Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

SHAPE Project - Design Methods:

Coupled Sail and Appendage Design Method for Multihull Based on Numerical Optimisation

Ubaldo Cella[a*], Francesco Salvadore[b], Raffaele Ponzini[b]

[a]Design Methods (www.designmethods.aero)
[b]CINECA

Abstract

An optimisation procedure for catamarans sail plan and appendages is descripted. The method integrates a parametric CAD model, an automatic computational domain generator and a Velocity Prediction Program (VPP) based on a combination of sail RANS computations and analytical models. The sailing speed and course angle are obtained, with an iterative process, solving the forces and moment equilibrium system of equations. Hull forces analytical formulations were developed and tuned against a matrix of CFD solutions. The appendages aerodynamic polars are estimated applying preliminary design criteria from aerospace literature. The procedure permits to find the combination of appendages configuration, rudders setting, sail planform, shape and trim that maximise the VMG (Velocity Made Good). Within the SHAPE programme, the possibility to implement the procedure using Open-Source software is investigated.

Nomenclature

β	=	Leeway angle	L_H	=	Hull side force (parallel to the sea plane)
β_A	=	Apparent wind angle	S	=	Reference surface
β_T	=	True wind angle	V	=	Boat speed
δ	=	Appendage dihedral angle	V_A	=	Apparent wind speed
θ	=	Mast side setting angle	V_T	=	True wind speed
φ	=	Heeling angle	W_{BE}	=	Boat empty weight
ρ_w	=	Sea water density	W_{BO}	=	Boat operative displacement
D	=	Drag	W_M	=	Crew weight
D_H	=	Hull drag	d	=	Distance between hulls centreline
F_B	=	Boat aerodynamic resistance	h	=	Appendage aerodynamic centre
F_M	=	Crew aerodynamic resistance	h_B	=	Height of boat centre of gravity
F_h	=	Sail heeling force	h_h	=	Arm of sail aerodynamic resultant
F_t	=	Sail thrust force	l_M	=	Arm of crew righting moment
L	=	Lift			

[*] Corresponding author. E-mail address: ubaldo.cella@designmethods.aero

1. Introduction

Design Methods is the name of the engineering firm created by Ubaldo Cella (principal author) after around fifteen years spent gaining competences and experience within European research projects in the aerospace field and in aircraft developing programs. The mission of *Design Methods* is to provide multidisciplinary engineering consulting and design services to industries and design teams supporting them with highly specialised competences on aerodynamic design, CAE analysis, software development, CAD modelling, numerical optimisation environment and customised design tools development including FSI (Fluid-Structure Interaction) analyses. The company operates in aerospace, automotive and marine fields and can provide assistance in all phases of design processes with services as conceptual definition, preliminary design, software development, development of preliminary design codes, aerodynamic design, experimental tests in towing tank and wind tunnel.

The objective of the activity reported in this paper is the development of a CFD based catamaran sail shape and appendages optimisation procedure with the specific task of investigating the possibility to adopt open-source software in all components of its environment. In the company view, this project has the double objective to improve its awareness on open-source engineering tools and to demonstrate the *Design Methods* capability to transfer its competences in engineering applications other than aerospace.

The method here proposed integrates a parametric geometry model, an automatic computational domain generator and a VPP based on a combination of CFD computations and analytical models. Sailing speed and course angle are obtained, with an iterative process, solving the forces and moments equilibrium system of equations. The hull forces are modelled by empirical analytical formulations whose coefficients are tuned against a matrix of known solutions of the isolated demihull. This model provides a very fast evaluation of the forces at a given velocity, displacement and leeway angle. Dagger boards and rudders are modelled as wings. Their aerodynamic polars are estimated applying preliminary design criteria from the aerospace literature. The closure of the equations system is assured by the sail forces and the position of the aerodynamic centre of effort provided by RANS (Reynolds Averaged Navier-Stokes) computations. The analysis tool is integrated in a numerical optimisation environment which permits to find the combination of appendages configuration, rudders setting, sail planform, shape and trim that maximise the boat VMG (Velocity Made Good). The test case used for the method development is an A-Class catamaran.

A preliminary version of the method, developed using commercial software, was already available at a good stage of development. It constituted the starting framework on which the new procedure, here detailed, was built. The *CINECA* consortium (which is the PRACE centre selected to support this activity) was requested to support the setup of the open-source CFD configuration, the modules integration in the optimisation environment and the improvement of the procedure efficiency.

The description of the method and an image representative of the typical numerical analysis was selected by *ANSYS* as “best-in-class” winners of the *2016 Hall Of Fame Competition* in the corporate category.

The specific SHAPE overall goals are:

- to investigate the possibility to replace commercial codes with open-source software;
- to demonstrate the capability to efficiently face computationally costly problems within HPC environments;
- to optimise the procedure to run within an economic hardware environment;
- to provide a demonstration of its value to the market by the application on the design of an A-Class catamaran sail.

The workplan of the project was divided into three main technical activities:

1. **Software selection** - This activity consisted in searching and proposing a list of open-source software, available within the “Open” community, to be used in the implementation of the procedure. From this list, the software (CAD system, CFD solver, mesh generation code and scripting languages) that best fit our requirements are selected.
2. **Optimisation procedure implementation** - This is the core of the project. In this task the complete workflow of the optimisation procedure is developed. It consists in developing all modules of the analysis tool (the parametric geometric model, the computational domain adaptation procedure, the CFD computation and the VPP model) and in their coupling by scripts that drive the automatic process within the optimisation environment.
3. **Optimisation of an A-Cat sail** - The optimisation procedure developed in the previous task is applied on a case of practical interest: the optimum definition of an A-Class catamaran sail setting. The aim is to verify the performance of the procedure and to fine-tune the configuration of the tool.

Due to lack of time, it was not possible to complete the coupling of the modules with the open-source based CFD analysis within the SHAPE programme. It was, however, manually replicated an optimisation process with a reduced number of variables, at a fixed velocity and course, in order to verify the operation of the CFD analysis workflow. The solutions were compared to the results of the commercial codes based environment.

This report is divided into three sections: The first introduces the VPP (Velocity Prediction Program) which couples the boat analytical model to the sail RANS solution, the second describes the development of the modules of the optimisation environment and the last reports its application on the test case.

2. Performance prediction model

A boat model, fully based on analytical formulations, is proposed. The objective is to provide a very fast and versatile tool able to estimate the boat characteristics with an accuracy suitable for an optimisation environment. The strength of this approach is the capability to easily parametrise several aspect of the boat components providing the possibility to involve in the optimisation a wide range of design variables: chord, draft, twist, setting, airfoil and planform of the appendages as well as a range of hull parameters. The boat model is developed in form of a function able to interact, in a comparative iterative process within the equilibrium equations system, with the sail RANS aerodynamic solution to constitute a Velocity Prediction Program (VPP).

A baseline version of the model was previously developed using *Matlab*. The language selected to develop the new VPP and to drive all the components of the procedure fell on *Scilab* due to the similitude of this language with *Matlab*.

2.1. Boat global forces and moment equilibrium

Figure 1 reports the orientation of the adopted reference frame and summarises the forces acting on the boat.

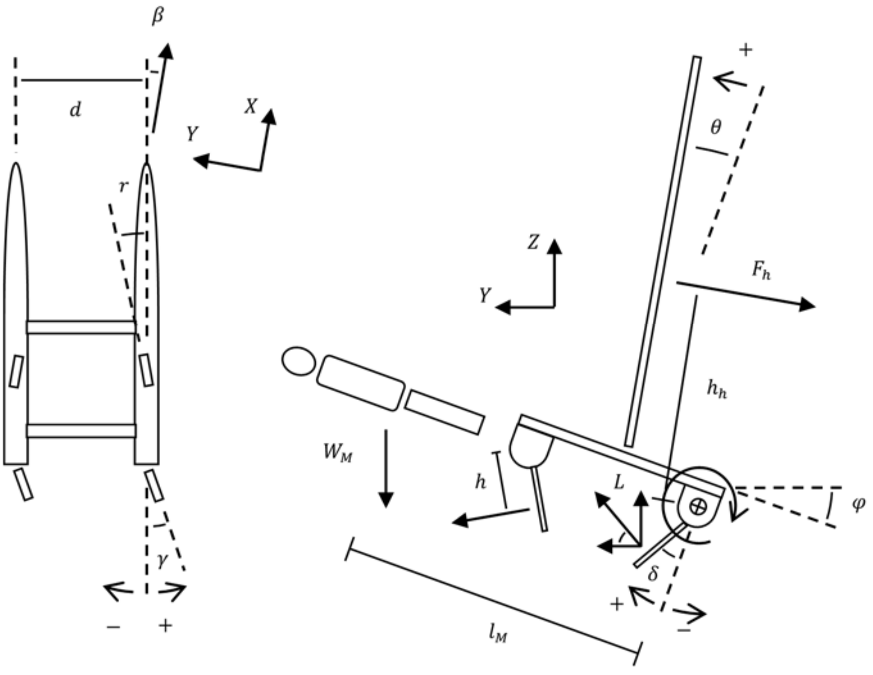

Figure 1: Scheme of forces *acting on the boat*.

The forces equilibrium equations of the complete boat, referred to a frame with the X axis aligned with the sailing direction and Z axis perpendicular to the water plane are:

X equilibrium

$$D_{TOT} = D_{D_D} + D_{D_U} + D_{R_D} + D_{R_U} + D_H + F_M + F_B = F_t \tag{1}$$

Y equilibrium (assuming to neglect the lift generated by the boat)

$$F_h\, cos(\varphi - \theta) = L_{D_D}\, cos(\varphi + \delta_D) + L_{D_U}\, cos(\varphi - \delta_D) + L_{R_D}\, cos(\varphi + \delta_R) + L_{R_U}\, cos(\varphi - \delta_R) + L_H \tag{2}$$

Z equilibrium

$$\begin{aligned} W_M + W_{BE} + F_h\, sin(\varphi - \theta) = W_{BO} + L_{D_D}\, sin(\varphi + \delta_D) + L_{D_U}\, sin(\varphi - \delta_D) + L_{R_D}\, sin(\varphi + \delta_R) \\ + L_{R_U}\, sin(\varphi - \delta_R) \end{aligned} \tag{3}$$

The subscripts $*_D$ and $*_R$ refer to dagger board and rudder. The subscripts $*_{D_D}$, $*_{D_U}$, $*_{R_D}$ and $*_{R_U}$ refer respectively to downwind dagger board, upwind dagger board, downwind rudder and upwind rudder.

The moment equilibrium around the centre of buoyancy of the downwind hull gives, assuming to neglect the influence of the mast setting on the centre of gravity:

M_X equilibrium

$$\begin{aligned} W_{BE}\frac{d}{2}cos\,\varphi + W_M l_M\, cos\,\varphi = F_h h_h + L_{D_D} h_{D_D} + L_{D_U}\big(h_{D_U} - d\, sin\,\delta_D\big) + L_{R_D} h_{R_D} + L_{R_U}\big(h_{R_U} - d\, sin\,\delta_R\big) \\ + W_{BE} h_B\, sin\,\varphi \end{aligned} \tag{4}$$

where the left hand side of the equation represents the maximum possible righting moment with the helmsman at trapeze.

It was decided to not involve the yaw and pitching moment equilibrium. The first in general impacts the rudder angle while the latter mainly influences the hull longitudinal setting. Both parameters can be controlled with an opportune sail rig/appendage centring and crew position. The assumption to neglect the two additional equilibrium equations is then considered acceptable at this stage.

2.1.1. Hull forces modelling

A wide literature is available and several strategies are offered to model a traditional monohulls sailing yachts, from databases solutions to accurate regression based polynomial expressions [1] [2]. Such methods are not valid in case of curved fast catamarans hulls. They, however, gave an input to develop simplified formulations customised to a typical A-Class cat hull shape and tuned against a matrix of isolated demihull CFD solutions, at several attitudes and leeway angles, in place of experimental data. Two formulations were developed for hull side force and for drag.

To estimate the hull side force (force laying in a plane parallel to the water plane and normal to the sailing direction), the bare hull is modelled as a lifting body as follows:

$$L_H = \frac{1}{2}\rho_w V^2 S_H \frac{\partial C_{L_H}}{\partial \beta}\beta \tag{5}$$

The reference surface S_H is the side projection, on the symmetry plane, of the submerged part of the demihull. It changes with displacement and is modelled by piecewise quadratic and linear curves approximating a set of values computed by a CAD system. According to the matrix of CFD solutions of the reference demihull, an analytical formulation for the curve slope $\frac{\partial C_{L_H}}{\partial \beta}$ is developed. It is tuned against the CFD data and is a linear function of displacement and exponential with velocity and leeway angle.

The bare demihull total resistance is expressed by:

$$D_H = \frac{1}{2}\rho_w V^2 S_{wet} C_T$$

The reference hull wet surface S_{wet} grows with displacement and is modelled with a formulation similar to the one adopted to model S_H. Also in this case the values to be approximated are computed by CAD. The model estimates the surface value starting from zero displacement in order to provide the procedure the capability to analyse also configurations in which the lifting contributions of the foils are predominant to the hull hydrostatic one. The total resistance coefficient is modelled as a combination of a friction and a residuary component [3]:

$$C_T = (1 + k)C_f + C_w$$

The form factor $(1+k)$ accounts to the over velocity generated by the thick shape of the body [4]. Its value has to be evaluated from literature or from a known bare hull drag value. The skin friction coefficient is estimated according to the ITTC-57 friction line expression:

$$C_f = \frac{0.075}{(log\, R_N - 2)^2}$$

where $R_N = \frac{VL_{wl}}{\nu}$ is the Reynolds number referred to the hull waterline length.

A significant simplification was chosen to model the residuary drag coefficient C_w. A combination of two quadratic formulations with Froude number, for speeds lower and higher than a critical value and linearly function of displacement, was adopted. The coefficients were tuned against the matrix of demihull CFD solutions.

Figure 2 compares the computed (by CFD) and modelled (by the developed analytical models) viscous and residuary drag. It is evident how the viscous component is dominant in most of speed range.

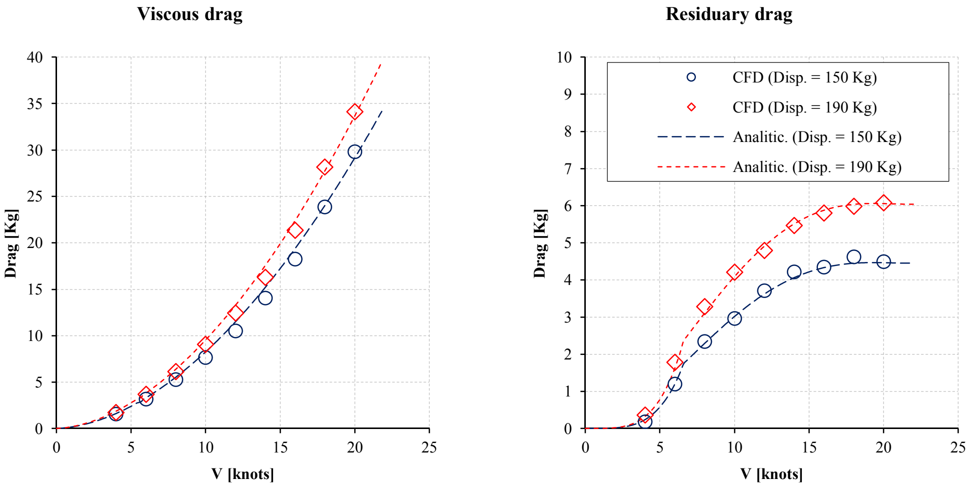

Figure 2: *Computed and modelled drag breakdown for the reference hull.*

2.1.2. Appendages forces modelling

Dagger boards and rudders are modelled as wings. The aerodynamic polars are estimated applying preliminarily design criteria from aerospace literature. The possibility to access to an experimental database of airfoils data, in UIUC .drg format [5], or to include a 2D analysis “on the fly” of an opportunely parametrised section geometry, by a coupled panel/boundary layer code [6], was also implemented.

The formulation of foils lift is:

$$L = \frac{1}{2}\rho_w {V_{eff}}^2 S C_L \qquad (6)$$

The lift coefficient C_L is function of lift polar slope, estimated applying empirical formulations used in aeronautics in the preliminary design phase [7], and leeway angle β. The effective velocity V_{eff} is the component of the boat velocity vector normal to the foil leading edge (for rectangular planform) and is the only velocity component responsible for the generation of lift. The spanwise component, in fact, does not affect the lift but only causes a shifting of the boundary layer [8].

The appendages drag formulation is:

$$D = \frac{1}{2}\rho_w V^2 S C_D$$

where the drag coefficient C_D is modelled by a quadratic analytical polar expression used in aeronautics and by a set of coefficients reported in literature as function of aspect and taper ratio [9].

Substituting the foils forces formulation in the equilibrium system of equations - equations (1), (2), (3) and (4) - and including the hull side force equation model eq. (5), we obtain, assuming the boat velocity V and the height of sail aerodynamic centre of effort $h_h$to be given as input, a system of five equations and five unknowns (D_{TOT}, F_H, W_{BO}, L_H and β). The solution of the equations system is implemented as a script function (written in Scilab) that produces as output the boat total resistance D_{TOT} and the sail heeling force F_H (which are the parameters to be compared with the CFD sail solutions) at a given speed V, centre of effort height h_h and set of parameters characterising the boat configuration.

2.2. Closure of the performance solution problem

No sail aerodynamic model is included in the function modelling the boat performance. As anticipated it requires, as input, two unknown parameters that are not related to the boat geometry or setting: the velocity of the boat V and the height of the sail centre of effort h_h. The closure of the problem is provided, with an iterative process, by the CFD aerodynamic solution of the sail at fixed sailing conditions.

Figure 14 in appendix describes the workflow to estimate the VMG for a given combination of parameters characterising the boat configuration. The procedure begins guessing an initial sailing speed V and course β_T. A CFD analysis, with the selected sail plan, shape and trim, is then run at these conditions. Sail forces and centre of effort are extracted and used to verify the equilibrium system. The verification consists in checking if the boat total resistance and the sail heeling force, computed by the analytical model for the given hull and appendage configuration, are equal respectively to the sail thrust force and the heeling force deriving from the CFD computation:

$$\begin{cases} D_{TOT} = F_{t_{CFD}} \\ F_h = F_{h_{CFD}} \end{cases} \tag{7}$$

If the two solutions are different new values of boat speed and true wind angle are selected. The CFD computation is restarted at the new conditions and the procedure is repeated until the equilibrium equations criteria are verified (within a prescribed tolerance). The VPP problem is completed with the production, as output, of the “Velocity Made Good” ($VMG = V \cos \beta_T$) value, which represents the performance of the boat [10] with the selected sail geometry (considered rigid) and appendages.

In order to speed up the VPP solution convergence, the procedure of sailing conditions exploration is split into two nested cycles. The principle is to use an external loop, which involves the RANS computation, to model analytical polars of the sail aerodynamics to indicate the inner search algorithm the direction where to find the sailing conditions that verify the equilibrium. The estimated sail aerodynamic model is then refined every external cycle until equilibrium is verified in both loops. The analytical polars formulations used to model the sail aerodynamics are similar to the one adopted to model the appendages. The sail reference freestream velocity and angle of incidence are respectively the apparent wind speed V_A and angle β_A which are obtained as function of the true wind speed V_T (for convention measured at 10 meters from the sea) and its angle β_T [11].

Since the drag polar is a quadratic formulation, the sail aerodynamics can be completely modelled after three iterations. The initial values of coefficients are then guessed. Figure 3 reports, for a typical A-Class sail plan, an example of the evolution of the polars computation during the progress of the first three iterations and the estimation of the values to be used for the computation of the sailing conditions in the fourth iteration (green circles). If the sail aerodynamic conditions fall in the linear region of the lift curve three iterations are in general sufficient to converge. If not, the reported analytical polars formulation are no more valid. The quadratic formulations, with which the non-linear sail aerodynamics is modelled in the following iterations, simply constitute interpolating curves whose coefficients have no particular physical meaning.

The searching criterion of the aerodynamic coefficients in the inner cycle is driven by an optimisation procedure, based on the Nelder-Mead Simplex algorithm [12], whose objective function is the minimisation of the differences between the forces derived from the boat analytical model and CFD computation in the equilibrium verification criteria:

$$Obj.Func. = \left|D_{TOT} - F_{t_{CFD}}\right| + \left|F_h - F_{h_{CFD}}\right|$$

The values of V and β_T that satisfy the equilibrium are used in the next external loop where the sail polars are update. The iterations continue up to the satisfaction of the equilibrium system in both loops. When convergence is reached, the boat VMG is computed and produced as output.

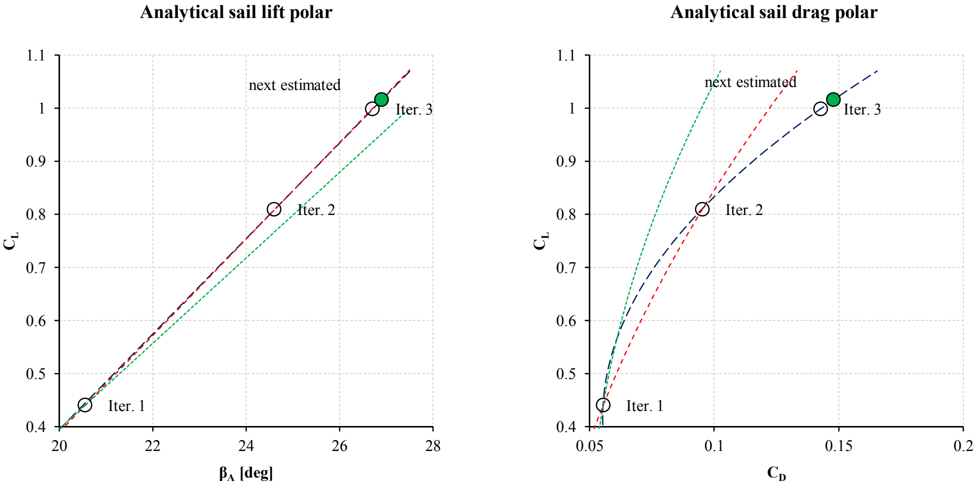

Figure 3: *Example of sail analytical polars computation progress.*

It was experienced that the RANS computations required to obtain a convergence rarely was higher than four or five (if sail is not stalled or, in general, if separations are not too large). Furthermore, a restarting procedure from the previous solution and a progressive reduction of the CFD solver number of iterations, was implemented. This strategy showed to be very efficient in boosting the convergence but its robustness is related to the capability to select starting sailing conditions as realistic as possible. The procedure fails in case of sudden sail separation. A check if complete stall occur has been implemented in order to reject such solutions.

3. Optimisation environment

The above described performance prediction procedure has been integrated in an optimisation environment in which the optimal sail plan, trim and appendage configuration is searched. The method integrates, in an automatic process, a sail parametric CAD model, a computational domain generation module, the RANS analysis and the VPP model. A set of open-source software candidates for each module was selected.

3.1. Open-source parametric geometric module

The selected strategy to parametrise the computational domain is based on a parametric CAD geometry update and in the CFD mesh regeneration. The chosen open-source software, to be used to perform the task of parametrise the geometry, is *FreeCAD*[†] [13]. Citing its user manual:

> *FreeCAD is a general purpose parametric 3D CAD modeller. The development is completely opensource (LGPL License). FreeCAD is aimed directly at mechanical engineering and product design [...]. FreeCAD features tools similar to Catia, SolidWorks or Solid Edge, and therefore also falls into the category of MCAD, PLM, CAx and CAE. It is a feature based parametric modeller with a modular software architecture which makes it easy to provide additional functionality without modifying the core system [...]. FreeCAD makes heavy use of all the great open-source libraries that exist out there in the field of Scientific Computing. Among them are OpenCascade, a powerful CAD kernel, Coin3D, an incarnation of Open Inventor, Qt, the world-famous UI framework, and Python, one of the best scripting languages available. FreeCAD itself can also be used as a library by other programs.*

FreeCAD has the capability to be totally controlled by Python scripts in command line mode without GUI interface. The set of geometric parameters are managed by the procedure and used as variables in a Python script to generate the model. The script is in charge also to export the model in IGES (Initial Graphics Exchange Specification) format. A further conversion in Stereo Lithography interface (STL) format, which is the format required by the selected open-source CFD mesh generator, was performed adopting the SALOME platform [14]. This step was required to improve the compatibility with the CFD module.

The modelled sail plan consists in a single mast/mainsail configuration. The CAD parameters were selected with the aim to investigate the largest possible range of geometries. Traditional sail plans, wing masts or rigid sails with

[†] The development of the parametric sail plan geometry module was supported by the Engineering department of the University of Messina.

a small portion of flexible sail can be generated. The sail is built by a loft surface through a foot, an intermediate arbitrarily positioned and a head curve that are used as control sections. The luff curve is used as guide. In the similar manner, the mast is generated from three geometries at the same stations. The planform is controlled by reference surface, aspect ratio, taper ratio and by other parameters that give the possibility to investigate any kind of shape. The examples in Figure 4 give the sense on the flexibility of the parametric model.

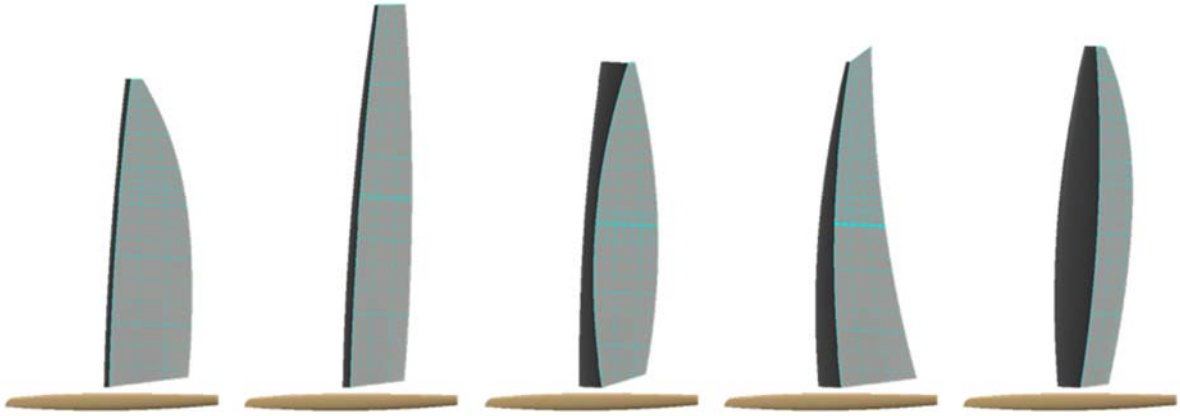

Figure 4: *Examples of sail planforms that can be generated by the parametric CAD module.*

The sail sections are modelled by cubic Bezier curves. The first point of the control polygon is connected to the mast luff, the last one coincides with the leech of the sail. The four coordinates of the two intermediate control points are parameters of the geometry (red polylines in Figure 5). The mast sections are generated by spline curves controlled by three parameters: the tangent tension at the leading edge, the tangent tension at the luff point and the angle between the latter tangent and the symmetry plane of the mast. The spanner and the three sections angle are sail setting parameters. The input reference surface area is kept unchanged. After the geometry creation, the final sail area is measured and the loft surface cut in order to restore the required value.

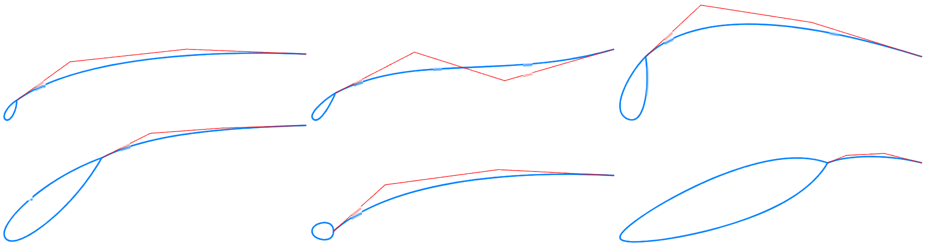

Figure 5: *Examples of mast/sail sections that can be generated by the parametric CAD module.*

3.2. Baseline sail CFD analysis module

The baseline procedure is linked to commercial codes to provide the CFD solution on the sail. A well consolidated mesh update procedure, and widely applied by the principal author to several aerodynamic optimisation problems [15], is currently implemented. It is based on an automatic generation of a structured hexahedral multiblock mesh using *ANSYS ICEM/CFD* and in the run of the CFD *ANSYS Fluent* solver. Due to the complexity of the geometry (all the boat, included the helmsman, is modelled) a mixed strategy, in the mesh generation, was adopted. The domain was divided in several regions with common boundaries and each region was meshed applying the more appropriate strategy. A structured CH grid topology was created in a limited volume around the sail and dimensioned to envelope the full range of possible geometries. An unstructured hybrid prism/tetra mesh was generated in a volume between the sail structured mesh boundaries and the water plane. The sail/boat mesh assembly is contained in a box, four boat lengths large and tall, in which another hybrid prism/tetra mesh portion was generated. The remaining volume was meshed with hexahedral cells growing toward the top with a progression aimed to better model the inflow air boundary layer. The full domain is 10 boat lengths wide and is extended 10 boat lengths upstream and downstream the model. In the final domain resulting from the assembly, the several parts are connected by zonal interface boundaries in which a simple "flow-through" condition between the non-conformal mesh zones is imposed. The first image of Figure 6 shows the assembly of the parts with the interfaces in evidence. The other two images detail the structured grid around the sail which is the only mesh part subjected to update every optimisation iteration. Figure 15 in appendix reports the final assembly of the mesh. The total dimension is around half millions of cells. This value was selected after a mesh sensitivity analysis. It was evaluated as a reasonable compromise between accuracy and computational costs with the view of being an analysis suitable for an optimisation process in which is more important the difference between the configurations than the absolute accuracy of the analysis.

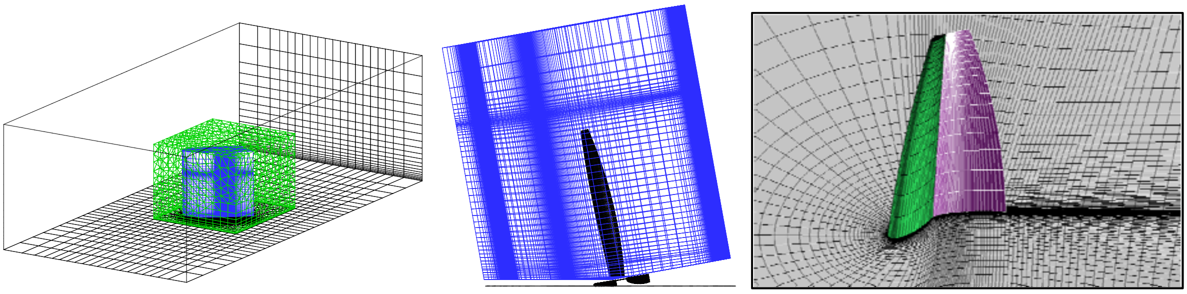

Figure 6: *Detail of the parts of the computational domain.*

The fluid dynamic computation consists in a steady fully turbulent RANS analysis in which the sail heeling and thrust forces, in windward sailing conditions, together with the resultant aerodynamic centre of pressure, are extracted. The two equation k − ω Shear Stress Transport (SST) turbulence model of Menter [16] was used. Wall Functions were applied to model the wall boundary layer. The boat is moving on a local reference frame at the given speed V and direction β_T respect to the true wind. These values are updated iterating with the analytical boat model in a process that constitutes the VPP module that provides the performance of the boat with the selected geometric configuration. The far field consists in an inlet boundary condition around the boat, in which the wind boundary layer velocity profile in the absolute reference frame is imposed [17], and a pressure outlet on the boundary behind. Figure 16 in appendix reports a typical solution obtained on an A-Class catamaran sailing windward with 10 knots of true wind speed at 10 meters from the water plane. The streamlines evidence the structures of the sail tip and root vortices. The wind boundary layer velocity magnitude is reported by a contour plot on a plane behind the boat.

Sails often exhibits separations in the region of the mast (Figure 17 in appendix reports an example of the typical evolution of this phenomenon behind a traditional A-Class rig). Furthermore, they usually have to perform in high lift conditions when trailing edge separation might also occur. The solver convergence histories could then be affected by unsteadiness or, in general, by irregularities. The only extraction of the last value could provide misleading information. A routine able to perform a more "critic" evaluation of the solutions was then developed. It consists in extracting a linearized interpolation of the last portion of the convergence history, in evaluating its slope and the solution maximum deviation from it and in extracting a value applying opportune constraints in order to reject unacceptable solutions. Figure 7 reports an example of how this filter works.

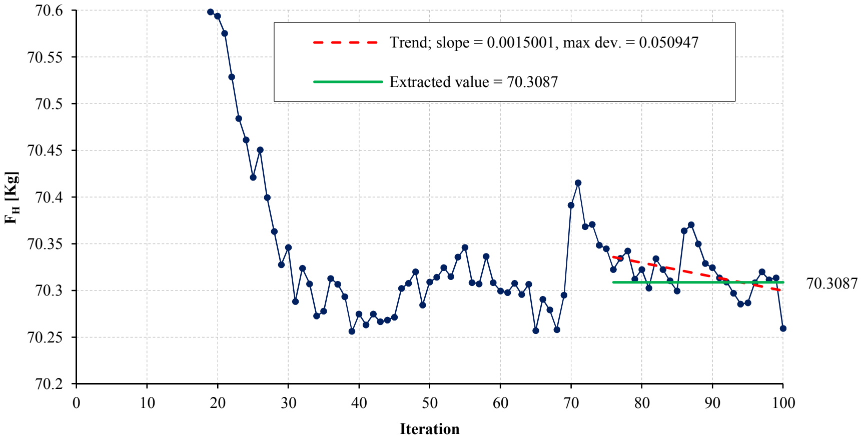

Figure 7: *Filter for solution extraction from convergence history.*

3.3. Open-Source based CFD analysis module

CINECA was in charge to replace the current sail analysis module with open-source codes. Due to consolidated internal expertise, the selected software packages to perform this task were totally based on open-source well-established codes as shown in Figure 8 here below. The detailed description of the activity is reported below highlighting a set of internal activities:

- CAD import and processing;
- Geometry meshing;
- Flow field solving;
- Data visualisation and post-processing.

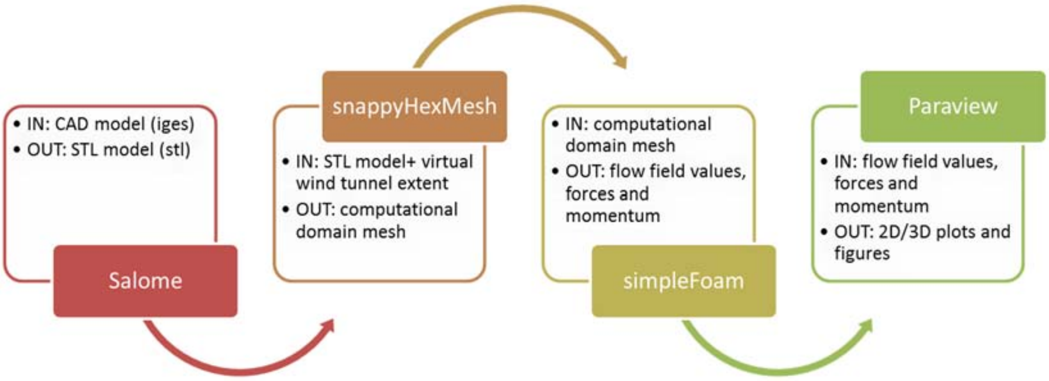
• IN: CAD model (iges)
• OUT: STL model (stl)
Salome
snappyHexMesh
• IN: STL model+ virtual wind tunnel extent
• OUT: computational domain mesh
• IN: computational domain mesh
• OUT: flow field values, forces and momentum
simpleFoam
Paraview
• IN: flow field values, forces and momentum
• OUT: 2D/3D plots and figures

Figure 8: *Schematic chart of the open-source based CFD analysis workflow.*

3.3.1. CAD import and processing

The starting CAD model provided by *Design Methods* is an IGES file (.igs) containing the boat (hull, platform, mast and sail) and the body of the sailor. The CAD also includes the artificial box defining the original computational domain. In order to produce a mesh using the *OpenFOAM* (OF) mesher *SnappyHexMesh* (SHM), we extract the solid bodies and export them in STL format. After some tests, we decided to use the open-source software Salome instead of the initially suggested *FreeCAD*. Salome can be freely downloaded and easily installed on a local machine.

The key-points of the procedure are described below:

1. [Salome-Geometry] Explode the initial CAD in faces
2. [Salome-Geometry] Rebuild solids joining faces. This operation was a bit tricky because sometimes Salome failed when joining faces or creating solids. To circumvent this problem, we needed many solids to represent the complex parts, e.g. the human body. The "Suppress holes" function was used to create these partial solids. Moreover, we decided to transform the platform giving it a finite height ("Extrude" function) and enlarging the planar surface a little bit to have a complete interpenetration with the hulls. This was done to ease the SHM meshing procedure later. The sail is still a surface object. The list of geometrical objects follows:
   a. Left hull
   b. Right hull
   c. Platform
   d. Sail (open surface, baffle)
   e. Mast
   f. Human body (composed by several surfaces)
3. [Salome-Mesh] Compute the surface mesh of the solids by creating a Mesh based on each of the previously created solids. We used the "Assign a set of hypotheses" option with the Automatic triangularisation choice. The reference length must be chosen to adequately represent the underlying geometry (something in the range 0.01:0.1 can be adequate depending on the geometry).
4. [Salome-Mesh] Now each solid can be exported as STL

At the end of the procedure we can check that the solids can be successfully used by OF using the *surfaceCheck* command provided by OF. We have to check carefully that the surfaces are closed except for the sail since is treated as baffle with zero-thickness.

Since the hulls and the platforms do not change across the simulations so we decided to merge them into a single STL file specifying different solid names labels according to the STL syntax. Also the different STL files necessary to describe the human body were merged this way into a single STL with internal labels.

3.3.2. Geometry Meshing

SnappyHexMesh is a part of the *OpenFOAM* application. For the project we used the *OpenFOAM 3.0.1* installed on the cluster and easily available through standard modules commands (*module load profile/advanced autoload openfoam/3.0.1*). In order to have also the additional swak4Foam utilities we also loaded the corresponding module with a similar syntax (*module load profile/advanced autoload swak4foam/hg_r2781*). For the configuration of the SHM and of the OF solver setups, we started from the motorbike tutorial case available in the tutorial directory of the OF installation. In the next description, the majority of the omitted details can be extracted from this tutorial case.

We first define and create the external box of the computational domain. A coarse but effective setting of blockMeshDict can be [-25:25,-25:25,0:30] domain with (15 12 9) as the number of points in each direction. Now blockMesh can create the initial mesh.

The configuration of SHM is done through the snappyHexMesh file. These are the basic settings used:

- In geometry we include all the STL -- hulls+platform, mast, sail, body -- and two refinement boxes -- one for the boat [-5:15,-5:5,0:12] and one for the boundary layer [-25:25,-25:25,0:1]
- The features contain the eMesh files for the hulls+platform, mast, sail geometries and specify a level 7 or 8. The eMesh files can be produced by means of the surfaceFeatureExtract command starting from the surfaceFeatureExtractDict file where the initial structure can be replicated for all the current STLs.
- The refinement surfaces define the named geometries with refinement levels of (6 8) or (5 7). The sail geometry must be specified as a faceType baffle, since it is a surface (open) geometry.
- The refinement regions define both refinement boxes. Level 3 per the refinementBox and Level 4 for the boundary layer box are adequate.
- The locationInMesh point is an arbitrary point inside the computational domain but not inside any solid
- In snapControls we increased the value of nSolveIter to 50 and the value of nRelaxIter to 50. We did not check the relevancy of these changes, though.
- In the layers dictionary we added all our geometries. However, the layers should not be important for our simulations (only 1 layer is generated and only on some boundaries).

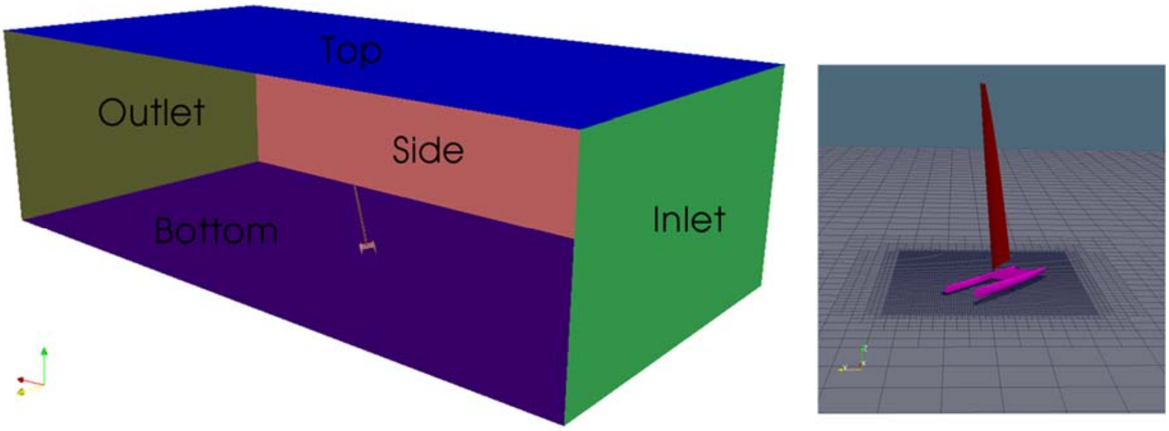

Figure 9: *Computational domain and mesh detail of the open-source based CFD configuration.*

The meshing execution is started using the *snappyHexMesh* command and can be performed in parallel. At the end we have to check that a valid mesh has been created. The *checkMesh* command is important in this context. The mesh can be also loaded into *ParaView* to quickly visualise its quality. It is also important to keep in mind that the output contained in the folder named “3” must be copied to the *constant/polyMesh* folder and that the boundary file contains the names of all the generated patches. Two patches are automatically generated for the sail geometry given its 2d structure. The resulting computational domain and mesh are showed in Figure 9.

3.3.3. Flow Field solving

The *simpleFoam* solver with $k-\omega$ SST model was used as solver for our simulations, as in the *motorBike* example cited above. The simulations have been conducted in the relative reference frame where the boat is stationary and the wind is composed by a physical component (tabulated boundary layer values) and a moving reference component compensating the stationarity of the boat. The groovyBC (https://openfoamwiki.net/index.php/Contrib/groovyBC) boundary conditions have been employed to implement the advanced boundary inlet conditions we need.

Some details on the involved configuration files are:

- system/controlDict:
    - endTime 500 is enough for our cases but a residual check is mandatory
    - writeInterval 100 to avoid an excessive number of output files
    - Include in the file the forces configuration: functions { #include "forces" }
    - Include in the file the usage of the groovyBC library: libs ("libgroovyBC.so");
- system/decomposeParDict required to run in parallel
    - A simple decomposition using 20 subDomains as (4 5 1) has been adopted
- system/forces specify the surfaces to be accounted into the force outputs. We included both the sail boundaries and the mast boundary. To replicate the results obtained with the commercial solvers, we also specified rhoInf value of 1.225.
- 0/U
    - inlet and sideleft are groovyBC with an interpolation formula built from the boundary layers tabulated values defined by Design Methods
    - sideright and outlet are zeroGradient
    - lowerWall is fixedValue at the moving reference frame value
    - upperWall is a slip condition
    - All the other boundaries are fixedValue with uniform (0 0 0) value
- 0/p
    - inlet and sideleft are zeroGradient
    - sideright and outlet are fixedValue (0)
    - upperWall is slip
    - All other boundaries are zeroGradient
- 0/k
    - inlet and sideleft are fixedValue with a finite value 0.1
    - sideright and outlet are inletOutlet with the finite value
    - upperWall is slip
    - All other boundaries are kqRWallFunction with the same finite value
- 0/omega
    - As k but using omegaWallFunction instead of kqRWallFunction and value 0.06
- 0/nut
    - Inlet, outlet, sideleft, sideright, upperWall are calculated with value 0
    - All other boundaries are nutkWallFunction with value 0

3.3.4. Data visualisation and post-processing

Post-processing, visualisation and velocity profile plotting were performed using VTK file format loaded into the *Paraview* (Kitware Inc.) visualisation software and using python basic scripting for 2D csv data plots.

A snapshot of typical result analysis, 3D and 2D plots, is shown in Figure 10.

For a given geometry and wind condition a preliminary validation has been performed comparing *OpenFOAM* results with the one obtained with the commercial code. According to the forces and momentum values the discrepancy was well bounded (values differing of about 3-5 %). For this reason we fixed the solver setup described here above and we use it by changing the CAD sail as it will be presented in the forthcoming of this document.

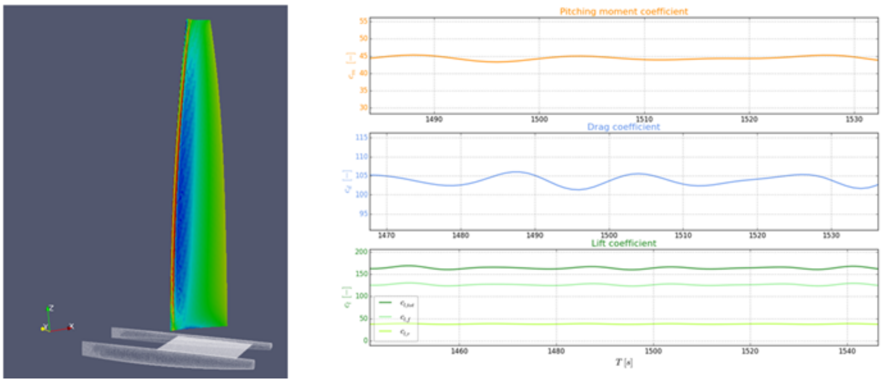

Figure 10: *Flow analysis, 3D and 2D plots.*

3.3.5. Automation setup for the DOE

To perform a DOE (Design of Experiments) analysis with several cases involved we implemented some automation improvements:

- Salome conversion from CAD to STL was implemented using a Python code which basically reproduces the tasks done using the Salome GUI. The translation from GUI to Python was quite easy given its dump history feature.
- A bash script has been written to perform the entire group of simulations starting from a set of IGES files, just running the produced script. The forces output results are automatically collected to a single summary file.
- The final results can be easily analysed and/or visualised using *Paraview*

3.4. Implementation of the optimisation environment

All the modules are managed by *Scilab* routines. Also the optimisation algorithms adopted are the ones implemented in the computing language developing an optimisation procedure already tested by the principal author for turbomachinery design [18].

Figure 11 sketches the workflow of the implemented optimisation procedure. After the CAD model update the process is guided to run a script that loads the new geometry, recomputes the mesh and exports the new grid in the solver format. The CFD configuration is then updated and the VPP module, described in the previous section, executed to provide the performance of the selected configuration. According to the solutions obtained, the optimisation algorithm selects a new combination of parameters and the cycle progress until the “Optimum” configuration is found.

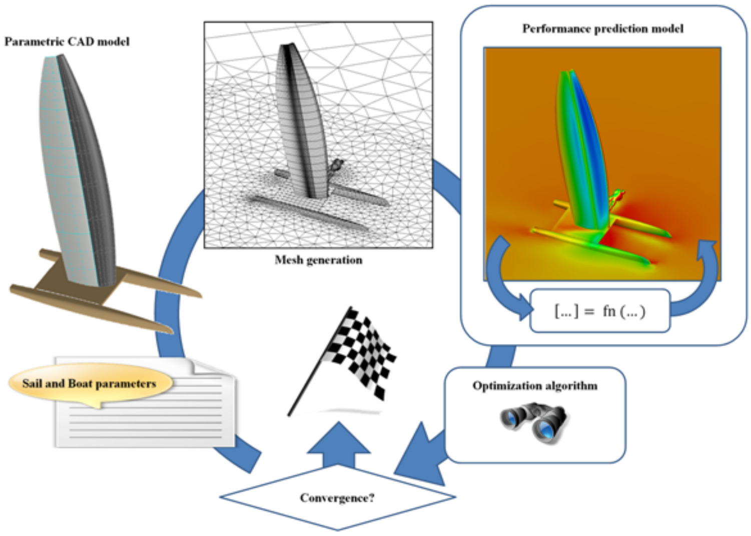

Figure 11: *Scheme of the optimisation procedure.*

More than 40 parameters can be selected to characterise the boat configuration. Such a wide range of potential design variables gives great flexibility in exploring innovative solutions. The optimisation environment, furthermore, provides the designer a powerful tool that supports the exploration of their limits.

4. Application on a test case

The optimisation workflow was tested on a simplified configuration. It consisted in defining a DOE matrix, using two design variables, and in the following definition of a response surface on which to search the optimum. The test has the double objective to find out potential criticisms of the workflow and to compare the performance of the *ANSYS Fluent* and the *OpenFOAM* CFD analysis based procedures. Due to lack of time it was not possible to complete the inclusion of the *OpenFOAM* analysis in the procedure. It was then decided to manually replicate the open-source based workflow excluding, at this stage, the interaction with the VPP module. *Design Methods* provided *CINECA* the geometries corresponding of the DOE table design points in IGES format and received the solutions, in terms of forces and moments, computed with *OpenFOAM.*

The test consisted in finding the optimum sail setting of an A-Class traditional rig at a boat speed $V = 10$ knots, a true wind angle $\beta_T = 45$ deg and a true wind speed $V_T = 10$ knots. The variables were the mast spanner angle (the angle between the mast chord and the boat symmetry plane) and the sail setting (angle between the sail chord at the base and the boat symmetry plane). It was decided a range of variation for the spanner variable from 35 to 50 degree with a step of 5 deg. The range of variation of the sail setting angle was from 1 to 7 degree with a step of 2 deg. The DOE table was then populated with 16 solutions. Figure 12 compares two extremes of the variables design space. Table 2 and Table 3 in appendix compare the forces and moments computed by *OpenFOAM* and *ANSYS Fluent* for all design points of the DOE table.

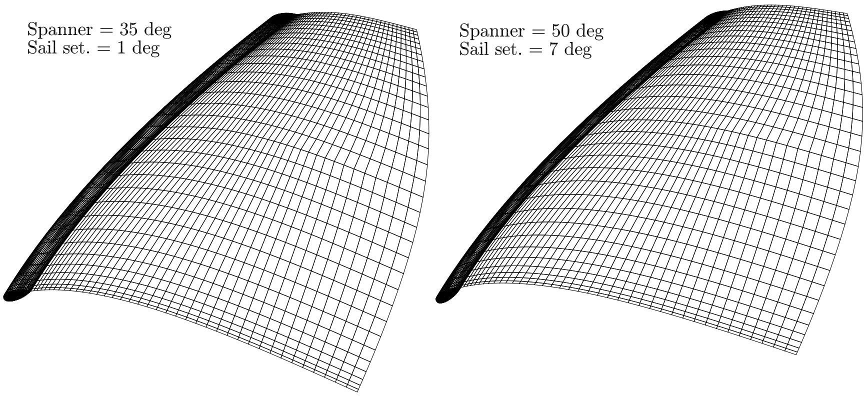

Figure 12: *Comparison between two extremes of the variables design space.*

The target of the optimisation was the maximisation of the sail thrust force. The objective function was defined as follows:

$$Obj.Func. = F_t = F_Y \sin\beta_T - F_X \cos\beta_T$$

where the input forces are referred to a frame aligned to the true wind speed direction.

Catamarans exhibits the maximum righting moment when the upwind hull is in flying condition (just outside the water). In equilibrium conditions, the heeling moment must be equal to the righting moment. Assuming the helmsman to be positioned at the trapeze, having a weight of 90 kilograms and an arm slighter higher than 3 meters, the maximum allowable righting moment, including the contribution of the boat weight ($M_{h_{max}}$), is around 3500 Nm. This value was implemented as an optimisation constraints as follows:

$$M_Y \sin\beta_T - M_X \cos\beta_T \leq M_{h_{max}}$$

Figure 13 compares the response surfaces computed with the solutions obtained by the two solvers. A second-degree polynomial formulation was sufficient to approximate the computed CFD solutions generating a residual always below 0.3%. As expected, the sail setting has larger impact on the objective function.

The largest differences between the two solutions are located in the regions toward the minimum values of the two variables which means in conditions of higher incidence of mast and sail elements when the rig is close to the stall conditions. The disagreements should be then related to a different evaluation of the separated regions.

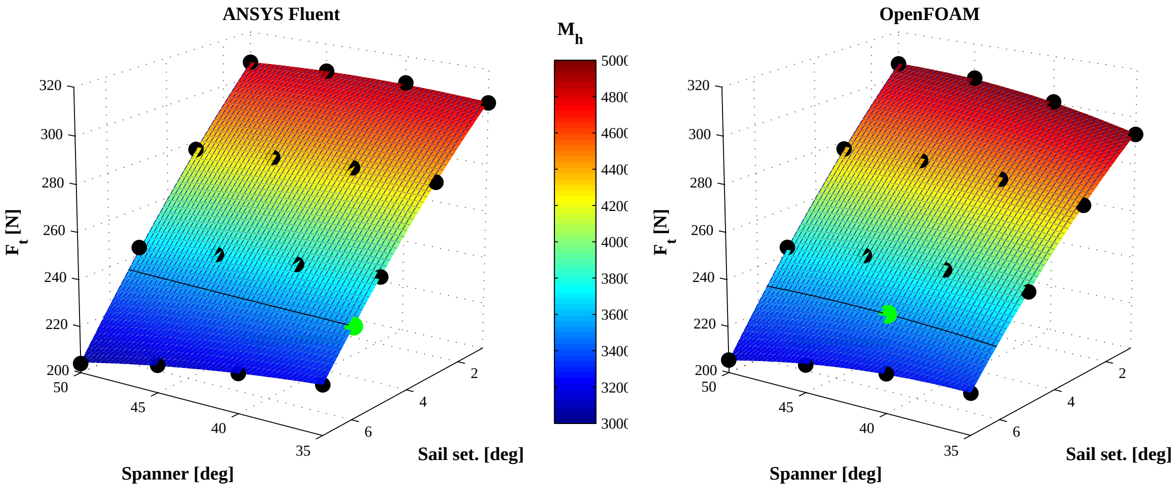

Figure 13: *Comparison of response surfaces obtained with the two solvers solutions.*

The colours of the response surfaces are associated to the value of the heeling moment. The black lines on the surfaces are the M_h isolines at 3500 Nm. The optimum solution has then to be searched along these curves. The green points indicate the position of the optimum found with the two methods. The optimum found using *ANSYS Fluent* is located on the boundary of the variables space since the isocurve do not have a maximum within this domain. The maximum found with the *OpenFOAM* based analysis procedure is in the middle of the spanner range. The Table 1 reports the two optima solutions.

	ANSYS Fluent	*OpenFOAM*
Spanner	35 deg	41.7 deg
Sail set.	5.9 deg	6 deg
Thrust force	238.2 N	232.5 N

Table 1: *Optima solutions.*

The *ANSYS Fluent* based analysis procedure ran on a workstation with 20 cores. The complete convergence was reached, for each design points, with 200 iterations in less than ten minutes. The *OpenFOAM* based procedure ran on HPC nodes each of them equipped with two Intel Xeon 2670 v2 2.5GHz. A single run required 15 minutes with 20 cores to complete the CAD setting, mesh generation, computing and solution extraction process. The computational costs of the two methods can then be in general considered comparable.

5. Conclusions

A numerical optimisation environment for catamarans sail plan and appendages, that couples a VPP based on analytical models and on a sail RANS computation, was developed. A preliminary baseline version of the procedure, fully developed with commercial software, was available ad used as a starting framework. The main objective of the SHAPE project was the implementation of the procedure using only open-source tools. *Design Methods* replicated the starting framework managing all the included modules by scripts written using the *Scilab* computing environment. The geometric parametric module was implemented by *Python* scripts used to drive *FreeCAD* to generate the CAD model. The analytical formulations, used to model the hull and appendages forces, were implemented in a form of independent functions and coupled to the aerodynamic solutions of the sail to solve the equilibrium system of equations of the boat in an iterative procedure. This procedure constitutes the VPP module that estimates the performance of the selected geometric configuration in terms of boat VMG. The sail aerodynamic solution is, in the baseline framework, provided by an automatic process based on the *ANSYS ICEM/CFD* mesh generator and the *Fluent* CFD solver. *CINECA* was in charge of supporting the setup of the open-source CFD configuration, the modules integration in the optimisation environment and the improvement of the procedure efficiency. A CFD analysis using *OpenFOAM*, suitable to replace the commercial solver, was setup on a reference configuration provided by *Design Methods* and compared to the implemented *Fluent* based CFD analysis. Its effective integration in the procedure was not possible due to lack of time. This activity will progress outside the SHAPE programme. In order to more deeply evaluate the performance of the solver, it was manually replicated the workflow on a simplified optimisation problem. *Design Methods* provided *CINECA* with a set of geometries, defined by a DOE table of a two-variables problem, and received back the solutions of *OpenFOAM* in terms of sail forces and moments to be compared with the *Fluent*-based analysis procedure. The two methods generated relatively similar optimal solutions (with a difference in the objective function in the order of 2%). With the exception of cases where significant separation is present (close to the maximum lift), *OpenFOAM* provided, in general, solutions rarely differed from Fluent more than 5% on the forces computation. Considering that an optimum solution it is expected to have no separations (or at least limited separated regions), it thus can be stated that *OpenFOAM* is a valid candidate to replace *Fluent* in the optimisation procedure. This assumption is also valid evaluating the two solvers computational requirements. Both codes complete an analysis of a case with attached flow in less than 15 minutes using 20 cpu.

References

[1] Gerritsma J., Onnink R., and Versluis A., "*Geometry, resistance and stability of the delft systematic yacht hull series*", In 7th HISWA Symposium, 1981.

[2] Keuning J. A., Onnink R., Versluis A., and Van Gulik A., "*The bare hull resistance of the delft systematic yacht hull series*", In International HISWA Symposium on Yacht Design and Construction, Amsterdam, 1996.

[3] Insel M. and Molland A., "*An investigation into resistance components of highspeed displacement catamarans*", Transaction of the Royal Institute of Naval Architects, (134):1 – 20, 1992.

[4] Hoerner S. F., "*Fluid-Dynamic Drag*", Hoerner Fluid Dynamics, Bricktown, New Jersey, 1965.

[5] University of Illinois. UIUC wind tunnel data on the web. Web site: http://m-selig.ae.illinois.edu/pd.html.

[6] Mark Drela, "*Xfoil, an analysis and design system for low reynolds number airfoils*", In Conference on Low Reynolds Number Airfoil Aerodynamics, University of Notre Dame, June 1989. doi: 10.1007/978-3-642-84010-4_1.

[7] Roskam J. and Lan C. T. E., "*Airplane Aerodynamics and Performance*", DARcorporation, 2000.

[8] Losito V., "*Fondamenti di Aeronautica Generale*", Accademia Aeronautica, 1991.

[9] Abbott I. H. and Von Doenhoff A. E., "*Theory of Wing Sections*", Dover Publications, New York, 1959.

[10] Larsson L. and Eliansson R. E., "*Principle of Sailing Yacht Design*", Adlard Coles Nautical, London, 1997.

[11] Claughton A., Wellicome J., and Shenoi A., "*Sailing Yacht Design – Theory*", Longman, 1998.

[12] Nelder J. A. and Mead R., "*Simplex method for function minimization*", The Computer Journal, 7:308 - 313, 1965.

[13] FreeCAD web page, http://www.freecadweb.org.

[14] SALOME, web page http://www.salome-platform.org.

[15] Cella U., Romano D., and Franchini F., "*Design and numerical optimization of winglets of a Piaggio aircraft*", In AIRTEC 5th International Aerospace Conference, Frankfurt, (Germany), 2 - 4 November 2010.

[16] Menter F. R., "*Two-equation eddy-viscosity turbulence models for engineering applications*", AIAA Journal, 32(8):1598 – 1605, August 1994.

[17] Musker A., "*Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer*", AIAA Journal, 17:655 – 657, 1979.

[18] Milanese G. and Cella U., "*Automotive axial flow fans: modeling and design*", In SCILABTEC2015, 7th International Scilab User Conference, Paris, (France), 21 22 May 2015.

Acknowledgements

Several people contributed directly and indirectly to this project. The adoption of the *ANSYS* software in the baseline procedure was possible thanks to the support of the university of Rome "Tor Vergata" and the "RBF Morph" software vendor. The authors wish to thank Marco Evangelos Biancolini for having provided the possibility to generate the required base of comparison with the *OpenFOAM* solutions. Special thanks are reserved to Filippo Cucinotta and Felice Sfravara form the Engineering department of the University of Messina. They actively contributed in the development of the parametric geometry module. The last thanks are for the ADAG Aerospace Engineering section of the university of Naples "Federico II" and in particular to Agostino De Marco which supported the generation of the CFD solutions database used for the hull analytical model development.

This work was supported by the EU's Horizon 2020 research and innovation programme (2014-2020) under grant agreement 653838.

Appendix

Variables		*Fluent*			*OpenFOAM*			*Difference*		
Spanner [deg]	Sail set. [deg]	F_X [N]	F_Y [N]	F_Z [N]	F_X [N]	F_Y [N]	F_Z [N]	ΔF_X [%]	ΔF_Y [%]	ΔF_Z [%]
35	1	475.2	908.0	-161.0	504.2	918.2	-167.3	6.1	1.1	3.9
35	3	423.2	821.8	-144.5	444.5	829.6	-149.1	5.0	0.9	3.2
35	5	368.9	726.7	-126.8	385.7	734.7	-130.7	4.5	1.1	3.0
35	7	315.0	627.1	-108.7	328.3	635.4	-112.1	4.2	1.3	3.1
40	1	476.0	912.6	-161.6	500.9	925.9	-167.3	5.2	1.5	3.6
40	3	422.2	820.1	-144.1	440.9	832.0	-148.7	4.4	1.4	3.2
40	5	367.1	721.7	-126.1	382.5	733.8	-130.1	4.2	1.7	3.2
40	7	312.1	618.6	-107.3	325.5	631.7	-111.3	4.3	2.1	3.7
45	1	475.0	910.8	-161.2	497.8	929.3	-167.1	4.8	2.0	3.7
45	3	420.0	815.1	-143.3	438.1	831.3	-148.2	4.3	2.0	3.4
45	5	364.2	714.3	-124.8	379.9	729.4	-129.3	4.3	2.1	3.5
45	7	308.2	607.6	-105.7	322.7	622.1	-110.1	4.7	2.4	4.1
50	1	472.5	906.1	-160.3	495.1	927.7	-166.5	4.8	2.4	3.9
50	3	417.3	808.3	-142.3	435.6	826.9	-147.4	4.4	2.3	3.6
50	5	360.4	704.4	-123.4	376.8	720.7	-128.0	4.5	2.3	3.7
50	7	302.3	590.8	-103.4	319.1	609.6	-108.4	5.5	3.2	4.8

Table 2: *DOE table - Forces solutions*

Variables		*Fluent*			*OpenFOAM*			*Difference*		
Spanner [deg]	Sail set. [deg]	M_X [Nm]	M_Y [Nm]	M_Z [Nm]	M_X [Nm]	M_Y [Nm]	M_Z [Nm]	ΔM_X [%]	ΔM_Y [%]	ΔM_Z [%]
35	1	-4532.8	2314.3	270.5	-4540.0	2443.4	284.9	0.2	5.6	5.3
35	3	-4087.6	2059.7	257.4	-4082.5	2151.8	268.9	-0.1	4.5	4.5
35	5	-3594.0	1794.2	241.6	-3595.4	1864.2	252.0	0.0	3.9	4.3
35	7	-3077.0	1526.3	224.1	-3086.1	1583.1	234.2	0.3	3.7	4.5
40	1	-4556.2	2321.7	274.4	-4578.7	2426.3	288.7	0.5	4.5	5.2
40	3	-4073.6	2055.1	259.5	-4092.7	2133.1	271.8	0.5	3.8	4.7
40	5	-3561.2	1781.9	242.4	-3587.6	1847.3	254.4	0.7	3.7	4.9
40	7	-3019.2	1507.8	224.0	-3063.2	1568.3	235.9	1.5	4.0	5.3
45	1	-4547.6	2313.6	275.5	-4592.0	2409.9	289.9	1.0	4.2	5.2
45	3	-4041.9	2041.0	259.7	-4085.8	2118.6	272.9	1.1	3.8	5.1
45	5	-3515.0	1764.6	241.9	-3558.3	1833.6	254.5	1.2	3.9	5.2
45	7	-2950.2	1482.6	221.8	-3005.9	1552.9	234.5	1.9	4.7	5.7
50	1	-4513.2	2300.6	274.6	-4578.0	2395.0	289.5	1.4	4.1	5.4
50	3	-3996.5	2024.7	258.9	-4056.4	2104.7	271.9	1.5	4.0	5.0
50	5	-3451.1	1742.4	239.9	-3507.1	1815.9	252.6	1.6	4.2	5.3
50	7	-2854.4	1448.1	218.8	-2933.4	1532.6	231.2	2.8	5.8	5.6

Table 3: *DOE table - Moments solutions*

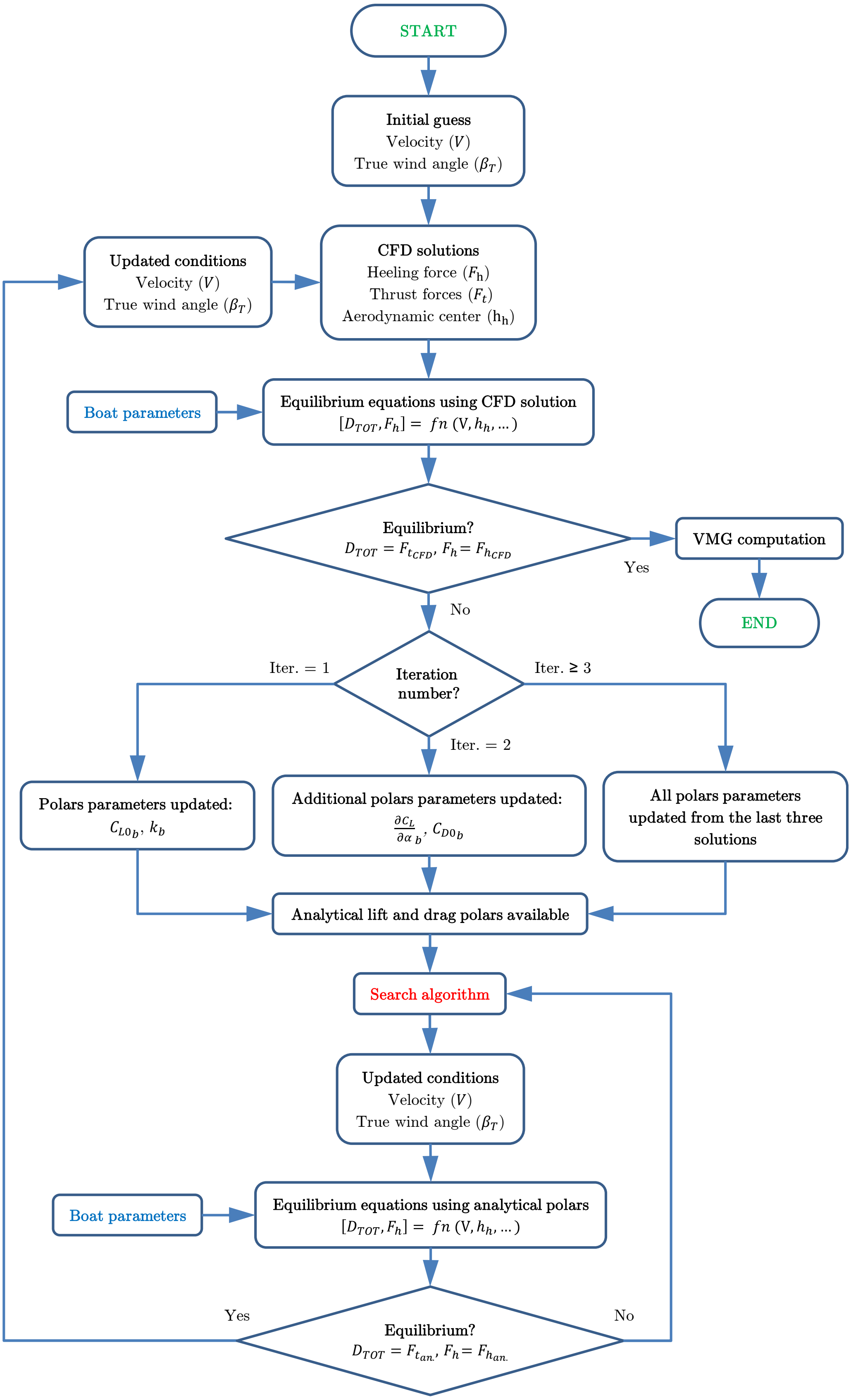
START
Initial guess
Velocity (V)
True wind angle (β_T)
Updated conditions
Velocity (V)
True wind angle (β_T)
CFD solutions
Heeling force (F_h)
Thrust forces (F_t)
Aerodynamic center (h_h)
Boat parameters
Equilibrium equations using CFD solution
$[D_{TOT}, F_h] = fn\ (V, h_h, ...)$
Equilibrium?
$D_{TOT} = F_{t_{CFD}}, F_h = F_{h_{CFD}}$
Yes
VMG computation
END
No
Iter. = 1
Iteration number?
Iter. ≥ 3
Iter. = 2
Polars parameters updated:
C_{L0_b}, k_b
Additional polars parameters updated:
$\frac{\partial C_L}{\partial \alpha}_b, C_{D0_b}$
All polars parameters updated from the last three solutions
Analytical lift and drag polars available
Search algorithm
Updated conditions
Velocity (V)
True wind angle (β_T)
Boat parameters
Equilibrium equations using analytical polars
$[D_{TOT}, F_h] = fn\ (V, h_h, ...)$
Yes
No
Equilibrium?
$D_{TOT} = F_{t_{an.}}, F_h = F_{h_{an.}}$

Figure 14: *Flow chart of the VPP module.*

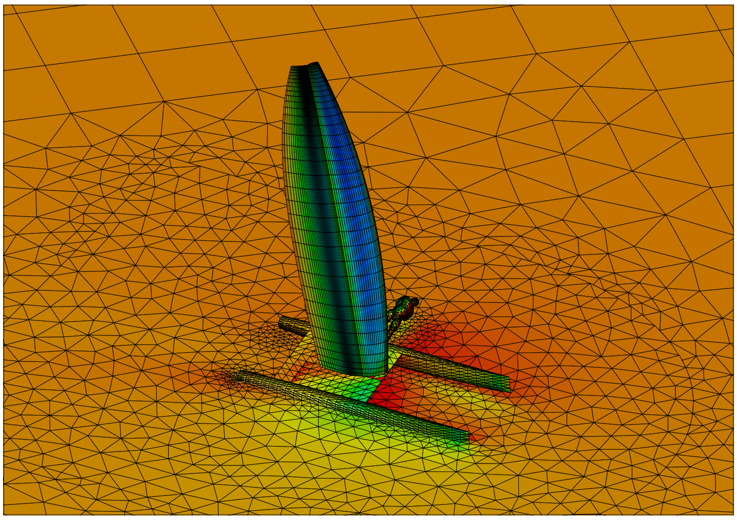

Figure 15: *Final assembly of the ANSYS Fluent computational domain.*

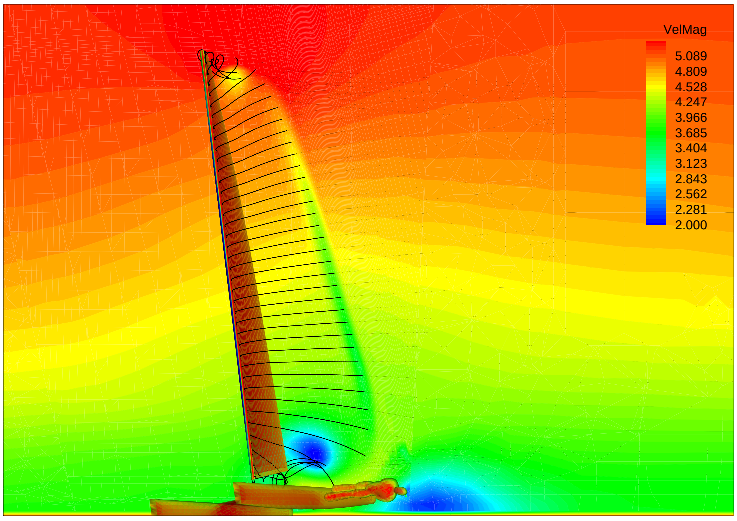

Figure 16: *Typical CFD solution on an A-Class catamaran.*

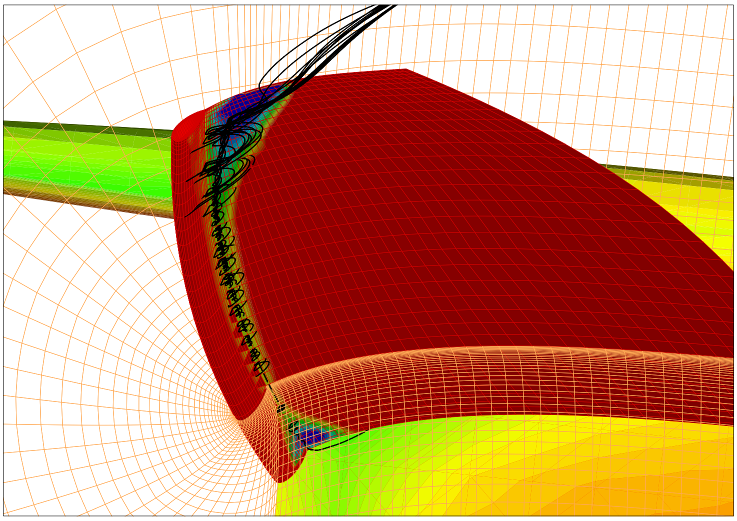

Figure 17: *Evolution of separations on a traditional A-Class rig.*