Industrial Application of the Meshless Morpher RBF Morph to a Motorbike Windshield Optimisation

M.E. Biancolini *, C. Biancolini, E. Costa, D. Gattamelata, P.P. Valentini
University of Rome Tor Vergata, Mechanical Engineering Department, Via Politecnico 1, 00133 Roma, Italy.

*corresponding author biancolini@ing.uniroma2.it

ABSTRACT

In this paper the aerodynamic optimisation of a motorbike windshield is presented. This challenging optimisation task was made possible tanks to Fluent and the embedded morpher tool RBF Morph, capable to modify the baseline mesh accounting for set-up and for shape changes.

The approach is based on a suite of UDF functions that allow to prescribe surface modifications, to smooth the volume mesh accordingly and to update the fluid solution. The morpher is based on a meshless approach (i.e. is defined using only a set of points and produces a deformation field), can be used both in serial and parallel sessions and allows to manage any possible kind of mesh elements. Morphing set-up is done inside Fluent through a comprehensive and user-friendly GUI which allows to define the problem interacting with Fluent entities; moreover TUI commands are also available to modify the shape by means of simple scripts.

1. INTRODUCTION

Shape optimization is a very important topic especially in the problems where the motion of a fluid has an important impact on performances. In fact, a slight shape modification can dramatically affect the behaviour of a component that interacts with the fluid. CFD can give an important aid to drive the design of such critical components but a true parametric CFD solver, suitable for optimization, is still missing on the market, especially when large problems need to be handled.

Despite shape parameterization is available in the CAD model, used as starting point for CFD model generation, the complex chain that allows to obtain a reliable CFD grid is very difficult to manage. As such, parametric properties and geometric features of the original CAD model are usually lost in the final mesh.

The effects of slight modifications can be addressed acting at final level of the complex aforementioned chain: the CFD mesh. In fact required modifications can be introduced by morphing the surface mesh at the boundary of fluid mesh and propagating such deformations inside the domain by means of a smoother. Original mesh topology is preserved but the final quality of the mesh depends on the action of surface morpher and fluid smoother.

In this paper the new morphing product RBF Morph is presented starting from the exposition of the background theory of Radial Basis Functions used for the implementation of the numerical kernel of the software.

To better understand how RBF Morph can be used for industrial cases, a practical application is considered in the present study: the optimization of a motorbike windshield.

2. RBF MORPH

The new product RBF Morph, an integrated system for morphing and shape optimization tailored for the CFD solver ANSYS Fluent, is herein presented. RBF Morph is fully integrated in the CFD solving process and combines a very accurate control of the geometrical parameters with an extremely fast mesh deformation. RBF Morph is the result of the joint between academic state-of-the-art research and top-level industrial needs. In the present implementation, the morpher has been tailored to ANSYS Fluent. However, the kernel of the software represents the most sophisticated component, and could be adapted to different tasks or stand-alone work.

2.1 The aim

The aim of the RBF Morph is to perform fast mesh morphing using a mesh-independent approach based on state-of-the-art RBF (Radial Basis Functions) techniques.

The use of RBF Morph allows the CFD user to perform shape modifications, compatible with the mesh topology, directly in the solving stage, just adding one single command line in the input file.

The most important requirements are:

- mesh-independent solution;
- parallel morphing of the grid;
- large size models (many millions of cells) must be morphed in a reasonable short time
- management of every kind of mesh element type (tetrahedral, hexahedral, polyhedral, prismatic, hexcore, non-conformal interfaces, etc.).

The final goal is to perform parametric studies of component shapes and positions typical of the fluid-dynamic design like:

- design Developments;
- multi-configuration studies;
- sensitivity Studies;
- DOE (Design Of Experiment);
- optimization.

2.2 Background

A system of radial functions is used to produce a solution for mesh movement/morphing, from a list of source points and their displacements [1,2]. This approach is valid for both surface shape changes and volume mesh smoothing.

Radial basis were born as an interpolation tool for scattered data and consist of a very powerful tool because they are able to interpolate everywhere in the space a function defined at discrete points giving the exact value at original points. The behaviour of the function between points depends on the kind of basis adopted.

The radial function can be fully or compactly supported, in any case a polynomial corrector is added to guarantee compatibility for rigid modes.

Typical radial functions are reported in the following table.

Radial Basis Function	$\phi(r)$
Spline type (R_n)	$\lvert r\rvert^n$, n odd
Thin plate spline (TPS_n)	$\lvert r\rvert^n \log\lvert r\rvert$, n even
Multiquadric(MQ)	$\sqrt{1+r^2}$
Inverse multiquadric (IMQ)	$\frac{1}{\sqrt{1+r^2}}$
Inverse quadratic (IQ)	$\frac{1}{1+r^2}$
Gaussian (GS)	e^{-r^2}

As will be shown in detail, a linear system (of order equal to the number of source point introduced) need to be solved for coefficients calculation. Once the unknown coefficients are calculated, the motion of an arbitrary point inside or outside the domain (interpolation/extrapolation) is expressed as the summation of the radial contribution of each source point (if the point falls inside the influence domain).

Details of the theory need to be given using some equations. An interpolation function composed by a radial basis and a polynomial is defined as follows:

$$s(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + h(\mathbf{x})$$

The degree of the polynomial has to be chosen depending on the kind of radial function adopted. A radial basis fit exists if the coefficients γ and the weight of the polynomial can be found such that the desired function values are obtained at source points and the polynomial terms gives 0 contributions at source points, that is:

$$s(\mathbf{x}_{k_i}) = g(\mathbf{x}_{k_i}) \quad 1 \le i \le N$$

$$0 = \sum_{i=1}^{N} \gamma_i q(\mathbf{x}_{k_i})$$

The minimal degree of polynomial p depends on the choice of the basis function. A unique interpolant exists if the basis function is a conditionally positive definite function. If the basis functions are conditionally positive definite of order m <= 2, a linear polynomial can be used. The subsequent exposition will assume valid the aforementioned hypothesis. A consequence of using a linear polynomial is that rigid body translations are exactly recovered. The values for the coefficients γ and the coefficients β of the linear polynomial can be obtained by solving the system:

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix}$$

where g are the know values at the source points. M is the interpolation matrix defined calculating all the radial interactions between source points:

$$M_{ij} = \phi\left(\left\|\mathbf{x}_{k_i} - \mathbf{x}_{k_j}\right\|\right) \quad 1 \le i \quad j \le N$$

and P is a constraint matrix that arises balancing the polynomial contribution and contains a column of “1” and the x y z positions of source points in the others three columns:

$$\mathbf{P} = \begin{pmatrix} 1 & x_{k_1}^0 & y_{k_1}^0 & z_{k_1}^0 \\ 1 & x_{k_2}^0 & y_{k_2}^0 & z_{k_2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N}^0 & y_{k_N}^0 & z_{k_N}^0 \end{pmatrix}$$

Radial basis interpolation works for scalar fields. For the smoothing problem each component of the displacement field prescribed at the source points is interpolated as follows:

$$\begin{cases} v_x = s_x(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^x \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ v_y = s_y(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^y \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ v_z = s_z(\mathbf{x}) = \sum_{i=1}^{N} \gamma_i^z \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k_i}\right\|\right) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{cases}$$

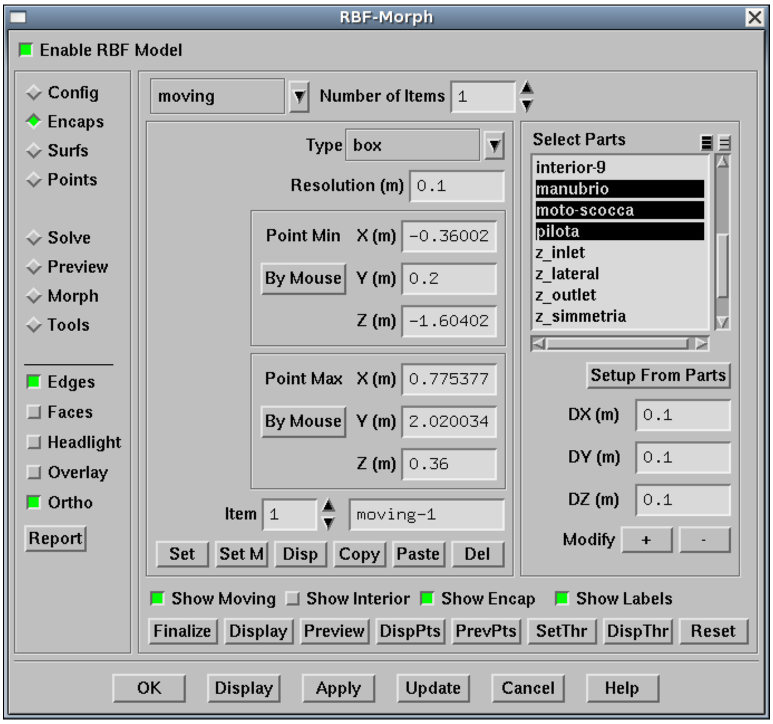
Radial basis method has several advantages that make it very attractive in the area of mesh smoothing. The key point is that being a meshless method only grid points are moved regardless of element connected and is suitable for parallel implementation. In fact, once the solution is known and shared in the memory of each calculation node of the cluster, each partition has the ability to smooth its nodes without taking care of what happens outside because the smoother is a global point function and the continuity at interfaces is implicitly guaranteed.

2.3 How does it work

Radial Basis Function interpolation is used to derive the displacement in any location in the space, so it is also available in every grid node.
RBF Morph requires three different steps:

- Step1: [SERIAL] setup and definition of the problem;
- Step2: [SERIAL] solution of the RBF system;
- Step3: [SERIAL/PARALLEL] morphing of surface and volume mesh.

The serial setup requires an intense use of RBF Morph GUI. The GUI offers several tools for the definition of the problem. It is composed by a switchable principal panel (Figure 1). Acting on the radio buttons on the left 8 different operative modes are accessed. The first 4 panels (Config, Encaps, Surfs, Points) are addressed to problem set-up, the other 3 (Solve, Preview, Morph) allows to calculate the rbf solution, to preview its effect and to apply it for morphing and the last panel contains some utilities useful for the RBF Morph software.



RBF-Morph

☑ Enable RBF Model

○ Config
● Encaps
○ Surfs
○ Points
○ Solve
○ Preview
○ Morph
○ Tools

☑ Edges
□ Faces
□ Headlight
□ Overlay
☑ Ortho

moving ▾ Number of Items 1

Type box ▾
Resolution (m) 0.1

Point Min X (m) -0.36002
Y (m) 0.2
Z (m) -1.60402

Point Max X (m) 0.775377
Y (m) 2.020034
Z (m) 0.36

Item 1 moving-1

Select Parts: interior-9, manubrio, moto-scocca, pilota, z_inlet, z_lateral, z_outlet, z_simmetria

DX (m) 0.1
DY (m) 0.1
DZ (m) 0.1

☑ Show Moving □ Show Interior ☑ Show Encap ☑ Show Labels

Figure 1: GUI of RBF Morph. The “Encaps” panel is shown

After completing the step1 it is possible to pass to the step2 and calculate the rbf solution. The effect of an imposed modifier can be verified previewing its action (an arbitrary number of surfaces can be morphed on the fly showing the results in the Fluent graphic viewport) without moving the nodes, or exploiting the undo capability that allows to examine the morphed mesh checking its quality and the possible appearing of negative cell volume areas. Once that the modifier is acceptable it can be saved on file. The operation can be repeated for each desired modifier.

The third step can be performed in serial or in parallel with or without the GUI. Once that the solutions are available they can be loaded and used to morph the mesh using the morph panel of the GUI or they can directly used by means of TUI commands that allow to prescribe a single morph or a multi-morph summing the effect of multiple modifiers. Considering that

each modifier can be applied with the desired magnitude (i.e. a scalar to set the intensity of the modifier) a parametric Fluent model results.
Since the modifiers are non-linear and large mesh motion are involved the effect of multiple modifier action depends on the application command sequence. For this reason, the multi-morph command superimpose the effects using the same baseline mesh as the starting point of each modifier. Different sequences can be imposed by the user applying the single morph after the action of a previous morph. But in this case a wise procedure is to direct control the effect of the sequence of morphing. For special cases a custom sequence of morphing actions can be programmed as an additional UDF.

3. OPTIMIZATION OF A WINDSHIELD

Variotouring windshield has been introduced in 2002 by the German company MRA. The idea is to guide the shape of the flux that acts on the driver. At a cruise speed of 130kph with a naked motorbike the air fluxes push the torso of the driver and push under the helmet; when a traditional touring windshield is installed the flux loads produce a fastidious tension on the driver neck. The aim of the variotouring (Figure 2) is to control the fluxes path to limit the annoyance thanks to a special shape of the screen and an adjustable deflector. The system acts as a flux splitter and, if properly tuned, allows obtaining a substantial benefit in term of riding comfort.

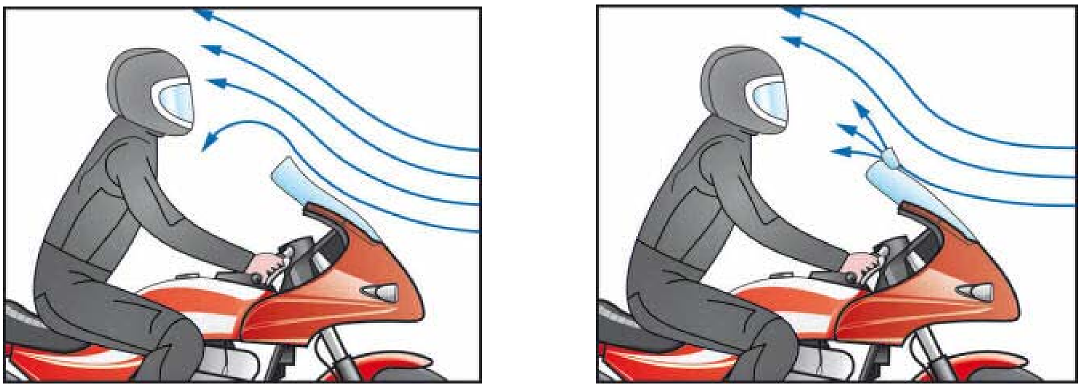


Figure 2: Working principle of the variotouring shield. The air flux is split by the channel between the deflector and the screen to obtain an optimal flow fields around the helmet.

Figure 3 shows the geometrical models of the study which were obtained according to the reverse engineering procedure. These two models consist of a bike (Ducati Multistrada) equipped, respectively, with its original windshield (on the left) and with the windshield MRA variotouring (on the right).

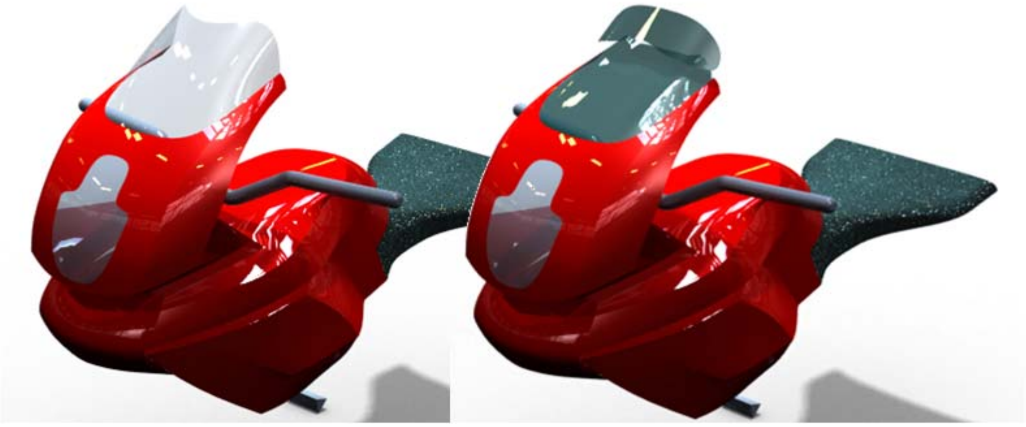


Figure 3: The virtual model with the original windshield (on the left) and that with the variotouring windshield (on the right).

A complete design loop has been defined with the aim to establish a fast method to introduce a new product at the first presentation of the motorbike model. The design loop consists in the acquisition of the actual geometry of part of the motorbike by means of a reverse engineering tool, the definition of the new windshield geometry, the introduction of the driver into CAD model, the definition of the calculation mesh, the CFD model of the baseline geometry (one with the original windshield and another one with the new component), the shape optimisation of the new geometry.

The optimisation goal is the trade-off between driver comfort, which can be addressed evaluating the resultant of aerodynamic actions on the helmet, and the vehicle drag. Given that the windshield is adjustable, an optimal angle can be found for the driver height. For this reason design optimisation becomes a multi-objective problem because design parameters (i.e. shape of the two components of variotouring) affect how the component will perform for different drivers.

3.1 Windshield and faring digital mock up

The Reverse Engineering (RE) process aims at reproducing physical and topological properties of a manufactured product. The duplication involves four main phases:

- Scanning/Measuring the component or the assembly;
- Pre-processing of the acquired data;
- Rebuilding and modelling;
- Rendering.

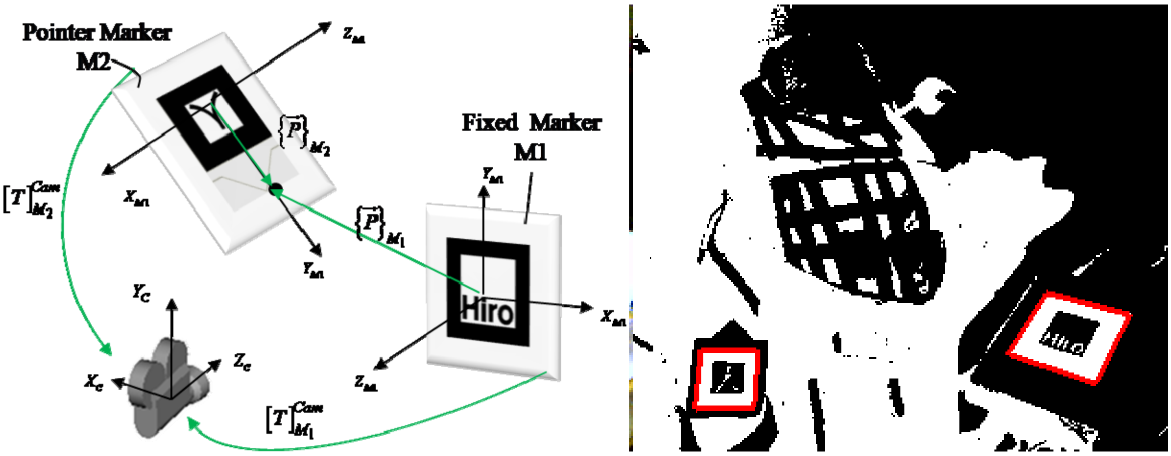


Figure 4: The system working scheme (left), the elaborated video frame with markers (right).

The result of the entire process is affected by the acquiring precision and typology of the first step instrument. There are two main groups of capture points devices: contact and non contact ones. The first group is generally more precise and allows the user to choose the points of interest. So, for this group, the pre-processing phase is simplified. The non contact point capture devices are faster than contact ones, but they require a time expensive pre-processing phase.

In this paper a hybrid method has been adopted in order to build the digital mock-up of Ducati Multistrada fairing and windshield. This system has been developed from the authors and is based on Augmented Reality technology. The purpose was to have a cheap and fast method with suitable accuracy.

The working principle is illustrated in Figure 4 (on the left): there are two markers and a monocular acquiring system. One marker is attached to the scene and acts as fixed reference frame, while the other is free and works as pointer in the user hand. The camera detects live video frames and save them in the computer memory. The computer, by means of image processing library [3], processes the camera frames (see Figure 4 on the right) and recognizes the markers in the scene by means of a pattern recognition methods. After that, through a modified DLT (*Direct Linear Transformation* [4]) method, the software computes the relative positions between camera and markers and stores the transformations in 4x4 omography matrices $[T]_{M_i}^{Cam}$. So, starting from the knowledge $[T]_{M_2}^{Cam}$ and $[T]_{M_1}^{Cam}$, the system computes the position of the pointer tip *P* in the reference frame of the fixed marker accordingly the following expression:

$$\{P\}_{M_1} = [T]_{Cam}^{M1} \cdot [T]_{M_2}^{Cam} \cdot \{P\}_{M_2} = [T]_{M_2}^{M_1} \cdot \{P\}_{M_2}$$

where $[T]_{Cam}^{M1}$ is the inverse matrix of $[T]_{M_1}^{Cam}$. So the user is able to pick points in the real environment by means of the pointer marker, while the application manager visualizes the acquired points on the display (see Figure 5).

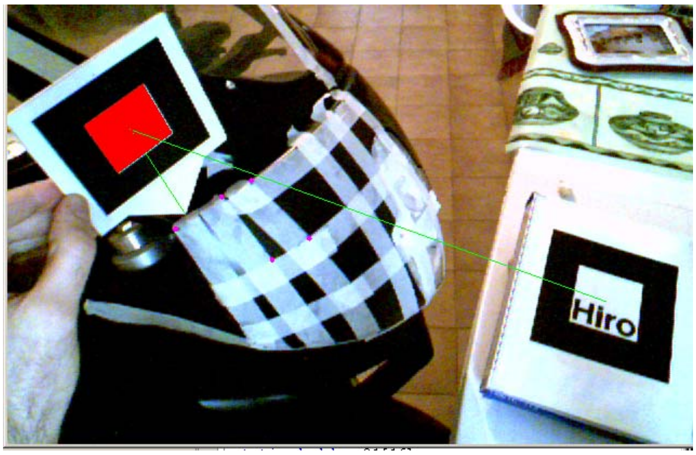


Figure 5. The acquired points visualized thanks to the Augmented Reality technology.

Since the system is based on image processing, the marker visibility in the frame and the illumination settings strongly influence the system precision and reliability. Indeed the system required a preceding phase of set-up for the object and marker disposition and for the light arrangement in the scene.

So an iterative procedure of light and object setting, followed by precision checking, has been applied to the system. After some optimization cycles the system was able to acquire points with an absolute error less than 2 mm; the precision of the acquisition has been further incremented imposing the exact matching of several global measures.

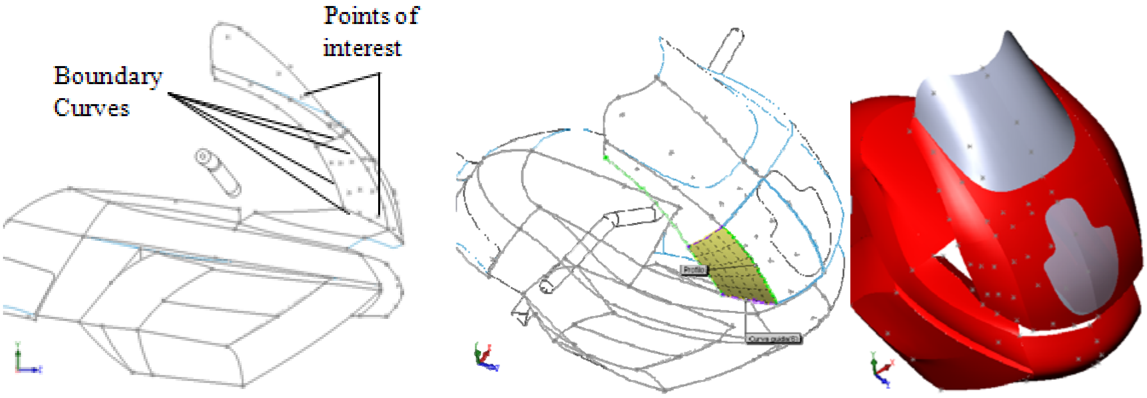


Figure 6. The modelling phases: building boundary curves (on the left), modelling surfaces (on the middle) and final result (on the right).

After the setting up phase, a new preparing phase has been adopted to choose the points of interest and to simplify the object modelling.

For this job an adhesive tape and a pen have been used. As Figure 4 and 5 show, the strips of tape has been applied to the boundary of the fairing patches and at the middle of each patches in order to catch information about surface curvature and bending.

After the acquiring procedure has been repeated some times, to reduce the measure uncertainty, the attention was focused on modelling procedure. Starting from the chosen points, the geometries have been duplicated with cad modelling commands. For the boundary surfaces *spline* features were used, while to draw the surface patches of the acquired geometries the authors used the *lofting* and *sweeping* features (see Figure 6).

In Figure 3 the original rendered model and the model with variotouring windshield are compared.

3.2 CFD models

CFD mesh has been generated according to the best-practice procedure used for external aerodynamic analysis. Solid models from CAD have been processed to obtain a valid surface mesh. The virtual wind tunnel has then been added to the motorbike model, only half model has been considered exploiting symmetry. Considering that the study is focused on the performance of the windshield only the relevant components of the vehicle have been represented in the model (Figure 7 left). Surface mesh has been used to generate internal fluid mesh adopting two techniques: for the first explorative analysis a simple tetrahedrons mesh has been considered whereas a hybrid mesh structure has been subsequently used. In the latter case a prisms layer has been first generated to properly capture boundary layer physic. Hexcore mesh (saved as polyhedral mesh) has been used inside the volume. The transition between hexcore and prisms has been built by means of a tetrahedrons mesh (Figure 7 right). The model size is about a few millions of cells.

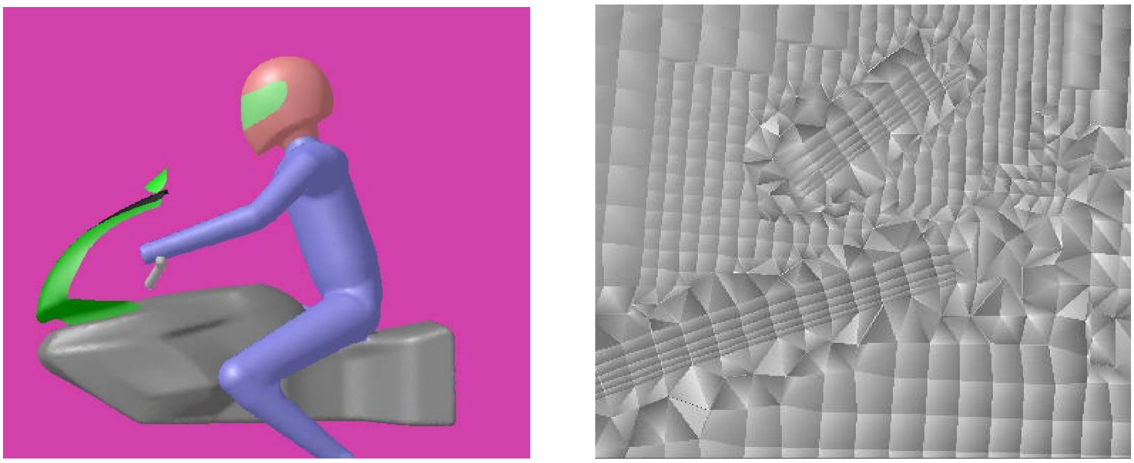


Figure 7: CFD mesh details: modelled components (left), fluid mesh cut showing the internal transition between boundary layer and hexcore.

Boundary conditions are imposed prescribing inlet velocity, ground velocity and static pressure at outlet. The solution type is steady. The reference solution has been obtained starting from inlet speed as the initial condition and using as convergence criteria the stabilization of aerodynamic forces acting on the vehicle. A good convergence has been observed after 1000 iterations. For morphed solution a fixed number of 300 iterations has been used considering that, according to some numerical tests, the starting solution resulting from the reference model provides a very good initial condition and convergence can be achieved very fast.

3.3 Morphing

The RBF Morph add-on has been used to deform the original CFD model considering three deforming actions:

- changing of driver height;
- changing of driver position acting on the hunching angle;
- adjustment of the variotouring acting on the deflector angle.

The set-up stage for changing driver angle (or height) starts with the definition of an encapsulation box (Figure 8 left). Encapsulation domains of various shape (box, sphere and cylinder) can be used to limit the action of the morpher. For complex shapes the encapsulation domain can be defined combining an arbitrary number of such shapes (only the effective envelope will be used to locate source points). The number of points located on the surface is defined imposing proper point spacing. The effect of encapsulation is to give a near zero solution on the boundary (in fact zero value is imposed only on the source points, the zero values in other points on the encapsulation surface depends on spacing); furthermore the geometrical information are used to apply the morph only to the mesh nodes that fall inside the encapsulation domain. Moving encapsulation are also available (not used in this example): they work with a similar manner of encapsulation domains but prescribe a simple deformation field (i.e. rigid motion or scaling) inside the encapsulation and move the source points on the boundary accordingly. This means that the morpher action is applied only to the nodes contained inside the encapsulation domain and outside of the moving domains.

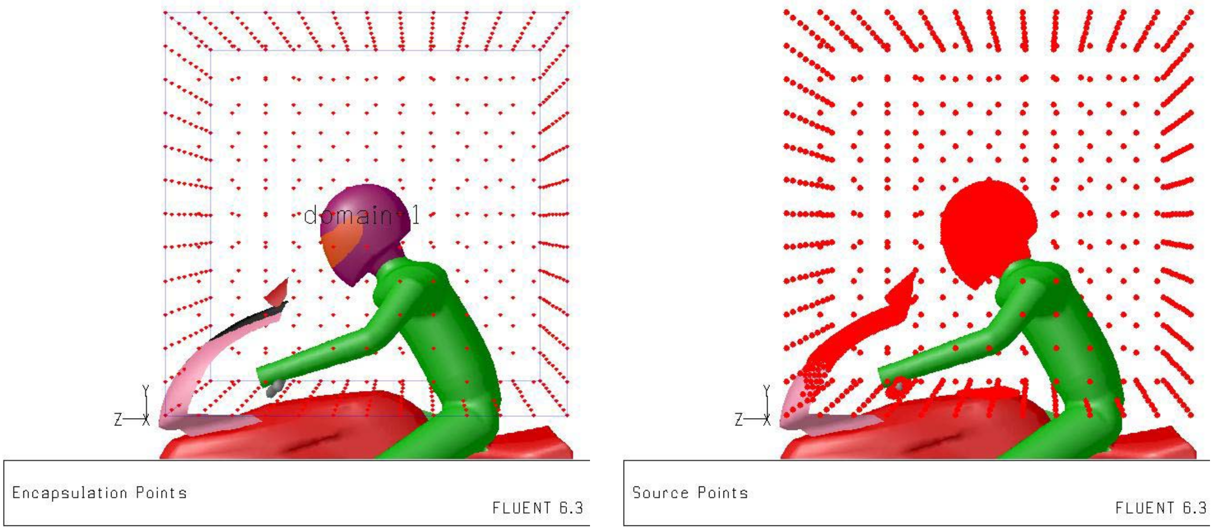


Figure 8: Set up step of RBF Morph. The morphed action is limited in the box region “domain 1” (left). The motion of the surfaces inside the encapsulation domain (right) is imposed to the points on the windshield (fixed), the fairing (fixed) and the helmet (moving).

To complete the set-up, two sets of source points on the surface are defined. The first one is composed by all the mesh nodes that belong to the helmet whereas the second one is composed by all the nodes on the bike and on the windshield. As can be observed in Figure 8 (right), only the nodes that fall inside the domain are selected (i.e. the encap domain, as the optional selection encaps, limits the action of the “on surface” selection). For the first set a rigid movement is imposed (a rotation about driver ankles or a displacement along driver neck) whereas for the second set a zero rigid movement is imposed to preserve the original shape of the bike components. The remaining nodes that fall inside the domain (i.e. the fluid and the body of the driver) remain free to deform under the action of the morpher. Before accepting the solution a preview of both cases has been examined (see Figure 9 and 10). After the preview the worst combinations of the parameters (i.e. maximum driver rotation for maximum and minimum driver height) have been tested, obtaining in both cases an acceptable quality mesh.

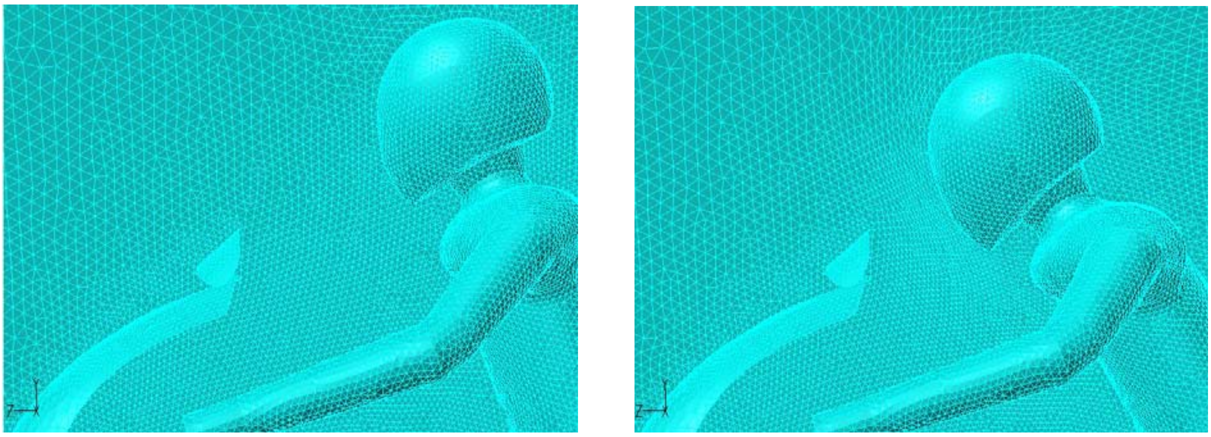


Figure 9: The mesh is morphed to change the driver angle of 15 degrees with respect to vertical axis

.

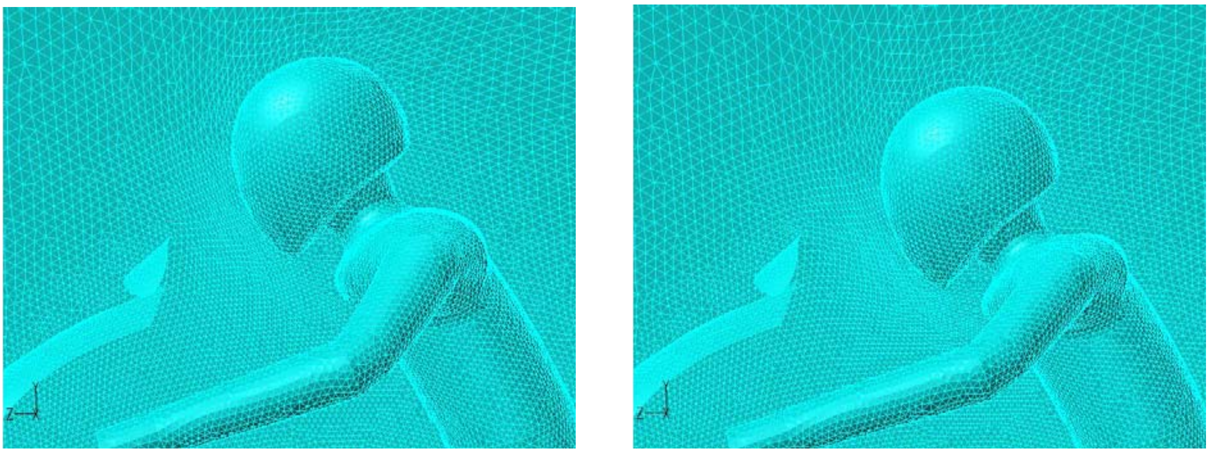


Figure 10: The mesh is morphed to change the driver height (5 cm), note that in this case the 15 deg hunched driver configuration has been used as the starting mesh.

3.4 Results

The first analyses aim to compare the standard windshield with the variotouring one. In the following figure the vectors plot of the velocity on the symmetry plane are shown.

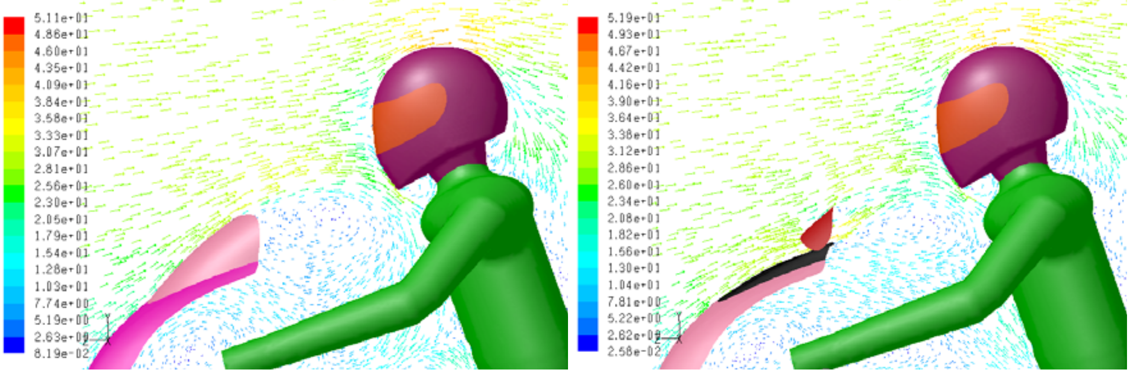


Figure 11: Flow field speed on the symmetry plane, original windshield and varioutouring.

Although the variotouring screen has a reduced size, a similar path of flow is observed on the driver and also the loads are quite similar (the horizontal load on the helmet is 3.2% smaller using the variotouring). The variotouring has an adjustable deflector and its performance can

change. In order to quantify this effect, five positions of the deflector have been considered changing its angle in the range +/- 10 deg (0 deg is the reference position of Figure 11). To better understand the interaction between deflector adjustments with driver position and size, three heights of the driver have been considered (+/-5 cm with respect to the presented baseline) and three driver angles (0 deg, 7.5 deg and 15 deg with respect to the baseline). A total of 45 simulations have been carried out. Thanks to the use of the morpher only three models were needed (one for each driver height, obtained morphing the reference mesh). The calculation of the 15 combinations analysed for each model has been automated using the multi morph feature to combine driver angle with deflector angle.
The results are summarized in Figures 12 13 and 14. In each plot three curves are exposed (one for each driver height) plotting the load versus the deflector angle considering the driver angle as parameter. It is worthy of notice that an improvement (i.e. a reduction of the load) with respect to the reference case (that performs similar to the original screen) can be obtained acting on the deflector angle. The vertical load can be reduced acting on the deflector and the optimum angle depends on the driver height and angle the load is higher for driver of reduced heights. The horizontal load on the helmet is higher for driver of increased heights (higher exposition to the flux) and decreases monotonically with the deflector angle. A quite similar behaviour can be observed for the total horizontal load acting on the driver.

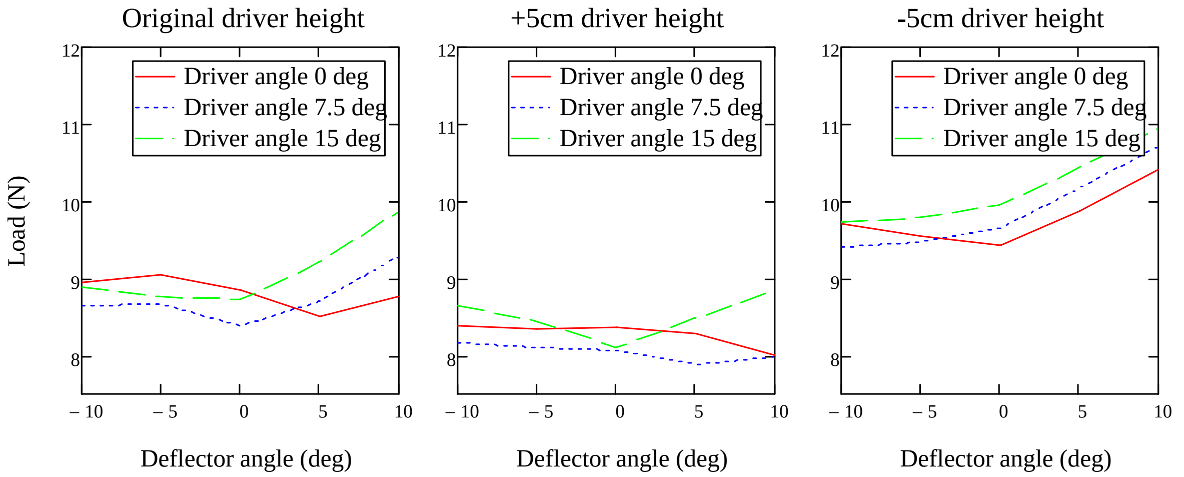


Figure 12: Vertical load on the helmet.

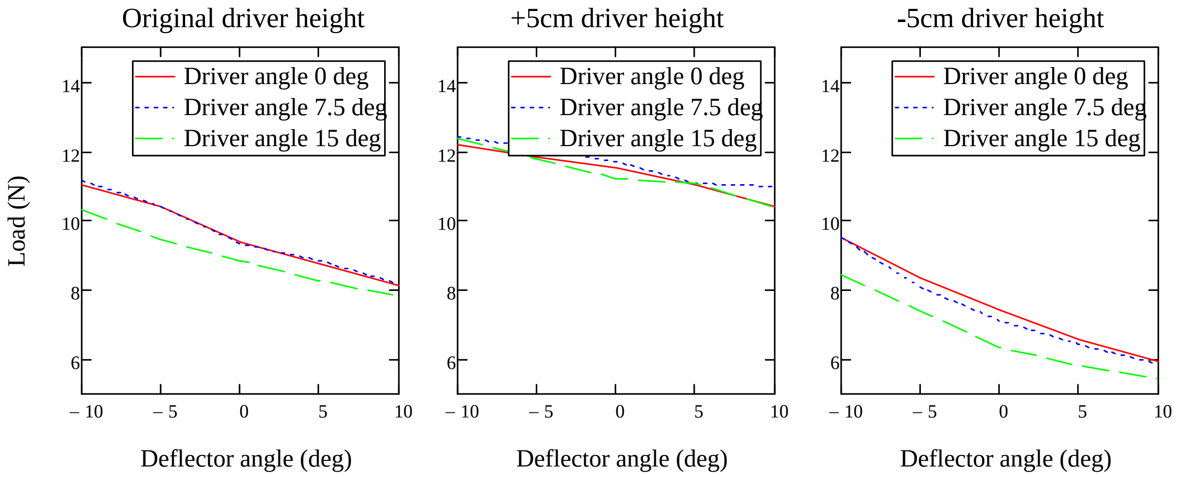


Figure 13: Horizontal load on the helmet.

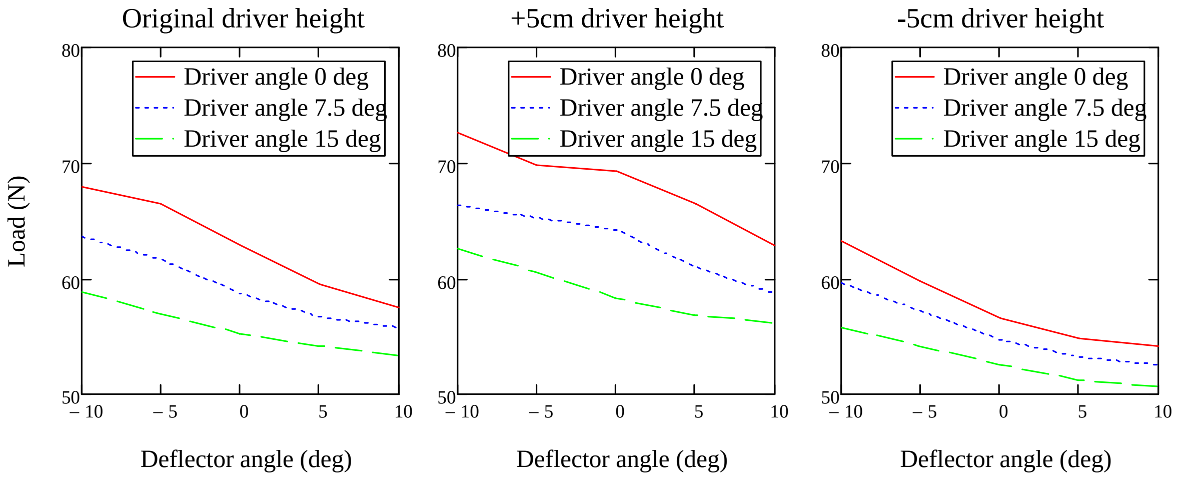


Figure 14: Horizontal load on the driver.

4. CONCLUSIONS

In this paper the application of RBF Morph tool for the CFD optimisation of a windshield has been presented. The tool has proven to be very useful for industrial applications and allowed to successfully managing all the desired configurations. The windshield optimisation project is still open and ongoing activities include the study of the effect of screen and deflector shape, the definition of further comfort parameters (turbulence intensity, transitory analysis). Experimental testing of a new prototype defined according to the presented design procedure is also scheduled.

5. REFERENCES

[1] Jakobsson, S.; Amoignon, O., Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization Computers and Fluids Volume: 36, Issue: 6, July, 2007, pp. 1119-1136.

[2] de Boer, A.; van der Schoot, M.S.; Bijl, H., Mesh deformation based on radial basis function interpolation Computers and Structures Volume: 85, Issue: 11-14, June - July, 2007, pp. 784-795.

[3] ARToolKit http://www.hitl.washington.edu/artoolkit/.

[4] Richard Hartley and Andrew Zisserman (2003). Multiple View Geometry in computer vision. Cambridge University Press.

[5] RBF Morph www.rbf-morph.com.

[6] MRA-Klement Gmbh www.mra.de.

[7] Bricomoto www.bricomoto.it.

6. ACKNOWLEDGEMENTS

The authors would like to express their acknowledgments to Johannes Klement of MRA-Klement GmbH for the information provided about experimental optimization of the windshield and to MRA for the sponsorship fund provided. The windshields used for the experiments have been donated by Bricomoto.