 3 we show how the mesh morphing has been applied in simulations treating with interactions between fluids and displacements of solids in a finite volume stress analysis.

In Section 4 we expose how we used the morphing in simulations of rigid body motions (RBM); we also illustrate the automation capability of the code, together with the description of some metrics for mesh quality that the code puts at our disposal.

In Section 5 we describe the study case of the periodical translational motion of a piston inside a larger pipe. This situation gave us the input to develop various strategies of complete or partial re-meshing of the physical domain, needed when a high level of degeneracy of the cells required it.

In Section 6 we prove the compatibility of the morphing/re-meshing techniques with the physics involved, by taking into consideration and analysing two cases of complex physical domains that are subjects to morphing/rigid motions and to mesh re-creation.

2 Mesh morphing procedure in Star-ccm+

The volume mesh in a simulation is the mathematical description of the geometry of the problem being solved and it is formed by vertices, faces and cells.

The Morphing is that type of motion which makes possible the deformation of the computational mesh in response to various causes. During the process of morphing, the mesh vertices are redistributed in response to the movement of a set of control points. The "Mesh Morpher" solver, which becomes active when a morphing motion is assigned to a region, takes the control points and their associated displacements and generates an interpolation field. The interpolation field is then used to displace the vertices of the mesh (based on the

multi-quadric interpolation theory ([4]).

Control points can be initialised using existing mesh vertices on boundaries or interfaces, or can be specified independently of the mesh using a Control Point List. In the first case, the displacement can be specified directly, or by using a grid velocity, the displacement field of a finite volume stress calculation (therefore a stress-related deformation), or a rigid body motion calculation (6-DOF motion). In the code, the way the Morpher collects the control points and their known displacements from the surface mesh of the boundaries is implemented through various methods that we will briefly describe below with some examples.

The control points and their associated displacements hence form the underlying data that the morpher needs in order to generate an interpolation field. This can be considered to act as a field function that can be called for every mesh vertex in the region to find the new position to which the vertex should be moved. The steps of the morphing procedure would therefore be:

I. The morpher collects control points and their known displacements from boundaries (or, in case, from control points tables) within the region, with one of the following methods: **Fixed, Displacement, Grid Velocity, Solid Stress, Rigid Body, Motion.**

II. After an initial preparation, these control points and their known displacements are used to generate the interpolation field.

   Each control vertex is associated with a displacement vector that is to be used to move nearby vertices, using multiquadric interpolation theory.

   To generate the interpolation field, a system of equations is solved, using the control vertices and their specified displacements: for every control vertex i, its displacement $d_i^{'}$ is approximated with the combination

$$d_i^{'} = \sum_{j=1}^{n} \lambda_j \sqrt{r_{ij}^2 + c_j^2} + \alpha \tag{1}$$

   with $r_{ij} = \|x_i - x_j\|$ the magnitude of the distance between two vertices, λ_j expansion coefficient, x_i the position of vertex i, n the number of control vertices and c_j the basis constant. The additional constraint on the constant vector α

$$\sum_{j=1}^{n} \lambda_j = 0, \tag{2}$$

   is meant to make the expansion bounded for large x.

   From the equations (1) and (2) the Cartesian components of all λ_j and the components of the constant vector α are obtained. Once these components are obtained, the desired interpolation field is:

$$d'(x) = \sum_{j=1}^{n} \lambda_j \sqrt{r^2 + c_j^2} + \alpha, \tag{3}$$

   with $r = \|x - x_j\|$.

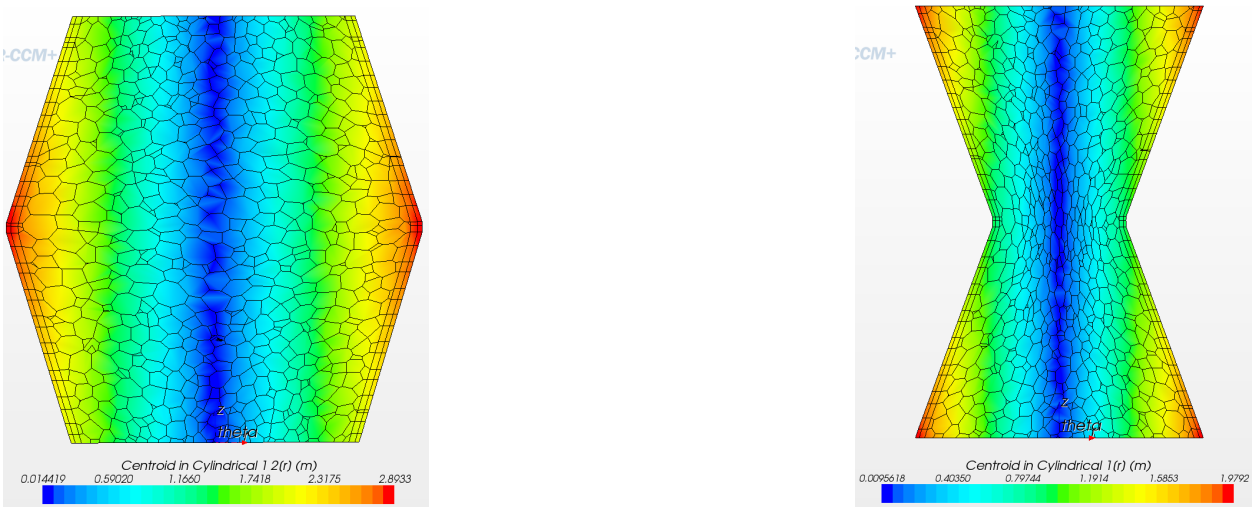
III. The interpolation field (3) is applied to all mesh vertices in the region.

IV. Some final adjustments are made to mesh vertices on or near boundaries.

In Star-ccm+, the Morpher collects the control points and their known displacements from the surface mesh of the boundaries using several methods that we will briefly describe below with some examples.

- **Displacement** - the displacement can be directly applied, when for example, to obtain the expansion of a cylinder (Figs. 1b, 1a), a linear displacement can be defined on the walls in the radial direction of a cylindrical coordinate system:

$$\delta_r = \begin{cases} 0.00625x_z, & x_z < 3 \\ 0.00625(6 - x_z), & x_z \geq 3 \end{cases} \tag{4}$$



(a) (b)

Fig. 1: Plane section in expanded Cylinder (a) according to rel.(4) and in contracted Cylinder (b) according to (5)

- **Grid Velocity** - using a grid velocity; for example, a velocity in the radial direction can be defined to obtain the contraction of a cylinder (Fig. **??**):

$$v_r = \begin{cases} -0.003x_z, & x_z < 3 \\ -0.003(6 - x_z), & x_z \geq 3 \end{cases} \tag{5}$$

where v_r is the radial velocity and x_z is the axial coordinate of the face centroid.

- **Solid Stress** - using the displacement field of a finite volume stress calculation; for example, for the simulation of a vibrating pipe in the fluid - structure interactions analysis, the fluid mesh representing the inner region of the pipe deforms under an external load applied on the pipe, while the solid pipe displaces in real time (Fig. 2).

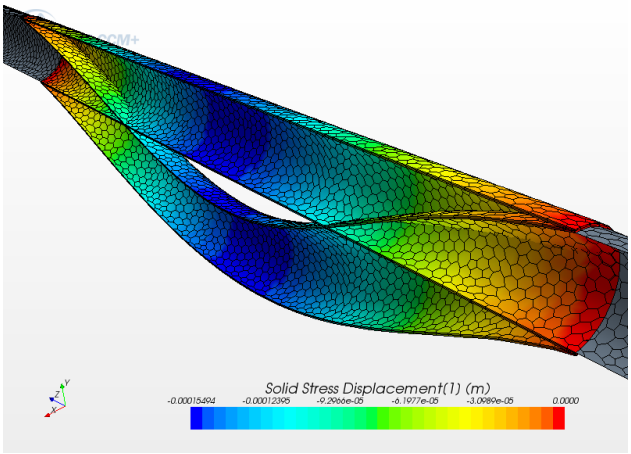


Fig. 2: Fluid mesh morphing provided by a Solid Stress calculation

3 Morphing in fluid-structure interactions (FSI)

In applications treating with interactions between fluids and solids, the morphing is the deformation of the fluid grid which occurs when the fluid vertices move to conform to the solid structure and to maintain a good quality of the fluid grid. This is called a "topologically constant" operation; the cells maintain the same neighbours but their shapes change in time. The fluid transport equations are solved using the Arbitrary Lagrangian Eulerian (ALE) technique, in which the motion of the vertices is arbitrary but it must conform to the moving boundaries and provide a reasonable quality mesh.

As showed in Section 2, the Star-ccm+'s Morpher uses a multi-quadric morphing model based on radial basis functions to define the motion of interior vertices and to transfer informations over the fluid-structure interface ([2]). In the case of the fluid-solid interactions, this motion originates from the motion of the vertices on the structural surface. Since a morphing strategy can easily lead to poor quality cells, the behaviour of fluid meshes in relation to displacement of solid parts (pipe walls or pistons) is here analysed.

3.1 The geometry and the physics of a simplified water-pipe system

In order to better highlight significant fluid-solids interaction aspects, in the very beginning of our approach we considered the geometry of a simple half-pipe, in which a water flow enters through a velocity inlet and exits through a pressure outlet. Initially, the pipe was 0.02 m in diameter and 1.20 m in length. The inner wall of the pipe acts as a solid boundary to the fluid flow but is dynamically coupled to the fluid domain only at a middle section of the overall geometry. In this section, the pipe is represented as a solid domain, 0.0005 m thick and 0.4 m long, with both ends fixed. The rest of the pipe is treated as a rigid boundary (the geometry is the one in Fig. 3). In Table 1 the settings of the physical properties of the solid are summarized.

For the physical modelling of the liquid, an user-defined density model was employed

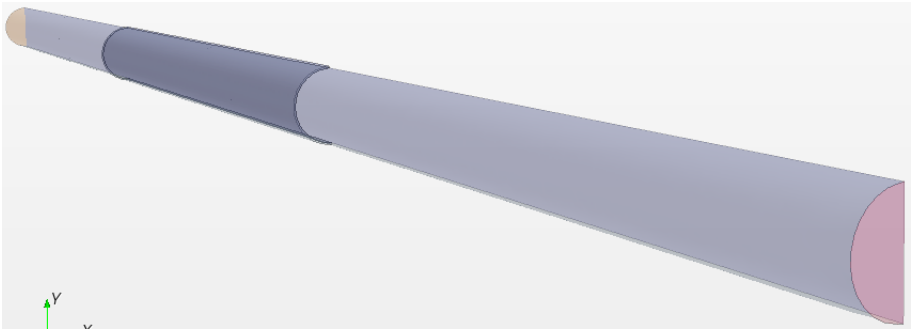


Fig. 3: Half - pipe geometry with a central solid section

Density	4500 kg/m^3
Poisson coefficient	0.34
Young's modulus	102.7 GPa

Table 1: Solid (Aluminium) properties

instead of the default constant density (incompressible) model, for reasons of numerical stability and accuracy. The interaction of a liquid with a solid is often more numerically stable if the liquid is allowed to be slightly compressible, particularly when the flow is internal to the structure. The density was hence defined using the following function:

$$\rho = \rho_0 + \frac{p}{c^2}, \tag{6}$$

where ρ_0 is a constant, p is the pressure and c is the constant speed of sound.

For the specific case considered in our analysis, the physical properties for the fluid are summarized in Table 2.

Speed of Sound in Water	200
Density Pressure Derivative	$1/(Speed \quad of \quad Sound)^2$
Density	$1000 + Pressure/(Speed \quad of \quad Sound)^2$

Table 2: Fluid (Water) properties

3.2 Solid displacement and fluid morpher boundary conditions

For this demonstrative case, two types of motions have been employed: **Solid displacement** and **Morphing**, both involving actual displacement of mesh vertices in real time. The fluid mesh representing the inner region is allowed to be deformed under an external load on the pipe (morphing), while the solid mesh representing the pipe must be allowed to be displaced in real time (solid displacement).

Particularly useful for cases of fluid structure interaction, the Solid Displacement motion, is used in conjunction with the Solid Stress physical model. This type of motion is characterized by the fact that it allows the solid mesh to be geometrically updated in response

to solid stress displacements, while the Solid Stress model enables the finite volume stress analysis.

Initially, the displacement of the solid section was obtained by applying a body load of 350 g to the pipe, hence increasing considerably the gravity value:

$$[0.0, -3433.5, 0.0] \quad m/s^2 \tag{7}$$

This value determined a suitable level of displacement, namely a maximum of 0.15 mm, as Fig. 4 shows. In this first part of the simulation, a deformed steady-state solution was

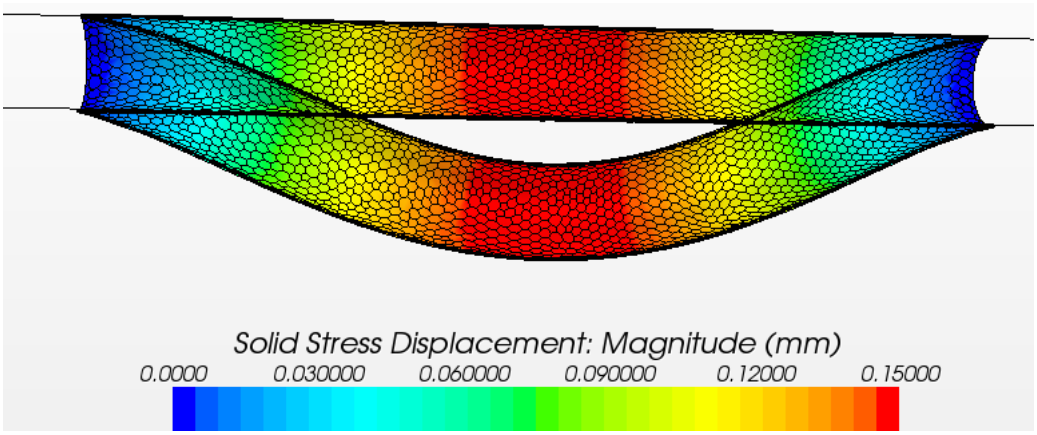


Fig. 4: Solid displacement 100 times magnified for the visualization

obtained, with a large time step. Then, the transient solution was obtained by removing the body load from the pipe and allowing it to vibrate in real time.

We wanted to impose a user-defined displacement to the wall of the pipe and verify if the fluid mesh inside the pipe was well coupled with the solid motion. In other words, the aim was to search for the conditions for a fluid mesh to "follow" well a solid motion, without degenerating or loosing quality.

When the Solid Stress model is enabled in the settings of a simulation, the **Stress/Displacement** boundary condition become available. This boundary condition allows to determine how the stress and displacement across the boundary will be specified. Hence, with this method we have the possibility to impose user-defined movement. Indeed, we defined an oscillatory motion for the solid section by means of the following vector field function:

$$\delta y = \begin{cases} 2 \times \sin t \times \sin^2(\frac{z}{l}\pi), & 0 \leq z \leq l \\ \\ 0, & elsewhere. \end{cases} \tag{8}$$

Here, l indicates the length of the solid section which extends from 0 to 0.4 meters in the Z-direction. With this imposed movement, a quite large displacement in the solid region is obtained and the fluid mesh correctly follows the solid one, as shown in Fig. 5.

Concerning the fluid region, it should be outlined here that by enabling the Morphing motion, the interface boundary between the fluid domain and the solid section of the pipe's wall will morph according to the solid stress displacement. In Table 3 we report the maximum

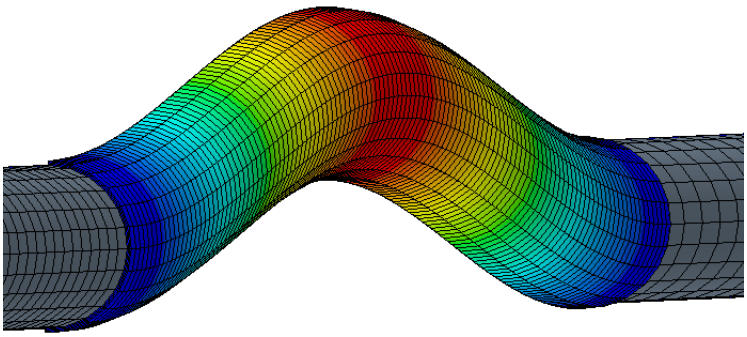


Fig. 5: Solid pipe deformation according to (8). The scalar field giving the color is the magnitude of the solid displacement, for a maximum value of 1 cm.

values for the displacements obtained with the two methods and also with different types of meshes.

Solid displacement	Cells	Body load (7)	Imposed (Rel. 8)
Structured-trimmed mesh	30300	$\delta y = 0.2 \quad mm$	$\delta y = 1 \quad cm$
Unstructured-polyhedral mesh	15000	$\delta y = 0.15 \quad mm$	$\delta y = 0.5 \quad cm$

Table 3: Maximum values for the solid displacement due to body load vs user-defined solid displacement (δy is the maximum reached displacement)

4 Morphing and rigid body motions

We then focused on the analysis of the behaviour of the fluid mesh when a solid piston was placed inside the water/pipe system. The aim at this point was to keep under control the expected deformations of the fluid mesh when the piston was subject to a periodical translation motion. It is very likely that problems of curvature and concavity arise in such applications, due to presence of curved surfaces, as the ones forming a cylindrical body.

4.1 Reviewed settings of simulation

With the above objective in mind we modified and scaled down the previous geometric model, considering a full pipe this time and giving to it the following dimensions:

- 0.5 m length and
- 0.1 m diameter for the external pipe;
- 0.2 m length and
- 0.02 m diameter for the internal piston;
- 0.2 the length and
- 0.001 m thickness for the solid section.

As we have seen, the stresses obtained in the solid section of the pipe during its displacement determined the morphing of the fluid inside the pipe. This was realised by means of

appropriate types of motion assigned to the regions (Solid Displacement for the solid section, respectively Morphing Motion for the fluid) and by means of appropriate boundary conditions.

In particular, for this model, the imposed displacement for the solid section was defined in the following form:

$$\delta y = \begin{cases} 0.01 \times \sin t \times \sin^2(\frac{z}{l}\pi), & 0.2 \leq z \leq 0.3 \\ \\ 0, & elsewhere. \end{cases} \tag{9}$$

A visible displacement (with a maximum magnitude $\delta y = 1$ cm) of the pipe's solid central section together with the morphing of the fluid mesh inside the pipe was obtained (as shown in Fig. 6).

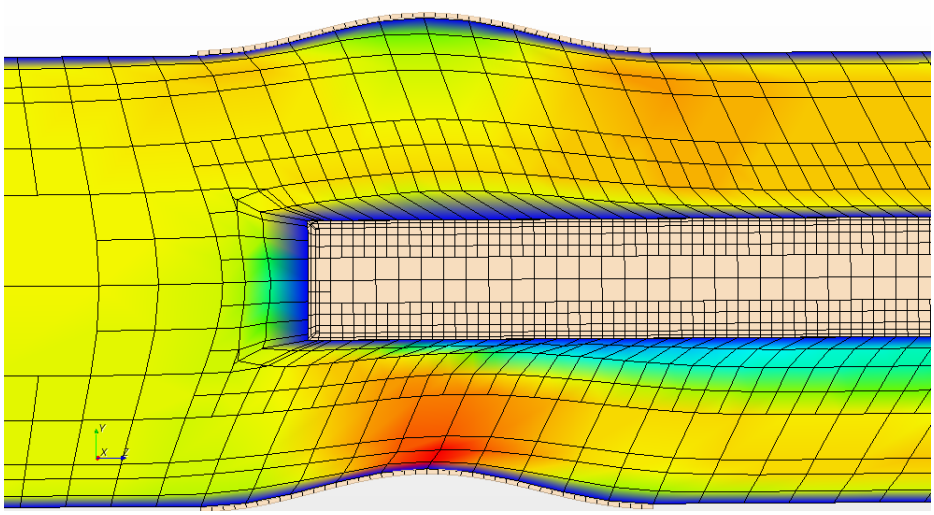


Fig. 6: Fluid mesh morphing provided by a Solid Stress calculation according to rel. (9), coupled with the piston translation movement

Furthermore, a rigid body motion, namely a Translation, was defined and assigned to the region representing the piston. In the following, the decisive settings in this sense are highlighted.

i) The boundary in the fluid region representing the interface between the fluid and the piston must obey to the **Solid Stress** morpher boundary condition (9) too.

ii) On the interfaces, the implicit coupling method for the fluid-structure interaction was enabled.

iii) The translation velocity of the piston was imposed by means of conditional expressions depending on the physical time of the simulation; in this sense, an example of field function used in the very beginning is the following one:

$$v_{translation}(m/s) = \begin{cases} +0.1, & 0.0s < t < 0.5s \\ -0.1, & 0.5s < t < 1s, \end{cases} \tag{10}$$

The displacement of the piston, obtained before the degeneration of the mesh determined the simulation to stop running, was equal to the diameter of the pipe (as shown in Fig. 7).

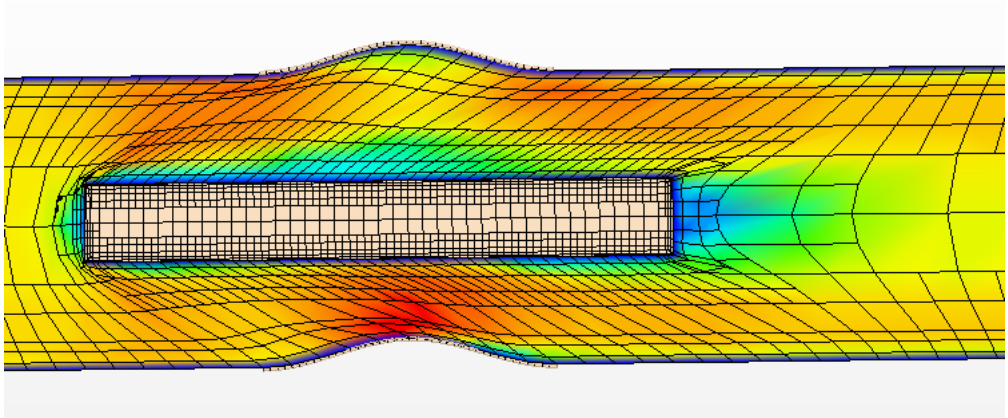


Fig. 7: Fluid mesh morphing provided by a Solid Stress calculation according to rel. (9) and deformation due to the piston translation movement defined in (10).

4.2 The re-meshing necessity and its automation. Metrics for the quality of the mesh

Once we managed to find the right settings for the simulation with an additional motion of a rigid body inside a fluid environment, it became clear that a large displacement would not have been possible without the re-generation of the volume mesh.

We started to take advantage of the object model contained within Star-ccm+'s Java client, fully available through the use of the Java macros. T

A macro is a Java program that is compiled and executed within the code's workspace, allowing the use of several programming constructs, such as loops and conditional constructs. In addition, the Star-ccm+ server contains a number of objects that we may instantiate and manipulate to carry out the desired sequence of tasks. The workspace may be used initially to record the principal actions we want to perform, then the resulting Java file may be edited and completed with the desired actions and the macro may be run as often as it is needed.

As expected, the recurring trouble in simulations employing morphing strategies is the appearance in the mesh, at a certain time, of negative volume cells.

Initially, in the simulation of the solid section's displacement and of the fluid mesh morphing, in avoiding this kind of error, keeping a concordance between the time step of the transient simulation and the amplitude of the displacement was sufficient. But, with the introduction of the moving solid piston this was not enough any-more. This is the reason why a closer examination of the quality (and of its measurements parameters) of the moving mesh became necessary.

In Star-ccm+ some important tools for the measurement of the mesh quality are available: skewness angle, face validity, cell quality metric. We give here a short description of each of them, with some sketches, as provided by the code's documentation.

Cell skewness angle metric (see sketch in Fig. 8): is the angle between the line connecting the centroids of two neighbouring cells and the orthogonal vector to the face that separates the cells. It should be kept minor then 90 degrees.

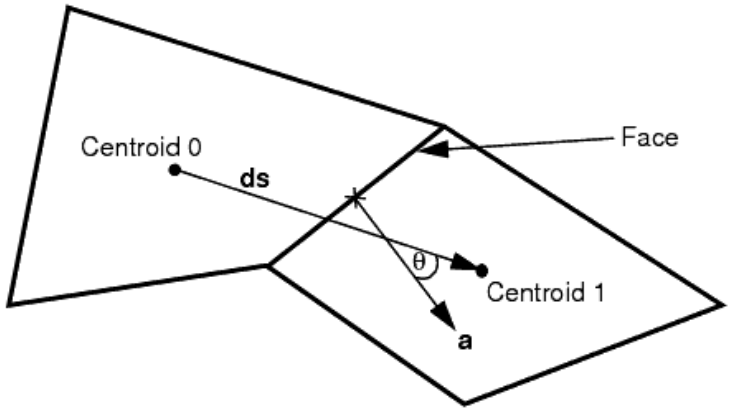


Fig. 8: Cell skewness angle description (reprinted from User Guide [9])

Face Validity metric (see sketch in Fig. 9): is an area-weighted measure of the correctness of the face normals relative to their attached cell centroid. For a good quality cell, the face normals must point away from the attached cell centroid; for a cell with bad face validity, one or more of the face normals will point towards the attached cell centroid. A face validity of 1.0 means that all face normals are properly pointing away from the centroid. Values below 1.0 mean that some portions of the face attached to the cell are not properly pointing away from the centroid, indicating some form of concavity. A value of below 0.5 means a negative volume cell.

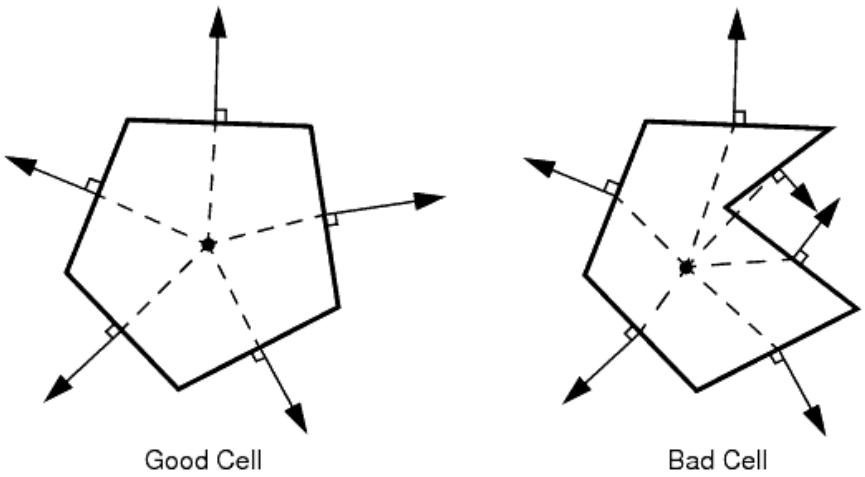


Fig. 9: Face Validity metric description (reprinted from User Guide [9])

Cell Quality metric (see sketch in Fig. 10): is a function that depends on the relative geometric distribution of the centroids of cells which are neighbours of a given face. It also

depends on the orientation of the cell faces. Generally, flat cells with highly non-orthogonal faces will have a low cell quality. A cell with a quality of 1.0 is considered perfect. A degenerate cell will have a cell quality approaching to zero.

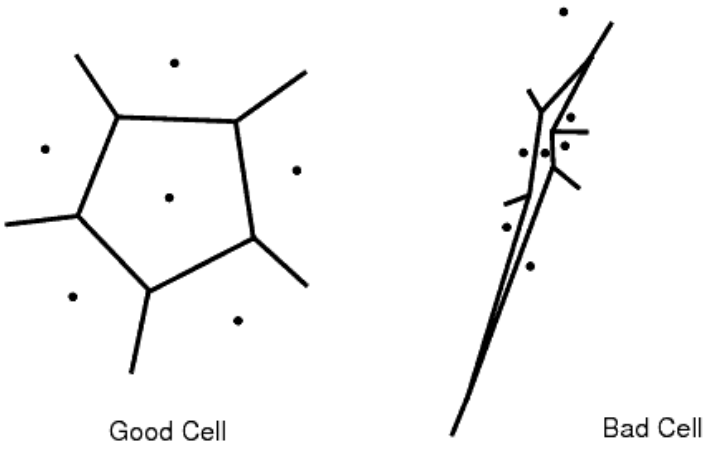


Fig. 10: Cell Quality metric description (reprinted from User Guide [9])

In Fig. 11 some cells with high skewness angle are shown, while in Fig. 12 two negative volume cells with a face validity value of 0.45 are illustrated.

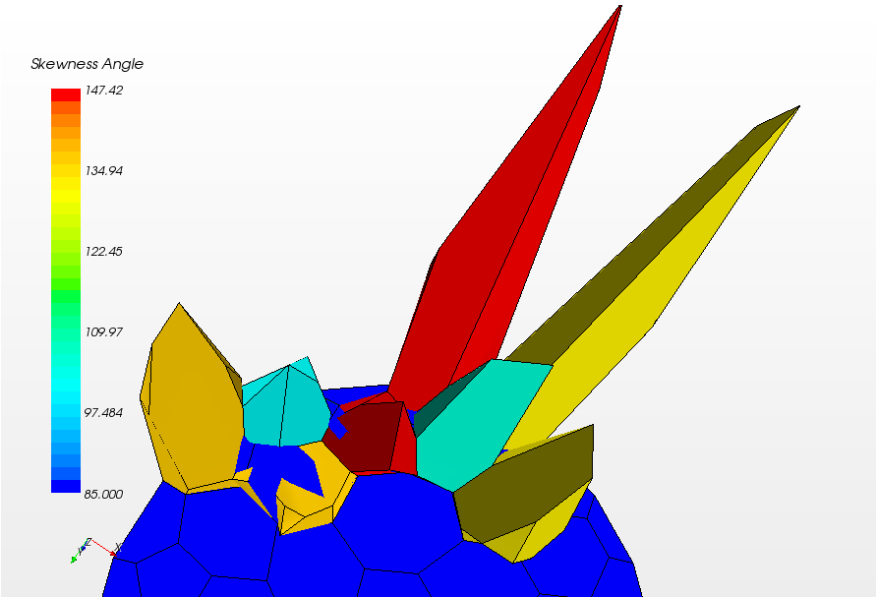


Fig. 11: Cells with high skewness angle laying on a piston's face

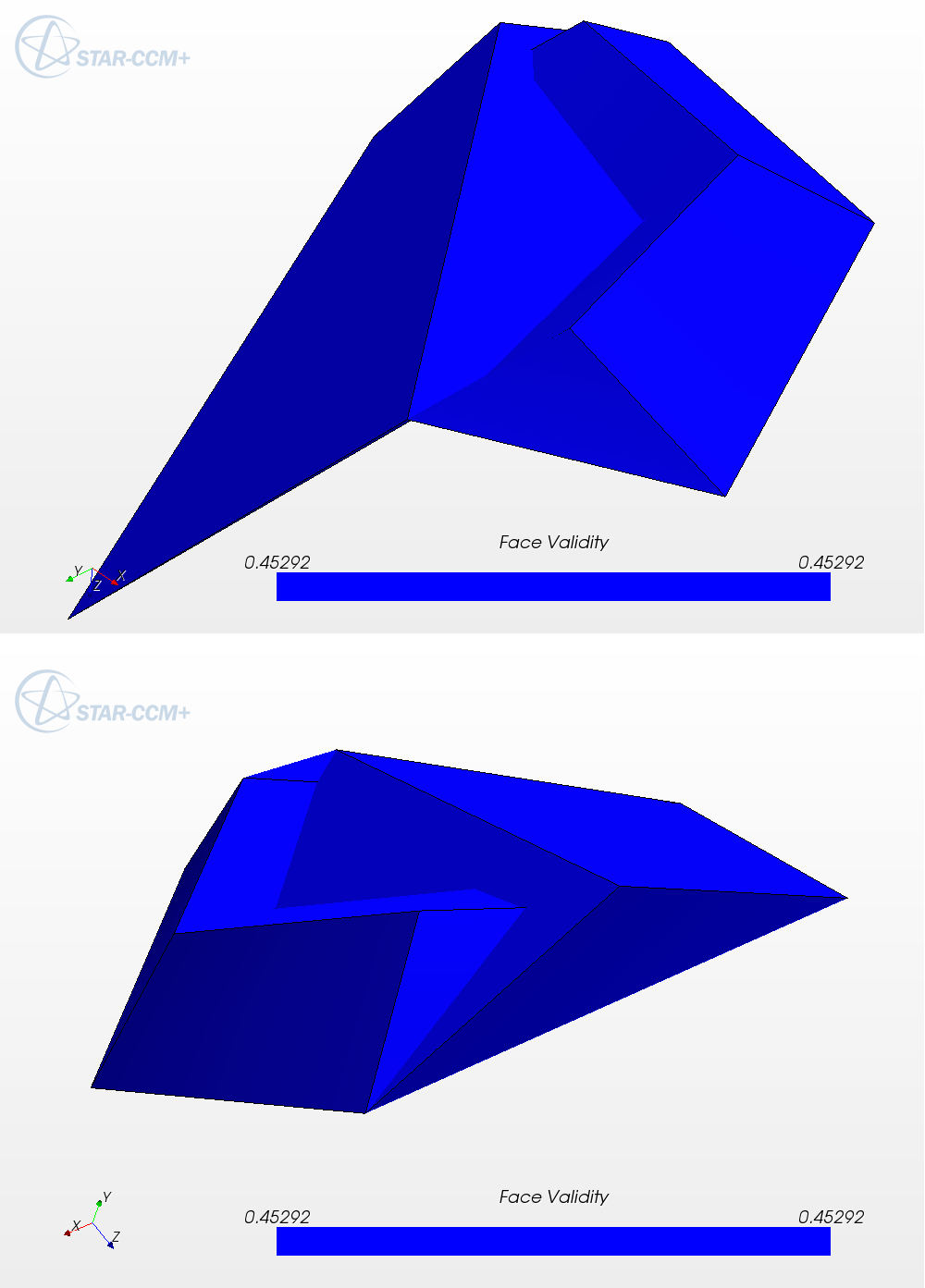


Fig. 12: Negative volume cells with low face validity

5 A study case for the development of some re-meshing strategies

The study case that we focused on took into consideration the geometric model already described in the Section 4.1. In addition, the translation velocity defined in (10) was pro-

gressively improved, passing from a discontinuous movement to a smoother one.

More exactly, a harmonic oscillatory motion was defined for the piston, with the translation velocity and the displacement vectors, respectively:

$$v(t) = [0, 0, -A\omega \sin(\omega t)] \tag{11}$$

$$\delta(t) = [0, 0, A(\cos(\omega t) - 1)] \tag{12}$$

The physical settings of the simulation with this motion enabled were:

- Maximum physical time for the simulation = 2 s
- Time Step TS = 0.001 s (or, for better visualization in animations, $TS = 10^{-4}$; the time step was chosen compatibly with the solid section displacement too)
- Period T = 2 s
- Frequency $f = \frac{1}{T} = \frac{1}{2}$
- Angular frequency $\omega = 2\pi f$
- Maximum piston displacement amplitude amplitude A = 0.04 m
- Fluid velocity in inlet = 0.5 m/s

As anticipated at the beginning of the Section 4.2, the simulation of an oscillatory motion of a cylindrical body interacting with a surrounding fluid inevitably required the re-generation of the volume mesh.

5.1 I^{st} re-meshing strategy: Surface boundary mesh extraction

The first procedure consisted in extracting a new surface mesh from the "almost" degenerated volume mesh in order to get an updated surface mesh as starting point for the new to be re-created volume mesh. Anyway, we noticed that the original feature curves of the piston weren't preserve with the generation of the new volume mesh. The consequence was that the volume mesh did not reflect any-more the exact original geometry, as the images in Fig. 13 show.

For this reason, after the extraction of the surface mesh and before the re-generation of the volume mesh, new feature curves have been added to the extracted surface mesh in order to maintain sharp edges and all the previous surface feature details in the surface remesher and in the volume mesh. Therefore, new edges for the piston region have been created based on an angle criterion.

Hence, as soon as the simulation got close to stop running for presence of negative volume cells, the following operations were executed, in order:

- a surface mesh was extracted from the deformed volume mesh;
- the initial and the re-meshed surfaces were deleted (to avoid duplication);

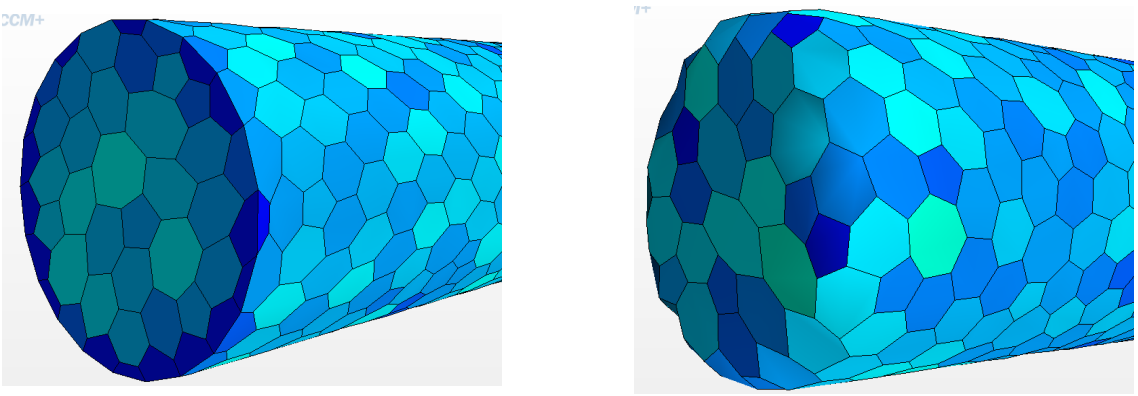


Fig. 13: Volume mesh of piston before and after surface mesh extraction

- new feature curves were added to the extracted surface mesh;
- the volume mesh was re-generated, taking the most recent boundaries as starting point (i.e. the extracted surface mesh).

This operations have been automated by means of a Java procedure that has been coupled with the software.

Remark 1 *It is an essential point that the re-meshing should be done before the mesh is actually degenerated.*

Based on this observation, the task of the macro Java was to control whether a given mesh quality criterion was satisfied or not and execute different instructions in basis of the evaluation of this criterion. Indeed, the macro executed the following instructions:

Algorithm 1

1. *Run the simulation for one time step.*
2. *Control the condition about the mesh quality criterion.*
3. *If the criterion is satisfied, turn back to step number 1. (i.e. take one time step forward), otherwise pass to instruction number 4.*
4. *Regenerate the volume mesh:*
   - *extract a new surface mesh from the deformed volume mesh;*
   - *delete the initial and the re-meshed surfaces;*
   - *add new feature curves to the extracted surface mesh;*
   - *re-generate volume mesh, taking the extracted surface mesh as starting point.*

Concerning the mesh quality parameters, at this stage, we principally used a minimum Face Validity (FV) criterion. The suggested minimum value is FV = 0.51, where a value of FV = 1 indicates perfect cells. However, we could notice during various test simulations that this minimum value didn't guarantee that cells reaching such a low level of quality do not degenerate into negative volume cells. For this reason, usually a minimum value of FV = 0.7 or 0.8 was imposed.

While playing the macro, we could notice that the number of re-meshing operations became progressively higher. In the first 0.1 s of the total solution time, one re-meshing was executed. In the next 0.9 s (until the first part of the translation was performed), from 3 to 6 regenerations every 0.1 seconds have been executed, with an increasing number when the translation velocity was reaching its maximum value, for a total of 35 re-meshing operations in one second, as reported in Table 4.

Table 4: Distribution of the automated re-meshing operations executed in 1 s (forward piston translation), accounted for every 0.1 s

Time (s)	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	< 0.7	<0.8	<0.9	<1	Total
Re-meshing	1	3	4	5	5	6	5	3	2	1	35

The volume mesh looked considerably modified as consequence of the too often re-meshing operations, especially in the piston region. In this region, the mesh suffered an overall thinning with respect to its initial form, as it is clearly visible in the images in Fig. 14[1].

5.1.1 Surface curvature

In the first tentative to avoid these changes in the mesh we made use of the code's tool named **Surface Curvature**.

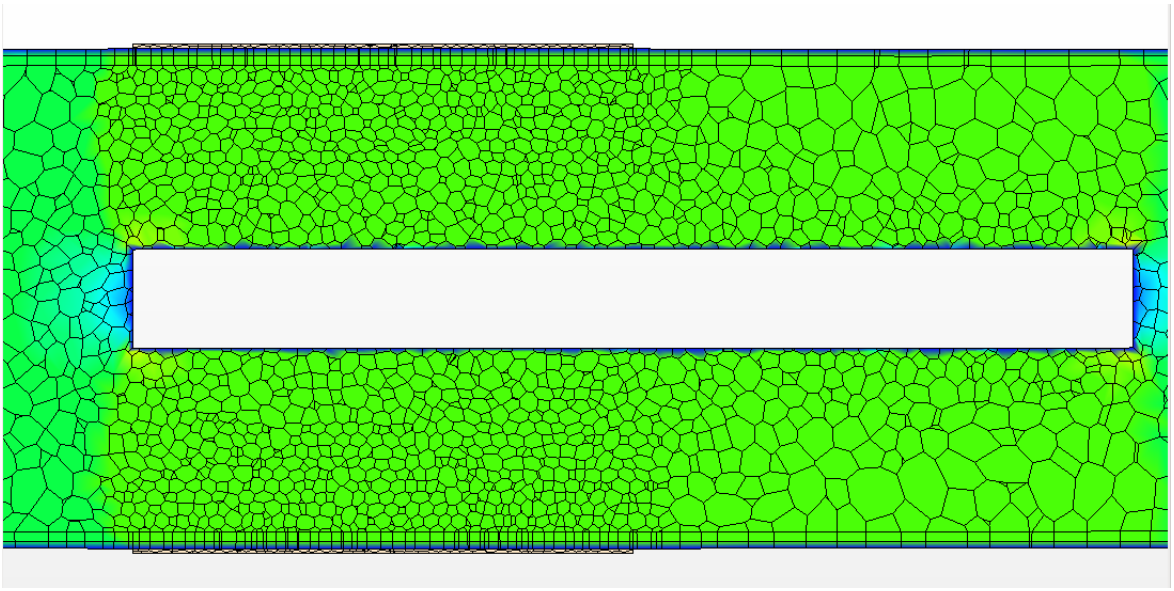
This tool allows cell refinement to be included for the surface remesher model based on the number of points around a circle. The default Pts/circle value of 36 indicates that approximately 36 triangles (for a surface) or cells (for a volume) would be used around a 360 degree cylindrical surface. Increasing the value would increase the relative refinement of the cells next to the curved surface. For pipes or tubes that have a relatively small diameter compared to the rest of the mesh region, decreasing the value to approximately in the range 16 to 20 may be required in order to avoid over-refinement of the area.

Initially, for the curvature the default value of 36 cells on a circle was maintained everywhere, including the smaller region of the piston. Then, we customised the surface curvature here with 16 cells in a circle. As consequence of this action we noticed that the number of the re-meshing operations decreased, reaching a maximum of 4 in 0.1 s, for a total of 20 in one second, as shown in Table 5.

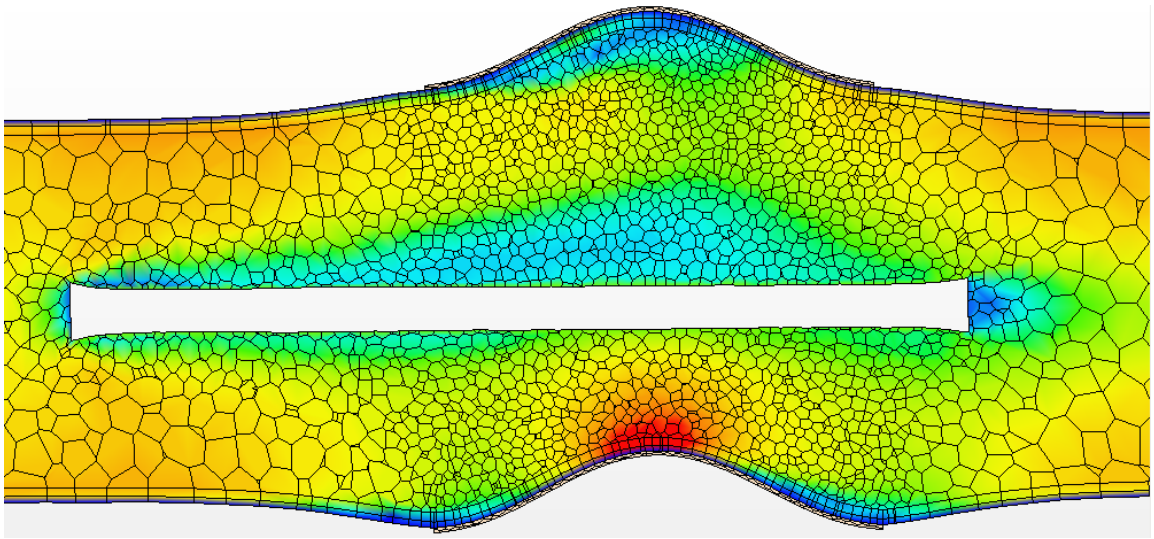
The volume mesh changed considerably also in this case, as it is clearly visible in the image in Fig. 15.

In order to have an idea of how much the diameter of the piston became smaller in confront with the original diameter we measured the area of a plane section cutting the piston. The measurement was done in both cases, with 36/16 Pst/circle, at the end of the first part

[1] An illustrative video may be provided on demand.



(a)



(b)

Fig. 14: Section in the initial volume mesh (a) and in the deformed fluid mesh (b)

Table 5: Distribution of re-meshing operations in 1 s (forward piston translation), accounted for every 0.1 s, for a value of 16 pts/circle for the Surface Curvature parameter

Time (s)	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1	Total
Re-meshing	0	2	2	2	3	4	3	1	2	1	20

translation. The measurements have confirmed that the thinning of the piston was indeed less in the second case. More exactly, denoted with ϵ the "error" given by the thinned surface area A^* with respect to the initial one A, i.e

$$\epsilon = 1 - \frac{A^*}{A}$$

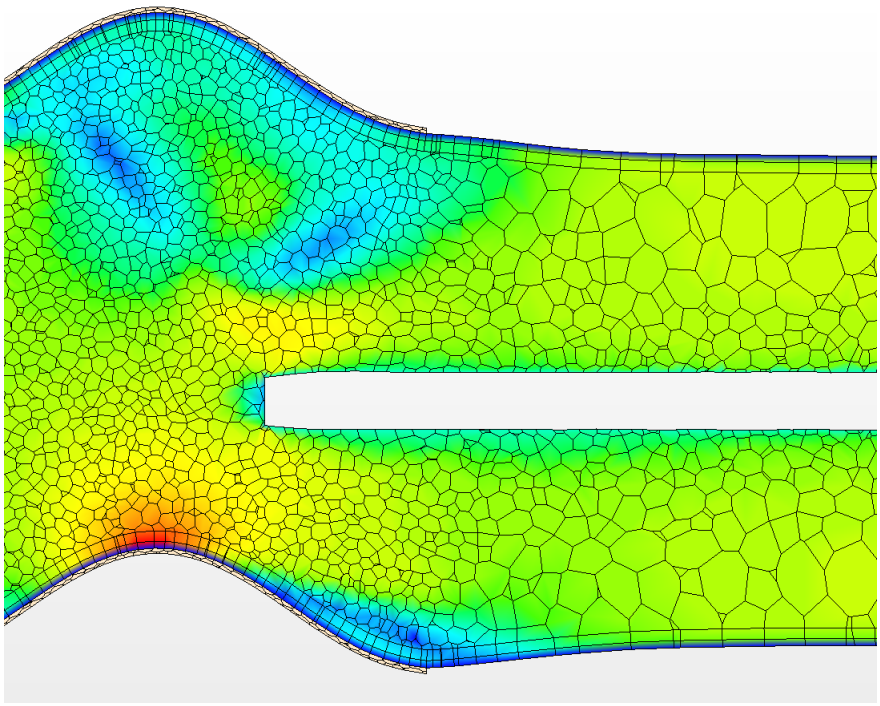
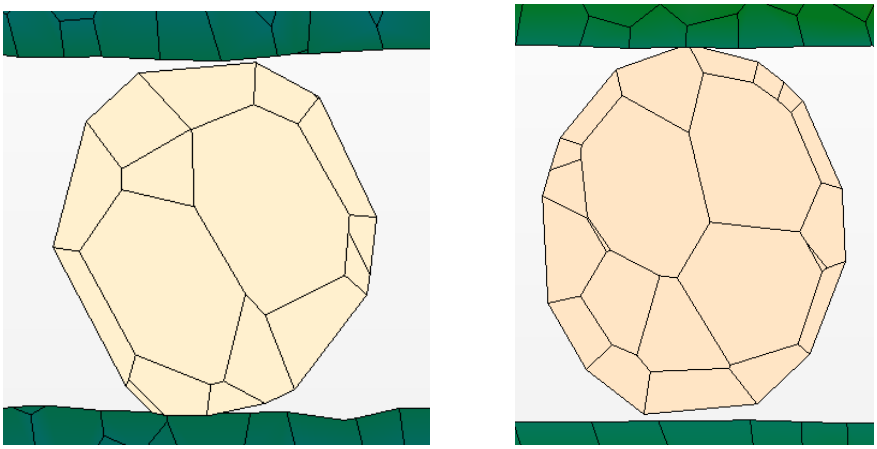


Fig. 15: Deformed mesh after 20 re-meshings

we obtained that

$$\epsilon_1 = 1 - \frac{4.7 \times 10^{-5}}{3.14 \times 10^{-4}} = 0.85$$

in the first case and $\epsilon_2 = 0.63$ in the second case (Fig 16).



(a) (b)

Fig. 16: The configuration after 35 (a) and after 20 (b) re-meshing operations of the piston volume mesh

Even if in equal periods of time the changes in the volume mesh were minor in the second case, it resulted clearly that the error was totally unacceptably.

In order to have a better idea about the "reduction" of the mesh in the piston region, we let the simulation to run forward and while the piston was performing its backward translation the changes in the volume mesh became even more significant. In Fig. 17 the initial mesh configuration on the plane section cutting the piston, accompanied by the same cutting section after about 50 re-meshing operations is shown, where the two pictures are captured on the same scale.

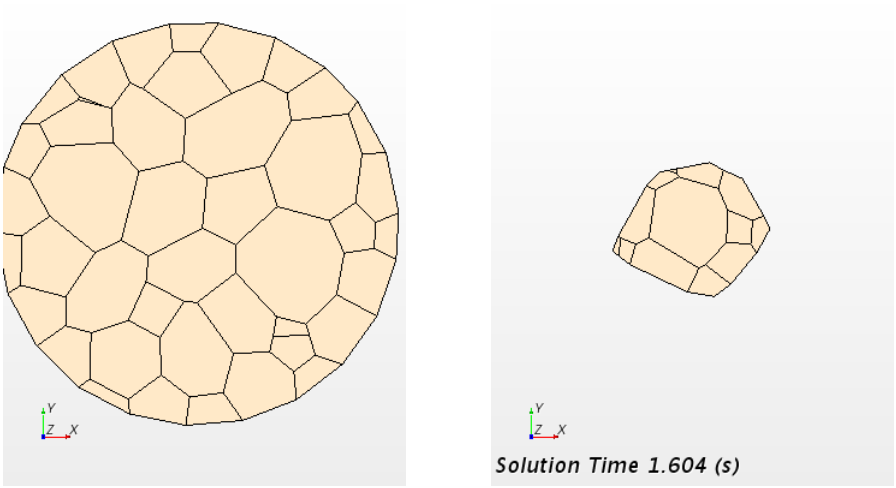


Fig. 17: Mesh configurations of a section cutting the piston at t = 0 s and t =1.604 s

5.1.2 Stationary solid section

A second approach consisted in running the simulation letting stationary the solid section, removing hence the solid displacement motion defined by the rel. (9). The minimum value for the Face Validity parameter was fixed to FV = 0.8 for the first part of the translation. However, due to the simplified physical settings, the face validity indicator could decrease until the minimum acceptable value (0.51). The situation of the re-meshing is reflected in Table 6 and the final volume mesh representation is shown in Fig. 18[2].

Table 6: Number of re-meshing operations during the complete translation of the piston

Time (s)	<0.1	<0.2	<0.3	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9	<1	Total
FV>0.8	1	2	4	5	5	5	5	4	2	0	33
Time (s)	<1.1	<1.2	<1.3	<1.4	<1.5	<1.6	<1.7	<1.8	<1.9	<2	Total
FV> 0.5	0	1	5	3	4	3	3	2	1	0	22

In Fig. 19 the initial mesh configuration of the plane section cutting the piston and the configuration at the end of the translation, after the 55 re-meshing operations, on the same scale, are shown. However, neither this tentative changed the result of the first strategy of re-meshing adopted. It only confirmed that problems of stability of the simulation should not be encountered any-more.

[2]An illustrative animation of the re-meshing in this case may also be provided on demand.

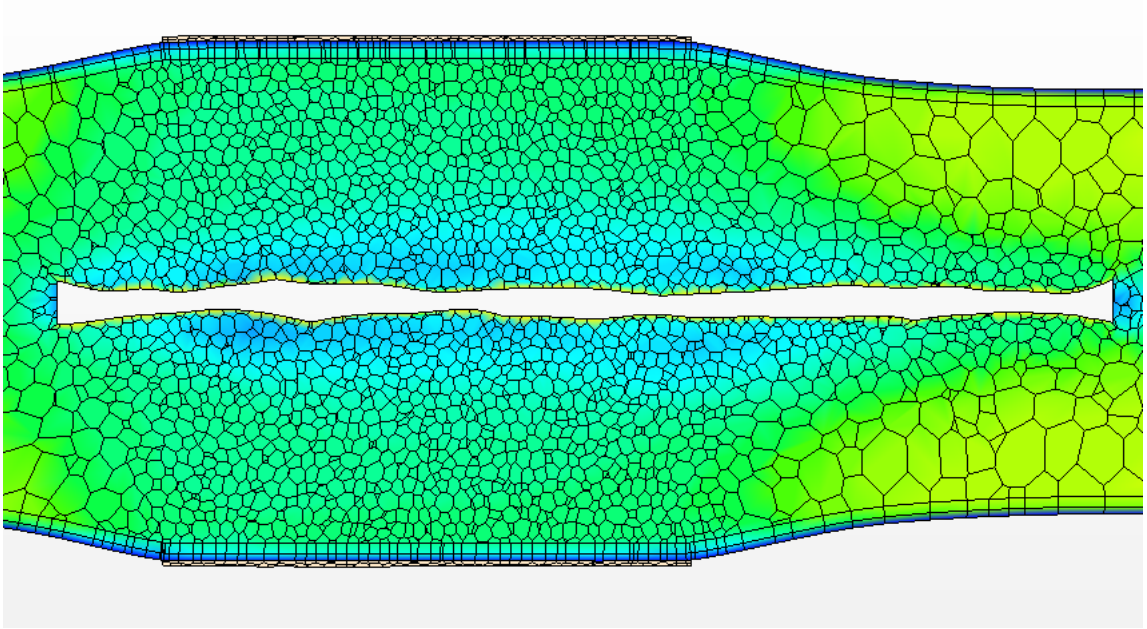


Fig. 18: Fluid and piston's volume mesh deformation after 55 re-meshing operations

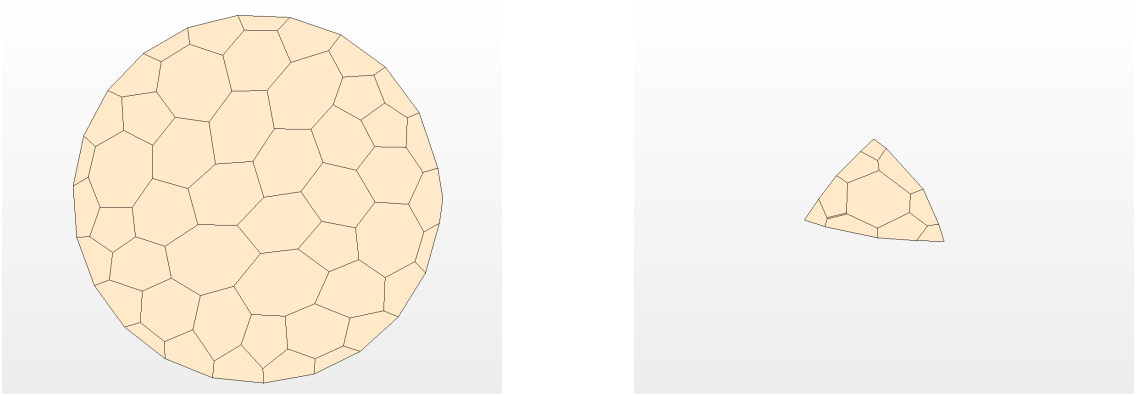


Fig. 19: Mesh configurations of a section cutting the piston at t = 0 s and t = 2 s

5.1.3 Conclusions and discussions about the surface mesh extraction strategy

By adopting a strategy of surface extraction we could notice that the initial geometry of the model (especially the one of the piston) didn't remain identically the same in the volume mesh representing it, after the re-meshing operations. Indeed, in the piston region significant differences between its initial geometrical form and its final mesh representation, obtained after an oscillation, verified.

In conclusion, we can say about this strategy that the reasons of its failing arise from the cylindrical form of the piston. The curved surfaces forming the cylindrical walls of pipes and pistons are constructed starting from two feature curves only (the two extreme circles). In such constructions, the feature curves are not sufficient to ensure the preservation of the original form of the object while undergoing re-meshing operations determined by physical requirements. Those bodies formed with planar faces like rectangles or triangle would not

have this problem at all because all the faces would have their own feature curves and these would be preserved without any problem. We have chosen this type of object in our study case for its complexity, knowing that bodies with planar faces would have not given rise to such problematic aspects.

Remark 2 *The fact that during the procedure of re-meshing upon the surface and volume mesh, the initial form of the geometry was not preserved, indicated that we should have considered a different approach and integrate and/or substitute the above operations with actions intent on preserving the geometry.*

5.2 II^{nd} re-meshing strategy: CAD geometry update

For the reasons indicated in the previous conclusions about the first strategy adopted, it has been decided to let out the previous operations of extracting a new surface mesh and new feature curves needed for the regeneration of the volume mesh. Instead of these operations, we acted directly upon the geometry, performing an Update operation at the geometric CAD modeller level.

For the previous changes made to the geometric model to take effect in the rest of the simulation, the 3D-CAD model and the geometry parts had to be updated, the new surface created with the update operation had to be re-meshed and the volume mesh regenerated.

The macro was modified removing the part of surface extraction and inserting a CAD model translation. The translation was given by a vector having the same magnitude as the displacement of the piston due to the translation motion imposed at region level:

$$Translation = [0, 0, dZ], \tag{13}$$

where dZ is determined from rel. (12)

Remark 3 *We outline here that the Translation Vector giving the updated position of the piston body was contemporaneously exposed as a design parameter in the CAD model.*

When the Face Validity parameter which controls the level of degeneration of the mesh reaches the minimum value of 0.8, we get the value of the displacement performed by the piston until that moment and assign this value to the translation vector exposed as design parameter with the task to update the initial position of the piston to the one reached by the correspondent boundary. In this manner, the geometry is being updated and perfectly preserved every time the degeneration of the mesh due to the piston movement through the fluid requires a new re-meshing.

Algorithm 2

1. *Run the simulation for one time step.*
2. *Control the condition about the mesh quality criterion.*
3. *If the criterion is satisfied, take one time step forward; otherwise:*
4. *Regenerate the volume mesh, including:*

– *at CAD construction level, apply a translation to the piston body and expose as design parameter the translation vector;*

– *read the displacement's value from a report and put it into a variable, dZ;*

– *assign the value of dZ to the translation vector, updating the piston's initial position;*

– *regenerate the surface and volume meshes, taking the most recent boundaries as starting point (i.e. the geometry with an updated piston position).*

At the same time, as the 3D-CAD solid modeller is fully integrated into the Star-ccm+ environment, the previously obtained solution will automatically be mapped onto the new geometry.

In the sequence of images in Fig. 20 some cells are illustrated, firstly of good quality, then near to the degeneration and then after the re-meshing operation[3].

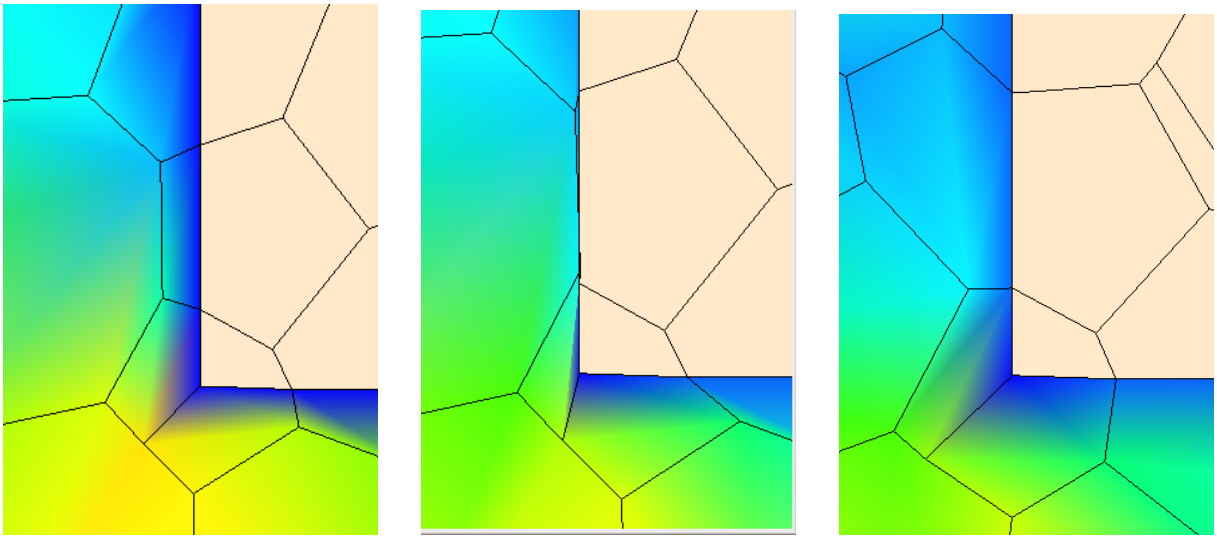


Fig. 20: Left - good quality, center - degenerated, right - regenerated cells

The number of re-meshing operations proves to be high also in the case of this strategy, due to the high minimum value for the control parameter (FV = 0.8), but the geometry of the model is not affected any-more and it is fully preserved together with its feature curves.

Some observations should be made about the variation in time of the Face Validity parameter. We noticed that the value of this control parameter measuring the mesh quality was decreasing very quickly from one iteration to another one, due to its discrete nature (in Fig. 21, 5 re-meshing executed in 0.1 s are shown).

For this inconvenient and in order to reduce the number of the re-meshing procedures, the employment of a better metric should be further taken into consideration.

[3]Animations may be provided on demand, with a general view of the translation motion of the piston and with a zoomed visualization where the degeneration and the re-generation of the mesh is clearly highlighted.

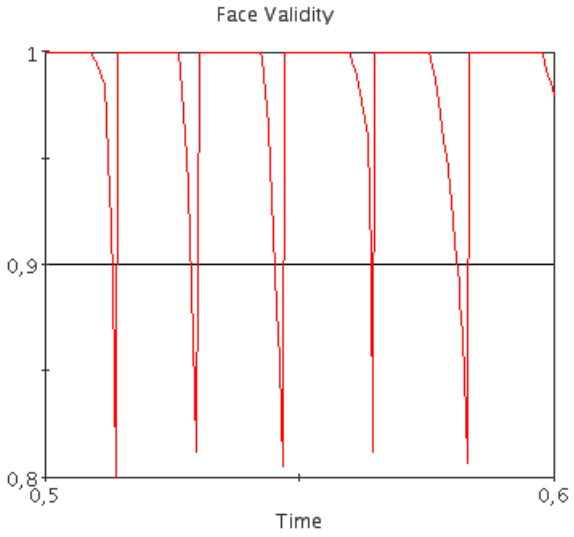


Fig. 21: Evolution of the Face Validity parameter

5.2.1 Simplified model - better mesh quality metric

We constructed from scratch a simplified CAD model: one single body/region with a boundary representing the piston, hence without keeping the physical solid region of the piston. Simple settings have been given: Morphing motion for the region, with two possibilities of morpher boundary conditions (MBC) for the boundary representing the piston:

- Grid Velocity (z-component)
$$v_z(t) = A\omega sin(\omega t) \tag{14}$$
- Displacement (z-component)
$$\delta_z(t) = -A(cos(\omega t) - 1) \tag{15}$$

The translation vector imposed as design parameter for the updating of the CAD model takes its value from a report that calculates the effective displacement of the piston with the rel. (15).

As anticipated, the mesh quality parameter employed for the control was not the Face Validity anymore. Instead, the **Cell Quality** metric was used, with a significant reduction of the number of re-meshing operations.

Some interesting observations should be highlighted here, observations derived from the various test simulations that we performed.

Remark 4 *With the **Displacement** MBC on a polyhedral mesh, the translation motion of the piston is performed without the need of re-meshing; the minimum cell quality is about 0.001 and is still good enough for the running of the simulation; on a trimmed mesh instead, the mesh quality degenerates very quickly.*

Remark 5 *With a **Grid Velocity** MBC, the minimum acceptable value for the Cell Quality parameter is 0.01; this value is used as control parameter by the Java macro: when the cell quality is less then this value, the volume mesh is re-created.*

From the two experimented MBC, we choose the Grid Velocity one, because of its compatibility with the physics involved.

Three illustrative images are given in Fig. 22, captured firstly at the beginning of the simulation, then nearby the degeneracy of the cells and after the application of the re-meshing[4]. Note that, in these images the variations of the Minimum Cell Quality during these three key moments are reported.

5.3 III^{rd} improved strategy: optimized re-meshing

In order to further reduce the number of re-meshing operations and to optimize the mesh regeneration, a third improved strategy was developed.

The idea was to divide the domain into three regions: a central one containing the piston boundary and two lateral regions. In the CAD construction, the three regions have been obtained by subtraction and intersection boolean operations, followed by imprinting operations of the three regions for a perfect conservation of their contacts (interfaces).

A Translation motion was given to the central region, via a velocity vector with the z-component given by rel. (14) and a Morphing motion was given to the other regions.

The boundary conditions for the morpher were:

- **Floating** for the Wall type boundaries. Points on the Floating boundaries have no boundary condition, hence they move wherever the other control points dictate (i.e the control points lying on two interfaces)
- **Grid Velocity** for the interfaces with the central region. The grid velocity value is the same as the translation velocity vector in (14).

The most significant improvement of this strategy consisted in the fact that in order to regenerate the mesh, the only lateral bodies had to be up-dated and hence the re-meshing was required only for these bodies. Initially, the updating was done by performing a translation operation on the lateral bodies during the CAD construction and imposing the translation vectors defined as in (15) as design parameters.

5.3.1 Advantages, drawbacks and improvements

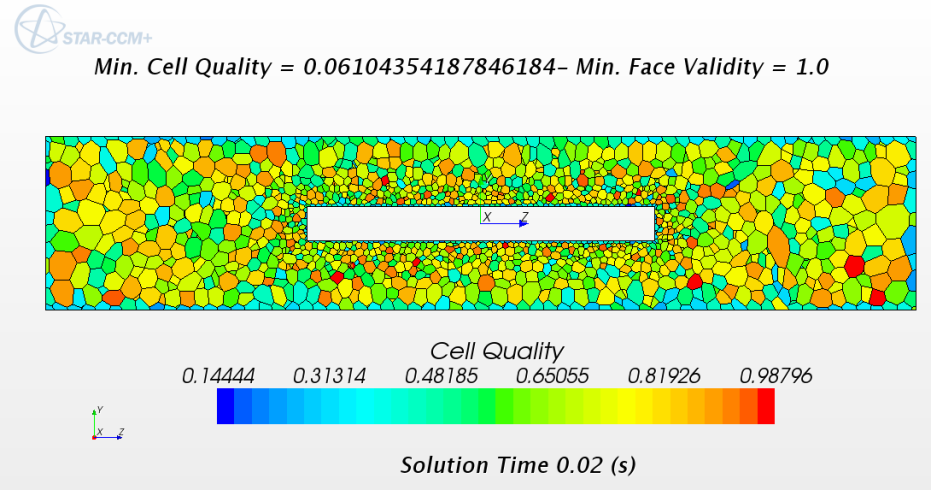
This approach offered some important advantages:

- the re-meshing took much less time, since the mesh in the piston's region remained unchanged;
- the solution didn't loose accuracy, since it didn't have to be interpolated in this region.

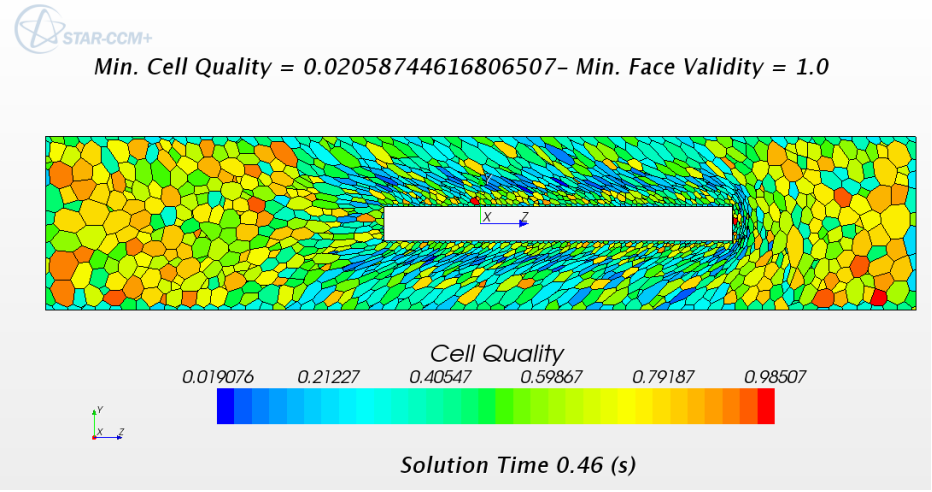
However, two anomalies were identified.

On one hand, we noticed that the displacement calculated with the rel. (15) and imposed as design parameter for the updating translation vector was not equal to the effective displacement performed by the central region containing the piston according to the rel. (14).

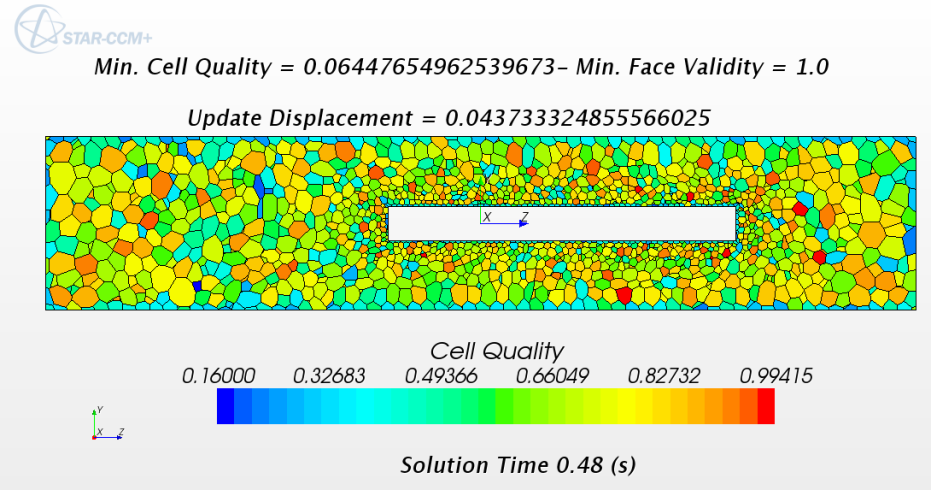
[4]An illustrative animation of the strategy may be provided on demand



(a)



(b)



(c)

Fig. 22: Good quality (a), degenerated (b), regenerated (c) cells

Since the displacement was obtained by intregration of the velocity, in a simulation with a large time step, it is very likely to obtain this kind of errors.

The bad topological effects coming out while the simulation was running and the volume mesh was automatically re-generated consisted in the fact that the CAD model didn't maintain the imprinting of the interfaces between the three regions. The consequence of these topologically inconsistent procedures was that after a few re-meshings the simulation stopped running with serious errors.

Hence, instead of the displacement's value calculated with (rel. 15), we used a measurement of the arrival position of the piston: we measured the maximum position p_1 reached by the piston's region, subtracted from it the initial position p_0 and obtained in this manner the actually performed displacement, $\Delta z = p_1 - p_0$.

On the other hand, we noticed that the geometric parts were displaced in space due to the translation operation, while the inlet and the outlet boundaries should have remained fixed, in their initial position.

For this reason, we gave up the translating bodies and adopted as design parameters controlling the variation of the geometry parts the **extrusion distances** used in the construction of the central part. The cylindrical bodies are usually constructed starting from circles that are then extruded with a certain length. In the 3D-CAD modeller included in Star-ccm+, the directions of extrusion are quite flexible, so we could choose a two asymmetric way of extruding the circular sketch.

If we denote the initial extrusion distances (symmetric and asymmetric) of a cylindrical body with d_0, d'_0, then the extrusion distances controlling the updating process will be:

$$d = d_0 + \Delta z \tag{16}$$

$$d' = d'_0 - \Delta z \tag{17}$$

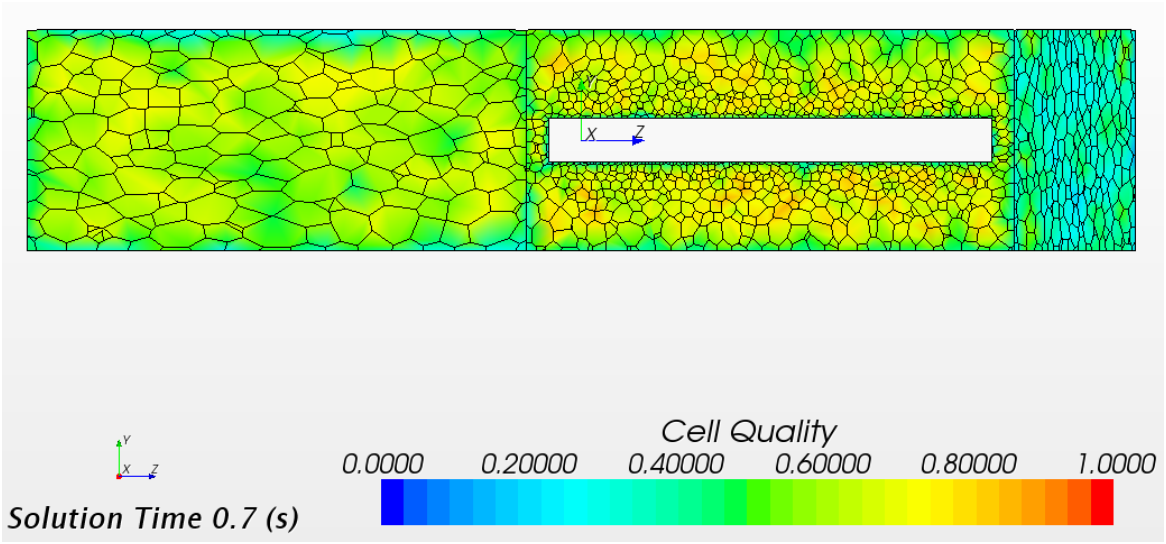
The implementation may be formulated into the following:

Algorithm 3

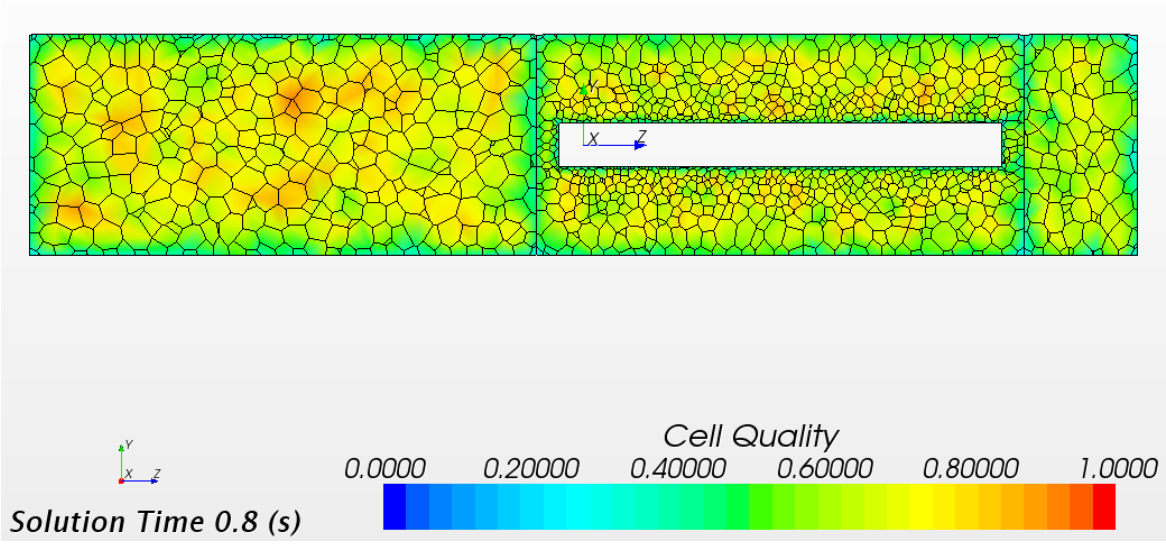
1. *Run the simulation for one time step.*
2. *Control the condition about the mesh quality criterion.*
3. *If the criterion is satisfied, take one time step forward; otherwise:*
4. *Regenerate the volume mesh, including:*
   - *at CAD construction level, expose the extrusion distances as design parameters;*
   - *read the effective displacements performed by the moving parts and put them into variables;*
   - *assign the value of d and d' to the symmetric, respectively the asymmetric extrusion distances;*
   - *re-generate the surface and volume meshes, taking the most recent boundaries as starting point (i.e. the geometry with an updated piston and central body position).*

In Fig. 23, the Alg. (3) is illustrated on two images[5].

Remark 6 *The relations (16) and (17) were applied both to the external domain and to the internal piston boundary surface.*



(a)



(b)

Fig. 23: Degenerated (a)/regenerated (b) cells in lateral bodies

5.3.2 Conclusions and discussions about the up-dating geometry strategies

Choosing a two-asymmetric way for the extrusion distances used in the construction of the cylindrical bodies allowed for a correct CAD model updating, consistent with the translation/morphing motions imposed to the regions.

[5]For a better understanding, an animation may be provided on demand.

The cells in the central region containing the piston were not deformed anymore, hence the volume mesh of this region didn't have to be re-created.

Choosing to get the exact displacement's values of the moving parts by means of an effective measurement of their actual position allowed for the contacts between all the regions to be perfectly preserved and the translational periodical motion to be obtained in a less expensive manner.

6 Analysis of the compatibility of the morphing and the re-meshing techniques with the physics

The further work consisted principally in getting the control of the morphing and re-meshing techniques when they were related to specific physical requirements. We analysed two particular situations.

In the first one, the three regions described previously were immersed into a larger fluid domain, with interfaces being created between the four regions. The intention was to analyse the impact of the morphing/re-meshing procedures on the development of the velocity field of a fluid flow.

In the second one, we considered a more complex geometry containing a complex internal structure, performing a translation motion and we analysed the impact of the morphing/re-meshing procedures on the drag force that the fluid exerted on the surface of the internal structure.

In the following subsections, we shall describe the two cases and discuss the results concerning the compatibility of the techniques previously developed with the simulated physical fields (force, pressure, velocity).

6.1 The coupling of the moving parts with larger domains

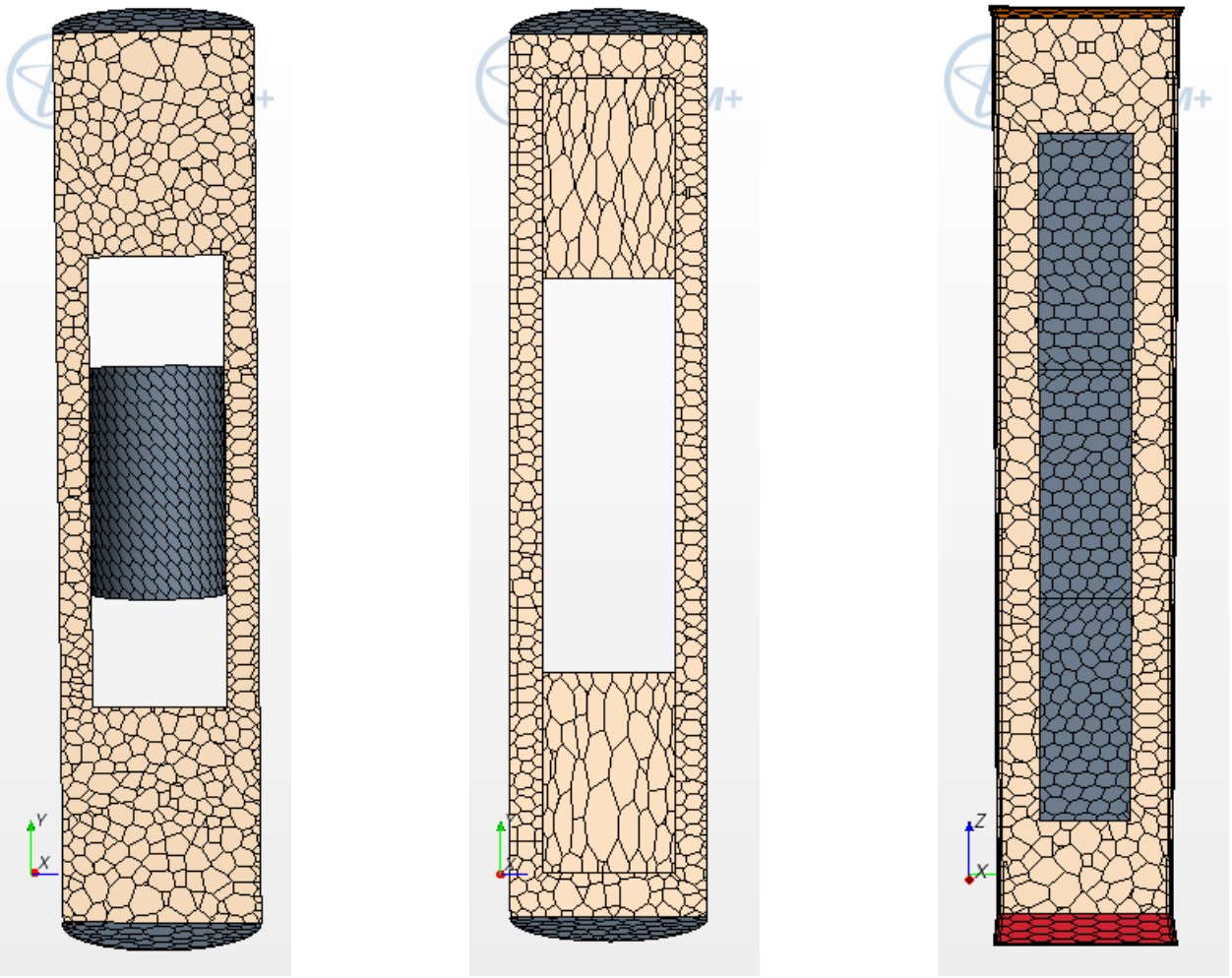
As a conclusive and applicative part of our work, our further objective was to analyse the behaviour of the partial (In-Place) interfaces, that is, of those topological constructs allowing communication between different regions of the solution domain. Since the interfaces should have been sliding in tandem with the moving domain, it was essential to correctly combine them compatibly with a flow entering through a velocity inlet and exiting through a pressure outlet of the larger domain without being perturbed by the internal translational motion of the piston or by eventual re-meshings.

The geometry of the new model was constructed gradually, starting firstly with a single internal region sliding inside the larger one, then with two regions and finally with three regions.

Cylindrical bodies, as already considered before, as well as rectangular and hexagonal bodies have been employed, as illustrated in the Figs. 24 and 25.

Similarly to the cases previously considered, a translation motion was given to the central region, via a velocity vector with the y-component given by

$$v(t) = [0.0, A\omega sin(\omega t + \phi), 0.0] \tag{18}$$



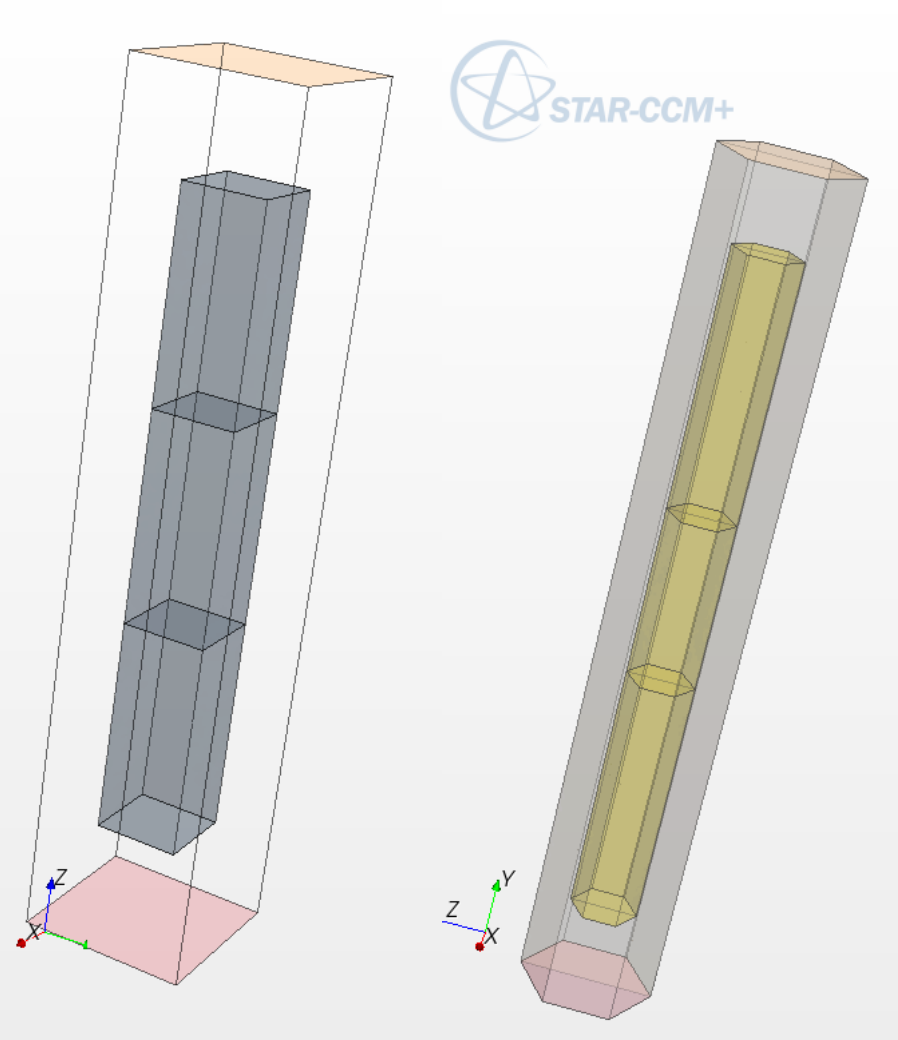
(a) (b) (c)

Fig. 24: One (a), two (b), three (c) sliding bodies geometry

and a Morphing motion was given to the lateral regions, while the rest of the domain remained stationary. The morpher boundary conditions were:

- **Floating** for all the sliding interfaces;
- **Grid Velocity** (the same in (18)) for the central region interfaces with the up and down neighbouring regions
- **Fixed** for the up and down regions interfaces with the external domain.

In order to re-generate the mesh, if necessary, the only lateral bodies had to be updated and hence the re-meshing had to be done only for these bodies. As before, for the updating of the geometry, we measured the maximum arrival position reached by the central region (by retrieving it from a maximum type report having the Position as field function), subtract from it the initial position and obtain in this manner the actually performed displacement, Δz.



(a) (b)

Fig. 25: Rectangular (a), hexagonal (b) sliding bodies geometry

The design parameters controlling the variation of the geometry parts were again the **extrusion distances** used in the construction of the central part.

The implementation was the following one: if we denote the initial distances (symmetric and asymmetric) with d_0, d_0', with p_0, p_1 the values of the maximum initial, respectively, arrival position, then the extrusion distances controlling the updating process will be given by:

$$d = p_0 + \Delta z = p_0 + p_1 - p_0 = p_1 \tag{19}$$

$$d' = p_0 - \Delta z = p_0 - p_1 + p_0 = 2p_0 - p_1 \tag{20}$$

Remark 7 *In the Java macro that controls the mesh regeneration, the CAD Update was implemented in the following manner:*

```
double d = maxReport_0.getReportMonitorValue();
ExtrusionMerge extrusionMerge_0 = ((ExtrusionMerge)
cadModel_0.getFeatureManager().getObject("Extrude 2"));
extrusionMerge_0.getDistance().setValue(d);
extrusionMerge_0.getDistanceAsymmetric().setValue(2.3-d);
```

As easily predictable, while considering cylindrical bodies, the matching between the interfaces was not perfectly done, due to the presence of the curvature in the surfaces forming the bodies, but the simulation didn't suffer and the flow was not perturbed, in conditions of refined mesh and small time step.

With rectangular bodies, there was no problem of interfaces matching and the flow was not perturbed at all. We tested this situation also by considering lateral inlet/outlet boundaries, hence giving a transversal direction to the flow with respect to the direction of the translation motion and to the interfaces.

After getting the control of the moving parts in these testing cases, hexagonal bodies have been considered. The sliding interfaces have been created without further problems. Oscillatory motions of the internal body have been obtained with amplitude, period and phase of the oscillation being changed from case to case.

The perturbation of the flow was evaluated by means of a measure of the flow velocity. Initially, for a few time steps, the simulation was run in stationary case. The fluid flow evolved normally until its complete establishment. Then the transient case was set, with the two motions getting started. The fluid flow didn't suffer any significant "shock" and the perturbations have slightly registered. The quality of the mesh was monitored by means of the Cell Quality metric. The regular development of the flow, illustrated by means of plots inserted in the scalar scenes showing the velocity field (Fig. 26) is shown[6].

The periodical motion of the internal body with the variation of the mesh quality before and after mesh re-generation (Fig. 27) is shown[7].

6.2 Application of combined strategies to complex geometries

In order to further study the compatibility of the morphing/re-meshing techniques with the physics, we applied these techniques to the case of a real model, a small part of an experimental nuclear facility. In this sense, we performed an useful exercise of simplifying complex CAD constructions provided by experienced engineers in order to reproduce the essential parts for our specific CFD analysis.

Hence, we constructed from scratch a new geometry that should serve for the simulation of the movement of a control rod system inside its guide tube. The geometry of the model was constructed as usually, with the 3D-CAD modeller included into the Star-ccm+ code, taking advantage of many of its features:

- the design parameters (like segments lengths, translation vectors, extrusion distances);

[6]Animations illustrating the regular development of the flow may be provided on demand.

[7]Animations illustrating the periodical motion of the internal body with the variation of the mesh quality may be provided on demand.

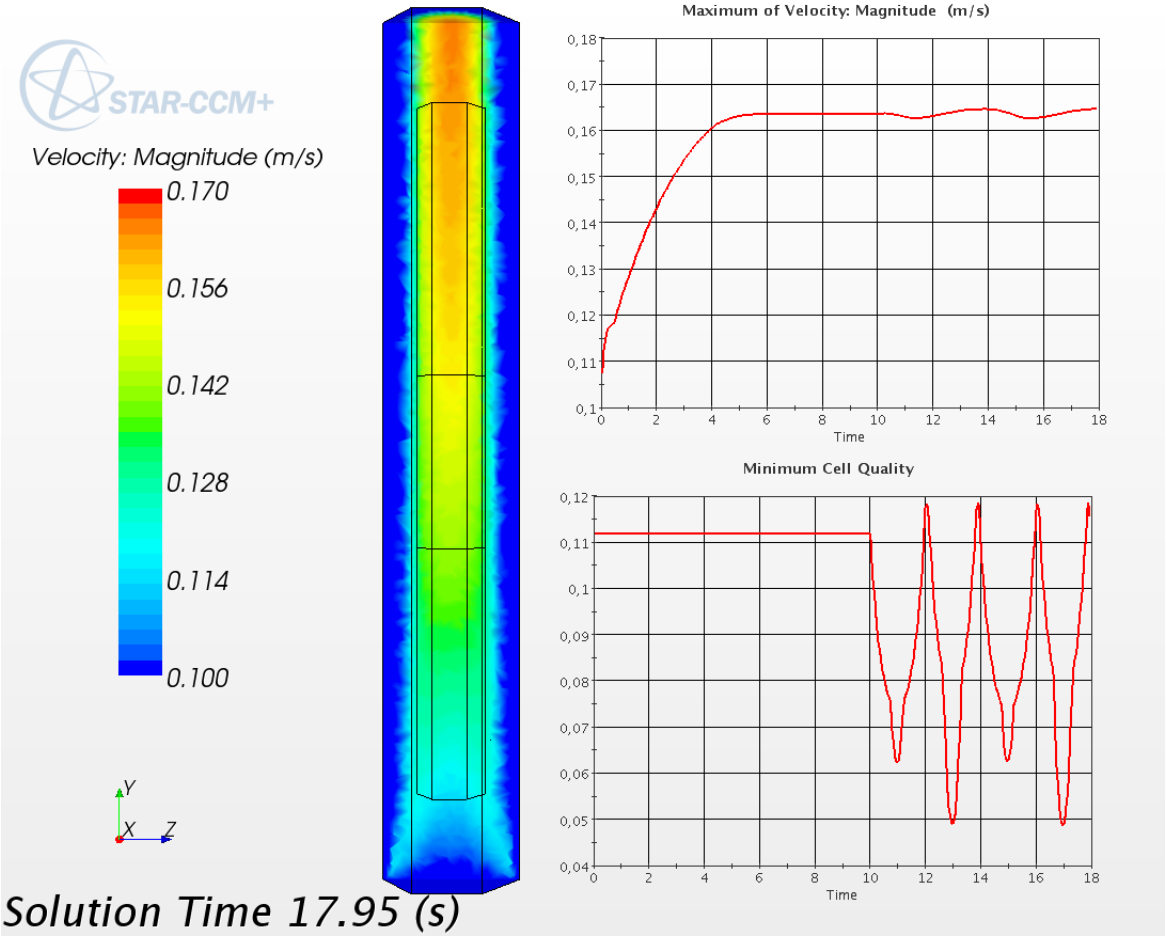


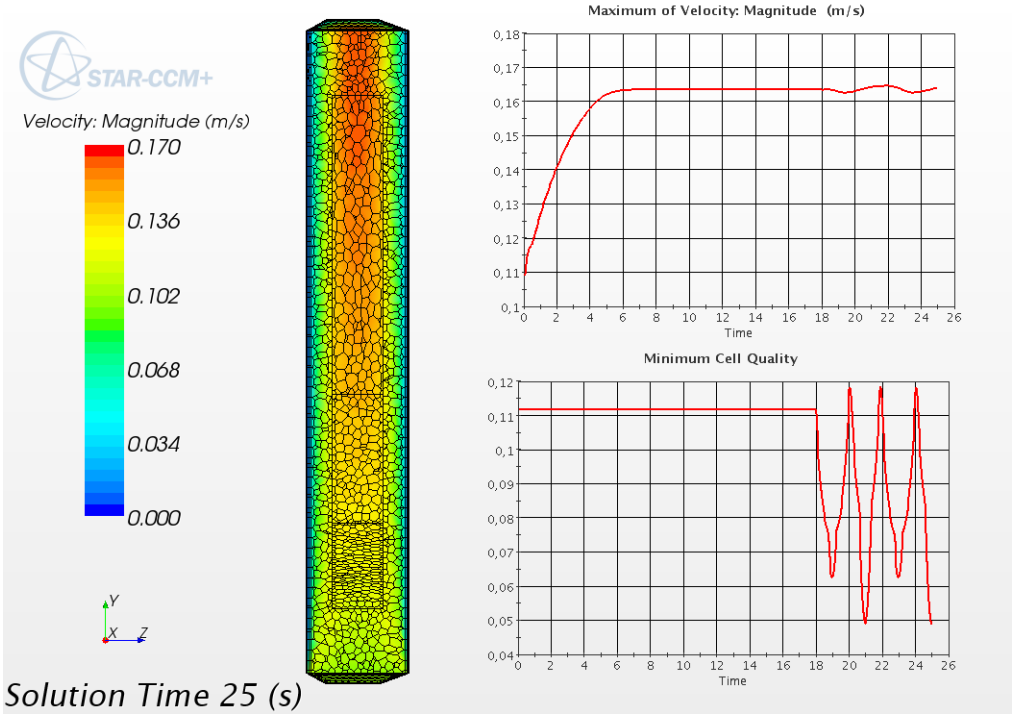
Fig. 26: Flow velocity development coupled with morphing/translation motions in hexagonal domain

- the various constraints (like points fixation, equal length line segments, collinear, parallel or perpendicular lines);
- the construction lines (not only right lines but also circular arcs), extremely useful for the construction of a correct geometry;
- the circular pattern created with respect to angle symmetries.

The geometry of the central body that should move inside the guide tube is the one illustrated in Fig. 28; the volume mesh of the internal structure is exposed in Fig. 29.

The settings for the simulation of the displacement were the same as before (Translation for the central body and Morphing for the up and down bodies that must follow the rigid motion through appropriate morpher boundary conditions).

In order to reduce computational costs, we chose to perform the simulation only for a sixth part of the entire physical domain, this choice meaning that instead of running with about 400 thousands of cells we run the simulation with about 70 thousands of volume cells.



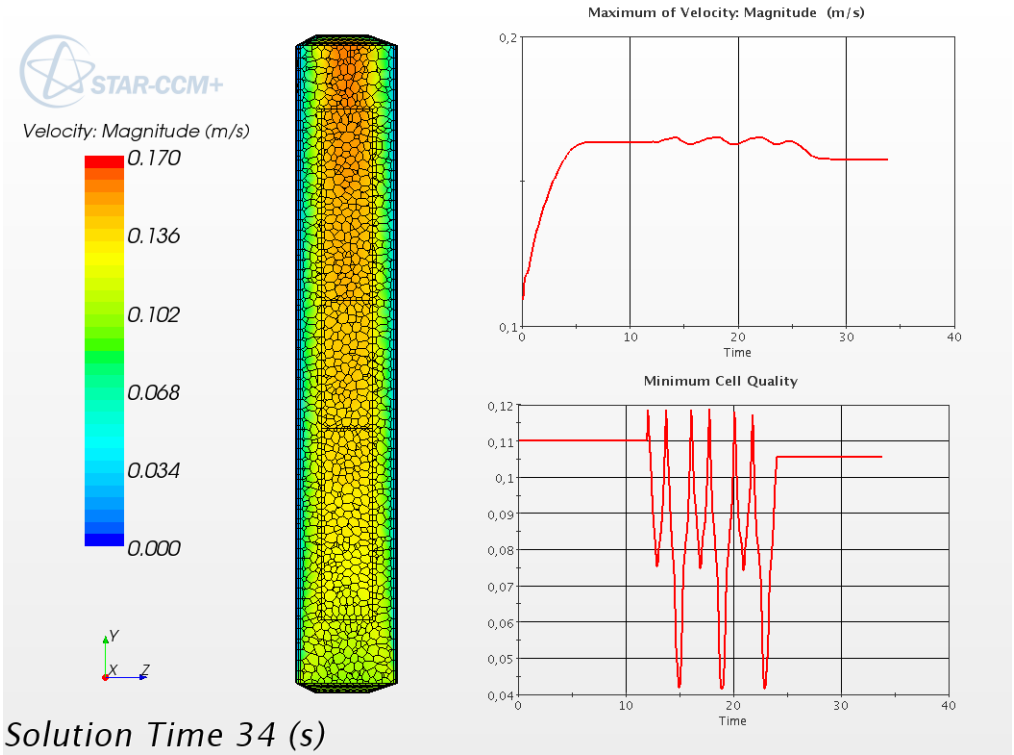


Fig. 27: Mesh deformation in the case of a hexagonal piston in hexagonal domain

6.2.1 Improvement of the surface mesh extraction strategy

During the development of the motions, the re-meshing became necessary and we could either extract a new boundary surface mesh from the deformed volume mesh and use it

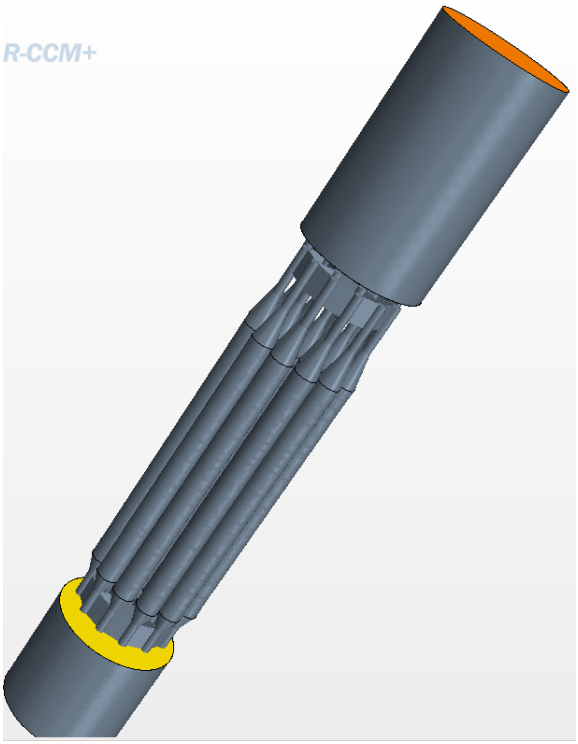


Fig. 28: Geometry of the control rods system inside a guide tube

as starting point for the re-generation of a new mesh, or we could return to the geometry through a CAD up-dating and re-generate the volume mesh only in the regions deformed with the morphing.

The reason for which we initially chose to extract a new surface mesh for the re-creation of the volume mesh, was principally due the fact that the feature curves for the given geometry were sufficient for the preservation of its original form and, in addition, they were perfectly translated by the imposed motions.

We outline here that the feature curves are an important part of the surface definition in order to get a high quality mesh (both surface and volume). As part of the meshing process, they are used to define edges and surface features from the surface geometry all the way through to the final volume mesh. If they are not included in the starting surface then such features will not be preserved. That is the reason why we included new feature curves on the extracted surface and, with the upgraded versions of the code, they coincide with the initial ones.

With the Release 6.02, the Surface Remesher model was improved with a new method for curvature refinement allowing refinement by curvature deviation distance. The method allows better control of the curvature refinement when there are large variations in the curvature size. In the Surface Remesher model, all the CAD edges are retained, getting an improvement of the surface mesh quality.

Indeed, by applying the surface mesh extraction strategy described in the Section 5.1 just a few times, the propagation of the resulting error was less significant.

Remark 8 *However, while facing with an initially failing strategy, the worth of our research*

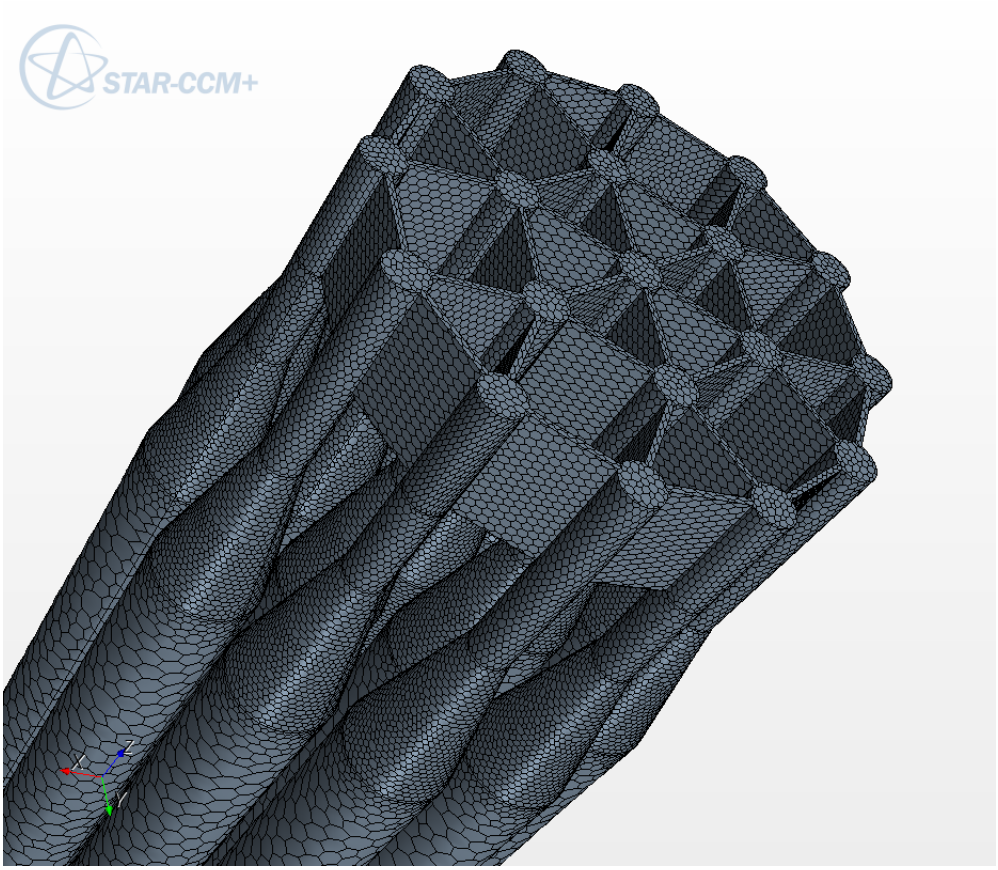


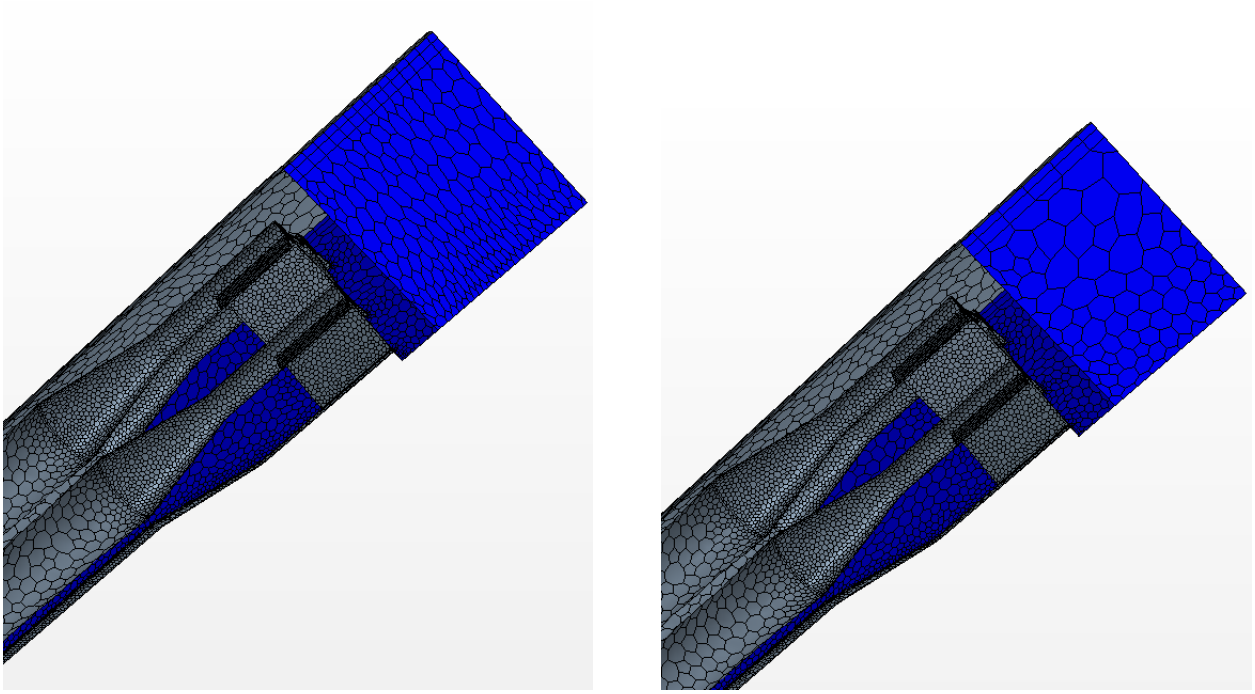
Fig. 29: Volume mesh of the control rods system body

is reflected into the development of alternative strategies and possibly less expensive, from the computational costs point of view.

In the images in Fig. 30 we illustrate the deformed state of the mesh on a symmetry wall, followed by its re-creation; for a better visualization and also for a demonstration of the good results of applying this approach, an animation may also be provided on demand.

The simulation was run firstly in the stationary case and a report was set to calculate the pressure force that the fluid exerted on the surface of the rods system. Then the translational motion got started, followed by a re-generation of the volume mesh obtained by surface extraction. In a plot based on the report mentioned above we can see the development of the force values. At the end of the stationary running, the Residuals display showed most of the residuals flattening out, which is a good indicator of convergence. To confirm that the solution has converged, also the force flattened out.

When we set the transient case, hence when the translation and morphing motions stared, the plot showed significant changes in the force values in the sense of passing from F = 0.02 N to smaller values approaching to zero. The remarkable aspect instead was that the force values did not register oscillations due to the re-meshing operation, demonstrating the compatibility of the techniques with the physics involved. When the fluid was let again to settle down, the force plot flattened again showing the convergence. As an illustration of the



(a) (b)

Fig. 30: Volume mesh: deformed by displacement (a) and recreated with re-meshing (b)

above observations, we captured some images reported into the Figs. 31 and 32[8].

6.2.2 Update CAD with two design parameters

Even if it was possible to apply the previously described technique, we decided to employ also the strategy of up-dating the geometry, combining both the procedures illustrated into the Sections 5.2 and 5.3.

More explicitly, for the realization of re-meshing we employed both the Algs. (2) and (3) with the following key elements:

- the relation (13), controlling the translation operation, to the internal structure representing the rods system;
- the relations (19) and (20), controlling the variation of the extrusion distances, to the external guide tube.

Remark 9 *In the Java macro that controls the mesh re-generation, the CAD Update was implemented in the following manner:*

```
double d = maxReport_0.getReportMonitorValue();
ExtrusionMerge extrusionMerge_0 = ((ExtrusionMerge)
cadModel_0.getFeatureManager().getObject("Extrude 4"));
MoveBodyFeature moveBodyFeature_0 =
```

[8] An illustrative animation may be provided on demand.

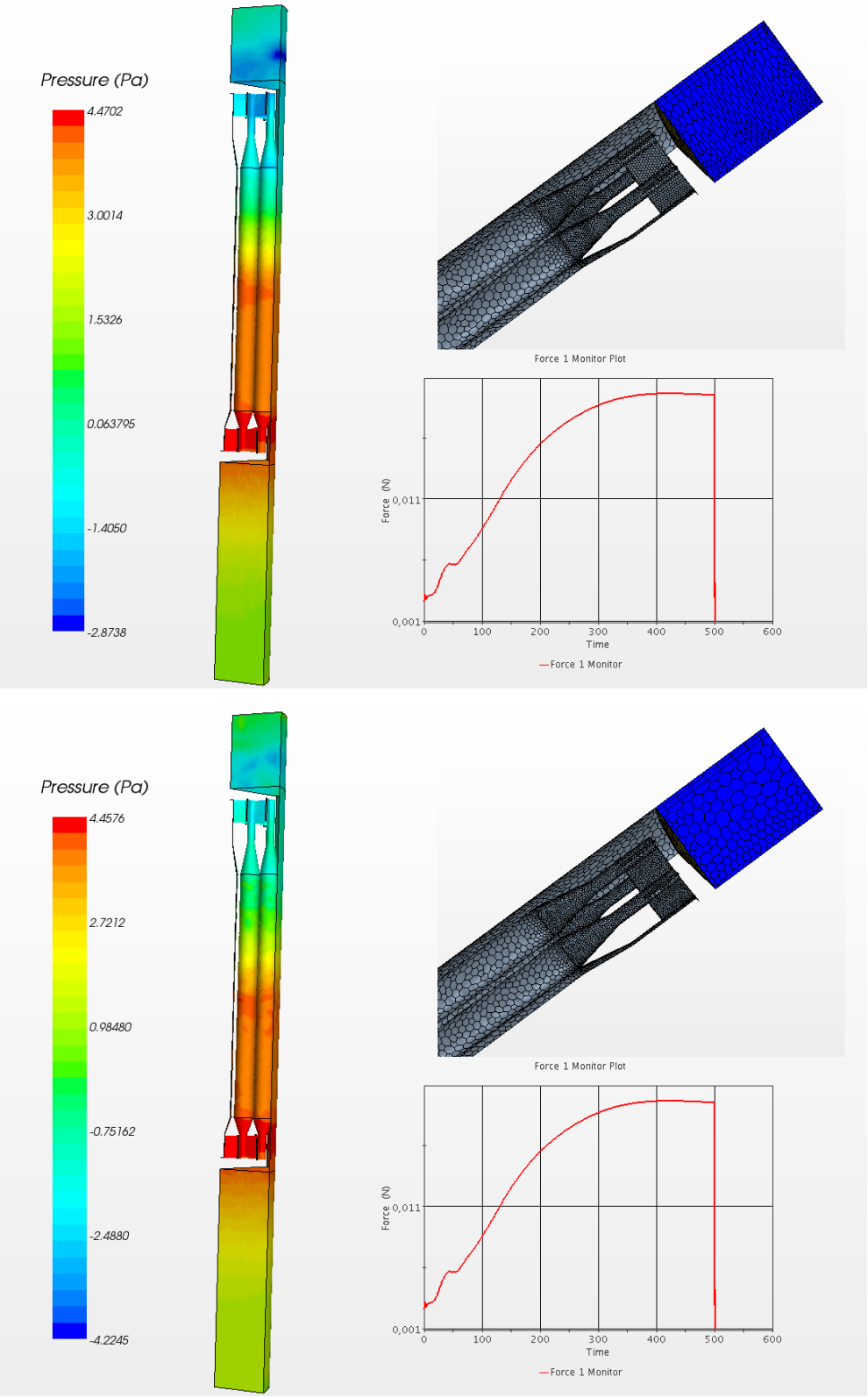


Fig. 31: Force development during re-meshing

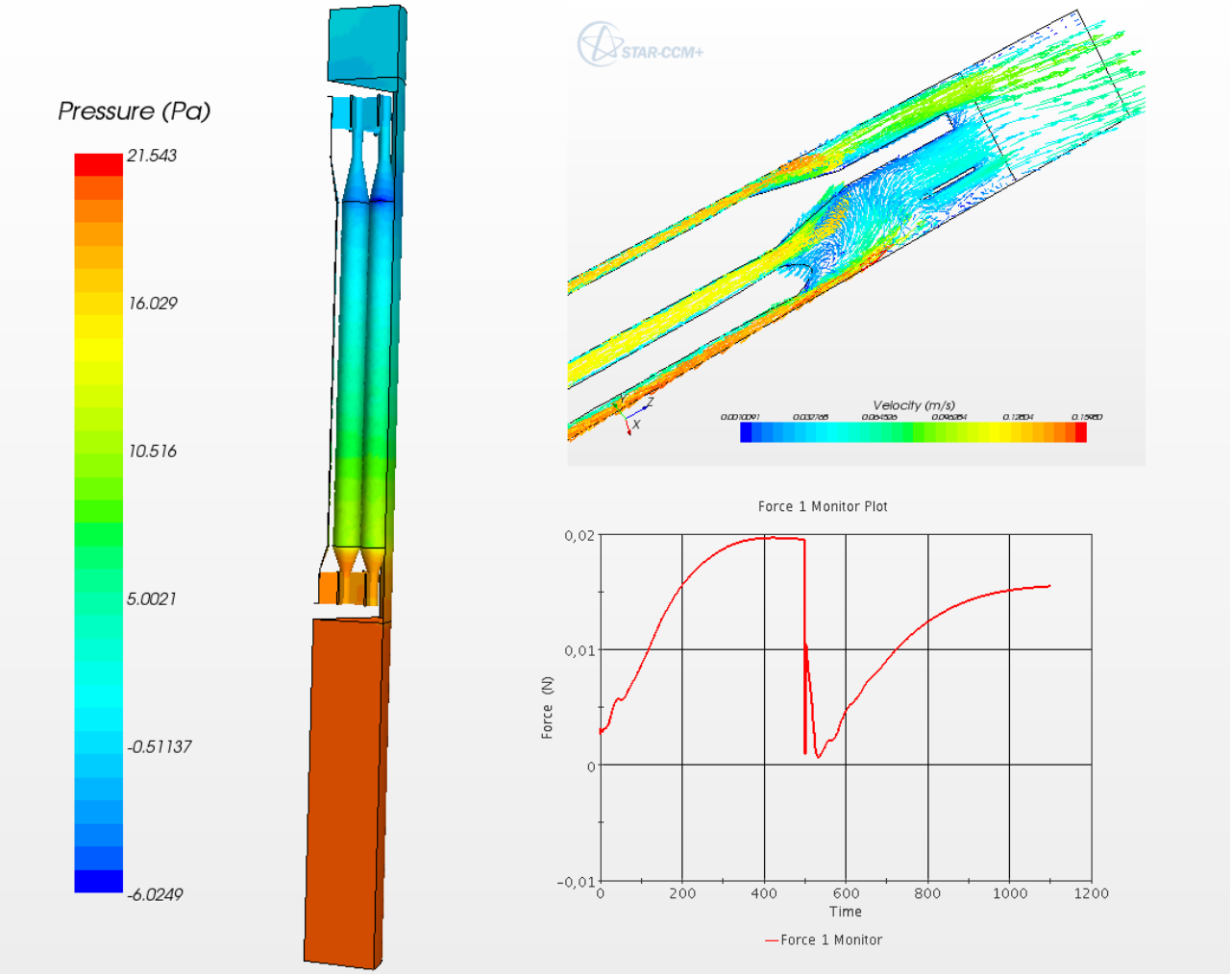


Fig. 32: Final force development

```
((MoveBodyFeature) cadModel_0.getFeatureManager().getObject("MoveBody 5"));

CadModelCoordinate cadModelCoordinate_0 =
moveBodyFeature_0.getTranslationVector();
extrusionMerge_0.getDistance().setValue(d);
extrusionMerge_0.getDistanceAsymmetric().setValue(0.34-d);
cadModelCoordinate_0.setCoordinate(units_0, units_0, units_0,
new DoubleVector(new double[] {0.0, 0.0, d-0.33}));
```

The main advantage of this approach was that the central body didn't have to be re-meshed. This is very important, since in the simulation we deal with a high number of volume cells. In Fig. 33, informations about pressure, force and translation velocity fields in the case of CAD Update re-meshing are reported[9].

[9]Animations with application of this strategy may be provided on demand.

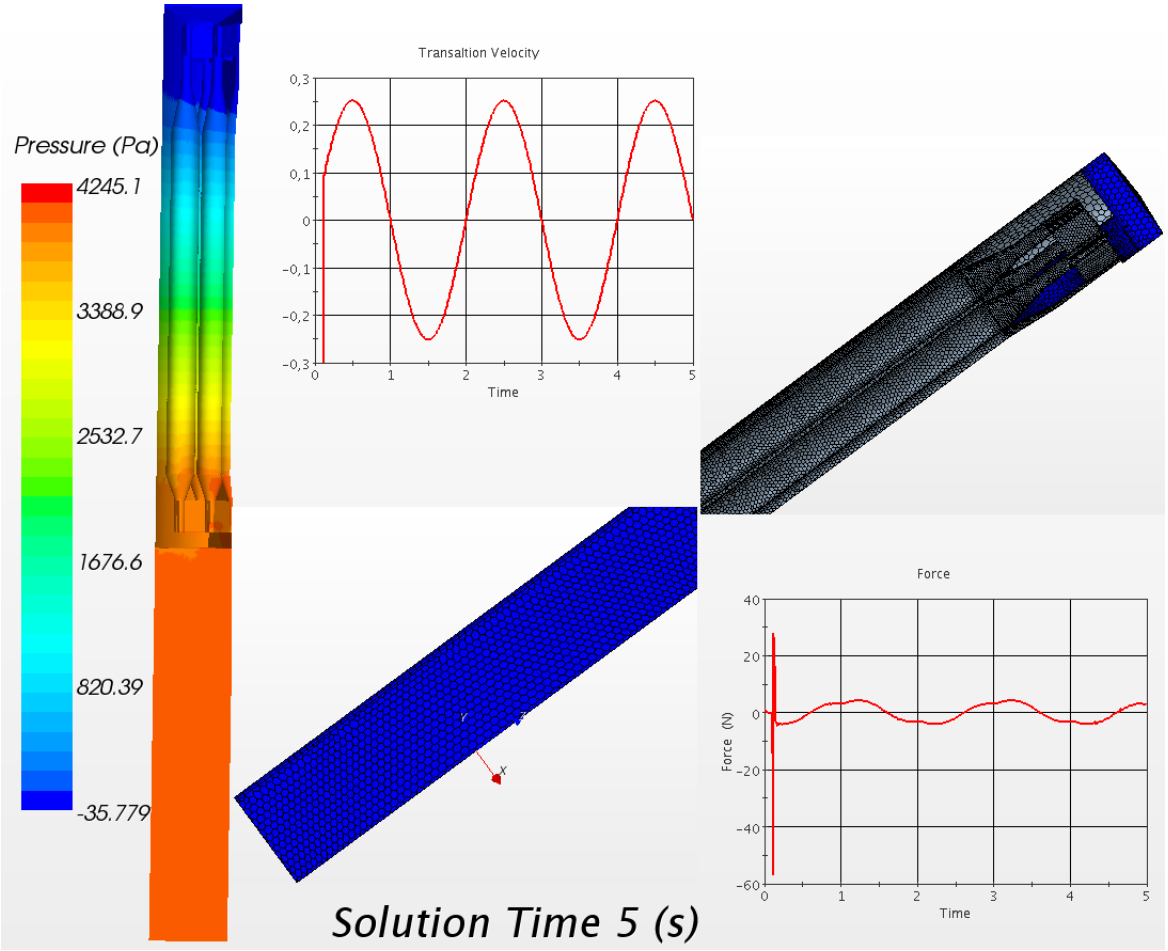


Fig. 33: Force and translation velocity development coupled with Update CAD re-meshing

7 Conclusions and further work

We presented in this report our approach to the mesh morphing techniques implemented into the Star-ccm+ CFD simulation code and their employment principally in applications of fluid-structure interactions and rigid body motions.

After getting the necessary preliminary familiarity with the implementation methods of the code, we concentrated the attention on those applications that required the preservation of a good quality of the mesh while simulating solid walls oscillations or contraction/expansion of fluid domains coupled with rigid body translations. Explicitly, we then considered and analysed the study case of a piston performing a translation motion inside a larger fluid pipe with all its implications and effects upon the surrounding mesh.

We demonstrated how it was possible to take full advantage of the automation capabilities of the implementation code to completely (or partially) re-mesh the model when a high deformation of the cells became an obstacle for the simulation to run. Indeed, we have been able to automate the necessary procedures by means of the Java scripting, using Netbeans as IDE.

Three strategies of re-meshing have been developed and illustrated, where the dumping

of the first one permitted to well and quickly develop the next one and optimize a third one. In facing the failing procedure matter, we have been able to develop alternative strategies that proved more appropriate for our purposes.

Finally, we successfully used these strategies for the simulation of smooth translation motions and morphing of cylindrical/hexagonal/complex geometry bodies inside a fluid pipe or a guide tube.

In conclusion, we obtained/developed the:

- capacity of extending the code capabilities, by using the Java macros;
- optimisation of the computation/mesh regeneration time, through a selective (region-wise) re-meshing;
- regular development of the fluid flow, even in presence of re-meshing operations;
- ability to perform simulations of relative bodies motions, combining sliding interfaces with moving/deforming domains, maintaining or quickly recreating good quality mesh.

As further developments, we should simulate the mechanical movement of the control rod system within its guide tube not only with imposed displacement but also in relation to the drag forces exerted by the fluid, the buoyancy, the gravity, ecc.

We should further work on the integration of the moving parts already described in a larger closed loop and the coupling with the physical requirements of a rather complex experimental nuclear facility.

Acknowledgements.

This project is supported by RAS (Regione Autonoma della Sardegna) through a grant co-funded with funds PO Sardegna FSE 2007-2013, L.R.7/2007 *"Promozione della ricerca scientifica e dell'innovazione in Sardegna "- "Promotion of the scientific research and technological innovation in Sardinia"* and is guested by CRS4, in the Energy and Environment Program.

References

[1] ALEXA M. *Recent Advances in Mesh Morphing*, Computer Graphics Forum (9/2002).

[2] BOER A., SCHOOT M.S., BIJL H., *Mesh morphing based on radial function interpolation*, Elsevier, 2007.

[3] FERZIGER, J.H., PERIC, M., *Computational Methods for Fluid Dynamics*, 3rd rev. ed., Springer-Verlag, Berlin, 2002.

[4] HARDY, R. L., *Theory and applications of the multiquadric-biharmonic method*, Comput. Math. Applic. (1990), Vol. 19, pp. 163-208

[5] NEALEN, MULLER, BOXERMAN, CARLSON *Physically Based Deformable Models in Computer Graphics*, Eurographics 2005

[6] CD-adapco website, http://www.cd-adapco.com/ (Last access March 2012)

[7] PROFIR M. M., *Mesh morphing techniques in CFD*, Proceedings of ISCOPAM, Iasi, July 12-16, 2010.

[8] PROFIR M. M., *Mesh morphing implemetation methods with examples*, Proceedings of FG60, Bertinoro, Italy, June 16-19, 2010.

[9] USER GUIDE STAR-CCM+ Version 6.06.011.