

## MULTI-OBJECTIVE OPTIMIZATION WITH MODEFRONTIER INTERFACES FOR ANSA AND METAPOST

<sup>1</sup>Alberto Clarich\*, <sup>2</sup>Rosario Russo, <sup>3</sup>Marco Carriglio

<sup>1,2,3</sup>ESTECO srl, Trieste, Italy

### KEYWORDS –

Multi-objective Optimization, Multi-Variate Analysis, Game Theory, Grid Computing

### ABSTRACT –

This paper is focused on the integration of multi-objective design environment code modeFRONTIER with ANSA mesh morphing and  $\mu$ etaPOST CAE visualization software.

In this environment, the users can easily define a workflow linking together different modules that represent the 'bricks' of the optimisation problem: design variables, model and mesh files, CAE nodes, specified outputs, as well as optimisation objectives and constraints.

In particular, several CAE tools can be easily integrated in the process through the available direct interface nodes, including ANSA and  $\mu$ etaPOST, allowing the automatic update of morphing parameters and extraction of specified results (related to optimization objectives).

This paper will also illustrate, through some applications of ANSA combined with FEM and CFD solvers, how the Grid capability of modeFRONTIER allows to distribute the simulations on a network of available machines, fully exploiting the available computing resources.

Several multi-objective optimization algorithms are available in modeFRONTIER, and in particular, for the applications of this paper it will be illustrated the efficiency of Game Theory and of fast algorithm FMOGAll (based on Genetic Algorithm and Response Surfaces), to obtain the optimal results with the lowest number of design simulations.

In addition, dedicated tools, including Statistical Analysis and Multi-Variate Analysis, can be used in order to help the users analyse the influence of the variables on the objectives, improve the efficiency of the optimisation strategy and select the definitive solution.

### TECHNICAL PAPER –

## 1. INTRODUCTION

Multi-objective optimisation in automatic and distributed environment, that allows direct communication between multi-disciplinary simulation software, is becoming continuously a key factor in design process.

Traditional design approach ('trial and error') usually requires many attempts to the designers, which every time need to modify their numerical models by hand and run several solvers, especially when it is difficult to know a priori in which direction of the multi-dimensional variables space to move in order to find the best solutions.

Conversely, the multi-objective design environment modeFRONTIER [1] allows to integrate different computational software (any commercial or in-house code) into a common design environment, thus allowing the automatic run of a series of computations proposed by a selected optimization algorithm, until the specified objectives are satisfied.

In this modular environment, each component of the optimisation process, including input variables, input files, scripts or direct interfaces to run any software, output files, output variables and objectives, is defined as a node to be connected to the other components.

In this scenario, the ANSA and  $\mu$ etaPOST direct interfaces available in modeFRONTIER play an important role for designers which deal with parametric mesh morphing through ANSA and extraction of results from any CAE model by  $\mu$ etaPOST, since all the morphing parameters and responses defined in the model are automatically recognised by the interfaces, and therefore the model can be easily integrated in the optimisation loop.

In this way, the complete logic flow from parameterisation to performance evaluation is defined by the user, that can select among several available optimisation algorithms, accordingly to the objectives defined, including Genetic Algorithms [2], Evolutionary

Algorithms, Game Strategies [3], Gradient-based Methodologies, Response Surfaces (to speed up the convergence of optimisation, approximating the response of the system by the use of the available Meta-Models), and Robust Design Optimisation (optimisation under uncertainties on input parameters).

Statistical and Multi-Variate Analysis tools [10], some of them being illustrated in this paper, can be applied to find important information about the influence of the parameters in the system, in order to face up problems characterised by a large number of parameters and large computational efforts with the highest efficiency.

Two industrial application examples are presented in this paper, one related to a structural optimisation (the harmonic analysis optimization of an engine crankshaft), and one related to a CFD optimisation problem (yacht sail flying shape design).

## 2. FIRST APPLICATION: CFD OPTIMIZATION OF YACHT SAIL FLYING SHAPE

### Problem description and ANSA model definition

The first application case focuses on the design of a flying shape for a yacht sail. In sail design terminology, the design shape is the undeformed shape of the sail, whereas the flying shape is the deformed one. In this regard, a common task of sailing yacht designers is to optimise flying shape performance using CFD simulation, and finally find the corresponding design shape through reverse-engineering process [6].

The primary objective in designing the flying shape is to maximize the driving force; in addition, in order to provide sufficient transversal stability, another objective needs to be introduced, which is to minimize the (transversal) heeling moment.

In our application, the baseline geometry of the deformed flying shape is obtained from an existing sail model of an high-performance 65-feet racing yacht. The mesh is made up by 2,5 million polyhedral elements, mainly composed of tetrahedral cells for the volume surrounding the main and Gennaker sail, while variable size hexas in the main volume are connected through pyramids and tetras with the surface mesh over the sails.

Morphing boxes are created in ANSA v13.1.1 and modelled in order to contain the whole Gennaker geometry inside. In this regard, four split layers are inserted in order to parameterise the Gennaker sections at 0%, 25%, 50% and 75% of total height (see Figure 1, left).

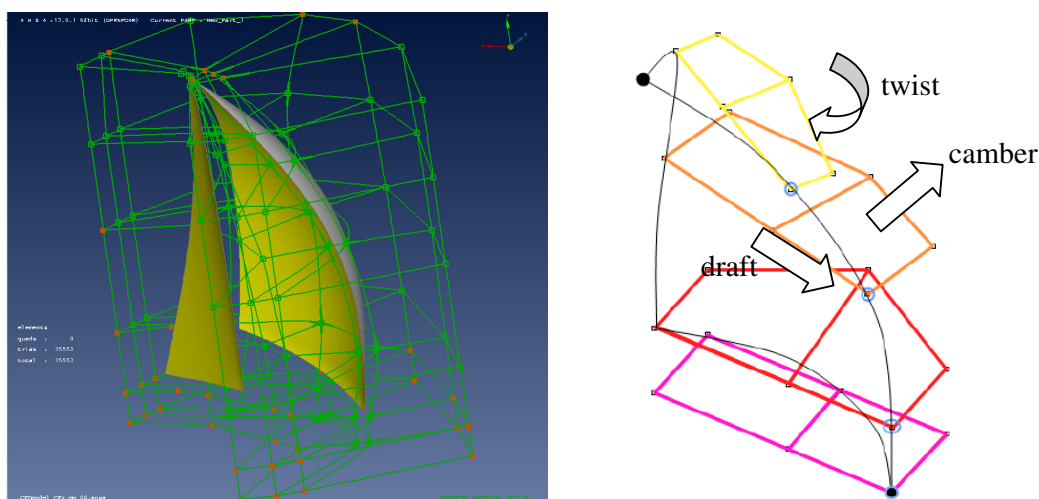


Figure 1 - Sail flying shape morphing box in ANSA (left) and morphing parameters (right)

Accordingly, draft, camber and twist angle at each section (fig.1 right) are defined as morphing parameters in ANSA model to generate new mesh shapes as variation of the parent mesh.

ANSYS CFX solver is used for CFD simulation. Boat speed is considered equal to 6.23m/s and wind speed equal to 6.23m/s, with a wind angle of  $-139^\circ$ ; Reynolds is equal to  $3E6$ .

## Workflow creation in modeFRONTIER and usage of ANSA direct interface

The next step needed to setup the optimization process is to define the process workflow in modeFRONTIER (fig.2), which looks as a modular network connecting the different “bricks” of the process design.

All the input parameters (illustrated in previous section) are defined by dedicated nodes, which specify their range of variation, and they are all linked to the first application node, ANSA direct interface. The execution of this node automatically updates the values of the design variable for each different configuration proposed by the optimization algorithm, and the updated mesh is linked by a transfer node to the following application, in this case the node to execute the CFX analysis. The latter one receives in input the session files that records all the operations needed to setup the CFD model, perform the simulation, extract the results, and gives as output a text file node (*Output\_forces*), from which for each design the two forces needed for the evaluation of the objectives are extracted by a relative parsing rule.

The two objectives are: 1. Maximize driving force (absolute value of Fx) and 2. Minimize heeling moment (Fz). The workflow also includes a constraint for the maximum surface area allowed.

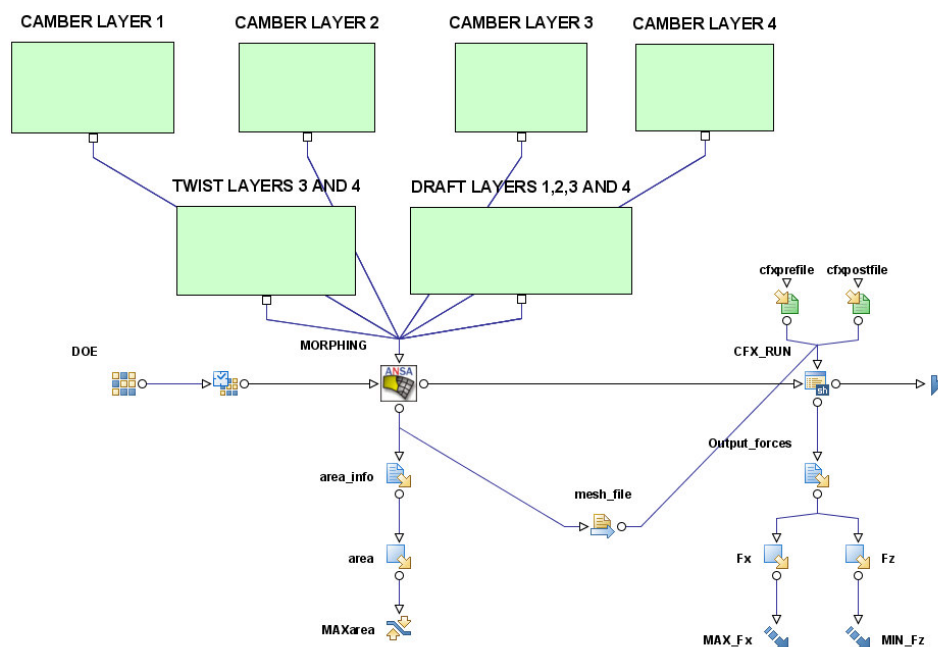


Figure 2 - modeFRONTIER workflow for sail optimization

About ANSA direct interface (fig.3), the user has just to indicate the ANSA model file name, the ANSA Design Variable File as defined in the Optimization task (DV file), and the name of the output file to be saved by the ANSA model (in our case, export for ANSYS CFX). At that point, the morphing variables already existing in the ANSA model are automatically introspected by the node, and it is possible to link each one of them to a design variable defined in modeFRONTIER workflow.

In this way, the integration of ANSA model in modeFRONTIER is completed: for each design proposed by the selected optimization strategy, the ANSA model will be automatically updated by modeFRONTIER, which will transfer the model output file to the next application (CFX simulation), that will be automatically executed and will give as result the needed output variables, repeating the loop until the optimization objectives are satisfied.

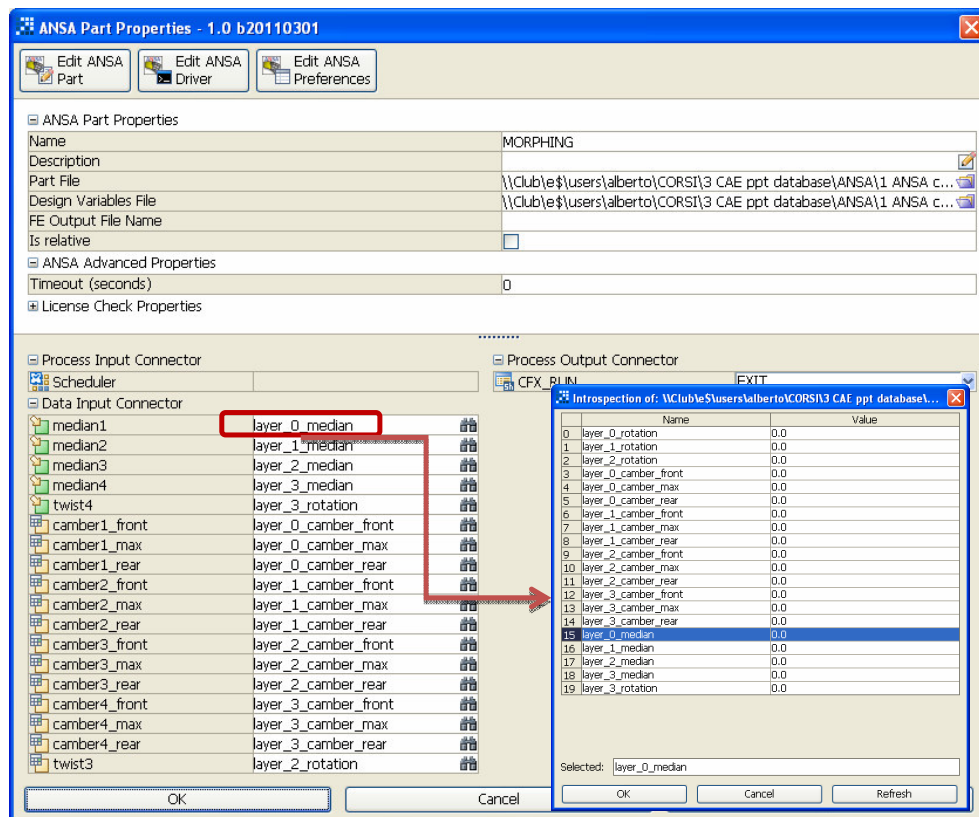


Figure 3 - modeFRONTIER direct interface for ANSA

Usage of modeFRONTIER GRID for distributed remote applications

From the Run options of modeFRONTIER, it is possible to activate the Grid tool [7].

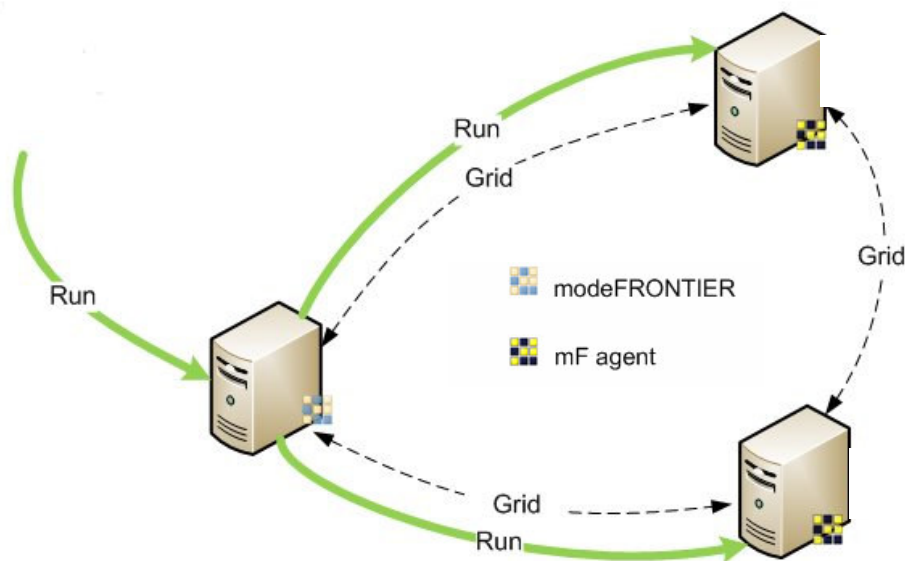


Figure 4 - Grid network launched by modeFRONTIER

If modeFRONTIER is launched on a server machine, it is possible to execute some application nodes (including the SH node that launches CFX simulation) on a set of specified remote machines available in the network (fig.4).

Jobs are therefore assigned by the system to different grid nodes according to the load balancing policy, excluding nodes that are disabled or can't execute jobs, and therefore fully exploiting the available computational resources of the local network.

As a requirement, it is needed to mount a common directory visible from all the nodes of the Grid (as a "Network drive" in Windows OS), and then the agent should be configured via Grid Manager on each machine that will enter in the Grid. As soon as the Grid System is activated on each machine, the optimization can be launched by modeFRONTIER.

For the application presented in this paper, four double-Cpu Linux machines had been used in the Grid, allowing to complete in 4 hours (the average time needed to complete one CFD simulation on each double-cpu machine) 4 simultaneous different CFD simulations relative to as many different model configurations proposed by the optimization scheduler.

At this point, we are ready to launch the optimization process, whose strategy will be described in the next two sections.

### Preliminary Design Exploration using DOE and results interpretation: Clustering Analysis

As a first step for the selected optimization strategy, a preliminary exploration of the design space is performed using Uniform Latin Hypercube [8] DOE (Design Of Experiments). Using this method, 100 points are created and distributed in the design space according to a uniform distribution applied to each of the input variables defining the optimization problem. The Uniform Latin Hypercube algorithm allows a regular equally-spaced sampling throughout the design space to attain the most of information out of the points analyzed with the minimum number of design evaluations.

The performance of this DOE algorithm can be measured by the Pearson correlation coefficient value calculated for each pair of design variables. In fact, correlation coefficient, which ranges from -1 (inverse linear relationship) to +1 (direct linear relationship), represents a measure of linear dependency between a couple of variables [9]. Correlation index almost approaches zero for each couple of variables of our problem (as resulting from the Correlation matrix available in modeFRONTIER Design Space), thus we can conclude that the input variables are uniformly distributed throughout the design space, resulting in a good quality sampling of the design space.

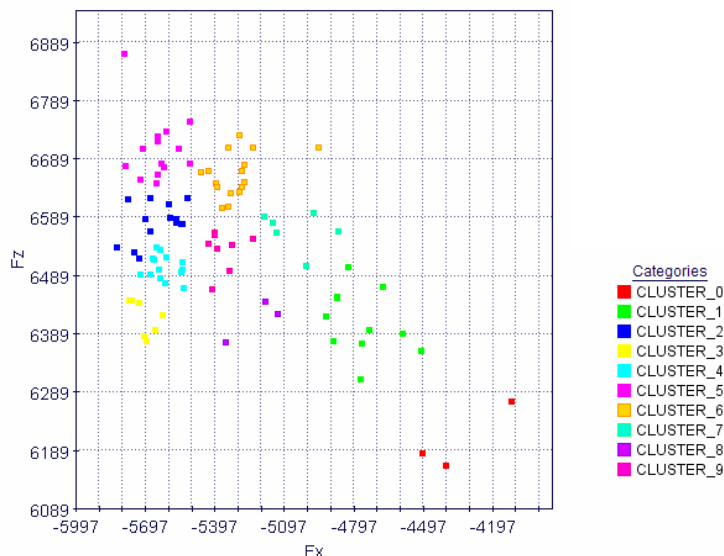


Figure 5 - Scatter chart of the design objectives: clusters are highlighted in the design space

The results of the DOE configurations for the sail model were analysed using a Hierarchical Clustering analysis [10] which allows to identify different groups of design solutions, by considering each feature related to the shape and performance of the flying shape sail.

There are many clustering methods available in modeFRONTIER, and Hierarchical Clustering analysis with Ward approach was used in this study: basically, the data are

divided in  $K$  partitions in such a way that the Euclidean distance between each design and his respective cluster centroid is less than the distances to any other centroid.

Figure 5 shows the effect of the clustering analysis applied to the dataset of design solutions evaluated using the Latin Hypercube Sampling DOE (Fx represent the driving force, whose absolute value is to be maximised, and Fz is the force component responsible of the heeling moment, to be minimised): 10 clusters of different solutions are found among the 100 design configurations available. In other words, 10 different groups of candidate solutions with similar characteristics in terms of shape parameters and output performance are found.

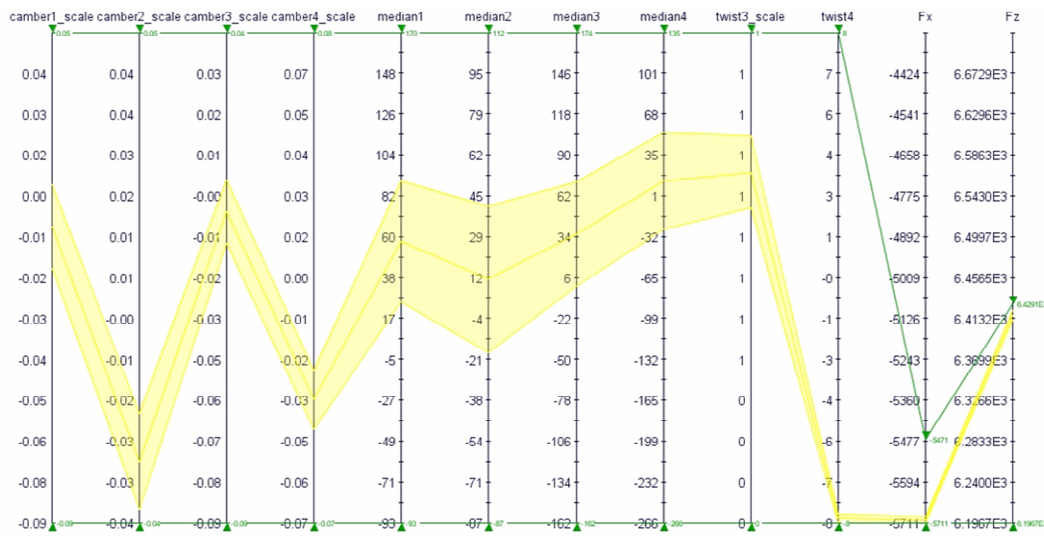


Figure 6 - Parallel coordinates chart showing the optimal cluster

Consequently, using a graphical tool like the Parallel coordinates chart to display in each column the numerical values of input and output variables for each cluster considered, it is possible to find out and screen the optimal candidate solution. In this regard, Figure 6 shows the range of input and output values corresponding to the optimal solutions which belong to the cluster labelled as 'CLUSTER\_3' and depicted in yellow in the chart (the width of the band is proportional to cluster variance). It can be easily pointed out that in order to minimize Fx and Fz (filtered in the chart) the twist angle for sections at 75% of total height (*twist4* variable) must be reduced significantly, and it seems to be the variable with highest influence. Similar considerations can be made for the other design variables considered.

The range bounds of all the input variables can therefore be reduced from the original values up to the extremes of the selected optimal cluster, and the optimization algorithm to be selected in the next step can therefore be applied in the reduced optimal variable space, improving the efficiency and the rapidity of the optimization process.

#### Optimization starting from the optimal cluster: FMOGAll algorithm

After the reduction of the variable space in the region defined by the optimal cluster as described in previous section, a multi-objective optimization algorithm is applied.

In modeFRONTIER several optimization algorithms are available, including Genetic Algorithms, Evolutionary Strategy, Game Theory, Downhill Simplex, Simulated Annealing, Particle Swarm, Gradient Based algorithms and FMOGAll.

The latter one is based on the integration of robust algorithms like MOGAll (Multi-Objective Genetic Algorithm) and the efficiency of the Response Surface Methodologies (fig.7).

Starting from a set of design data (DOE), in our case the designs belonging to the optimal cluster, different Response Surface Meta-Models (Radial Basis Function, Kriging, Neural Network, SVD, etc..) can be trained, and they can be used to automatically extrapolate the response of the system in function of the design variables. In this way, a full virtual optimization step can be performed (virtual in the sense that CFD and other application

analysis are not evocated, but results are extrapolated directly from the RSM mathematical Meta-Models), including a local refinement DOE phase around the best solutions, while the best solutions so far obtained (Pareto frontier) can be validated using real simulations (the chain of the workflow), updating this way the Response Surface Meta-Models with the new real designs evaluated in following steps, until a convergence is reached.

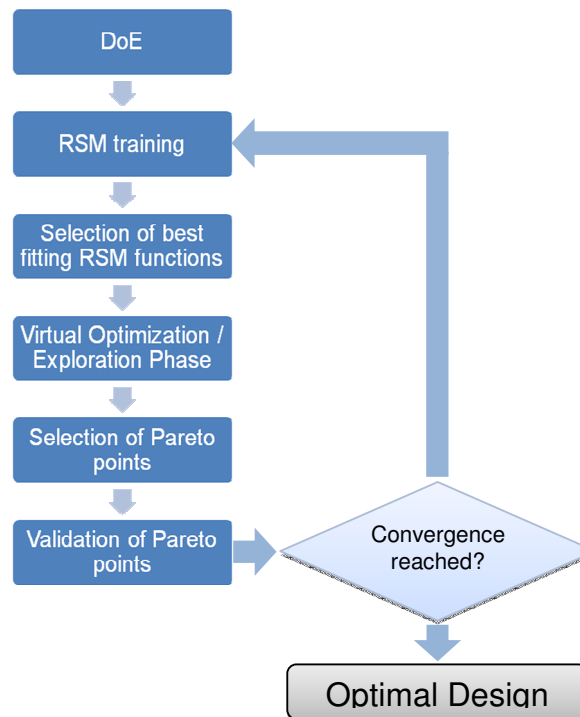


Figure 7 - Flow-chart of FMOGAI algorithm

Applying this methodology, it was possible to compute only further 50 Real design simulations, improving the design solutions as represented in fig.8. The new computed points (by FMOGAI) are represented in black in the performance space. The concentration of new black points in optimal zone of performance space further confirms the efficiency of combination of the filtering performed in first step, and of Fast algorithm employed in the second step.

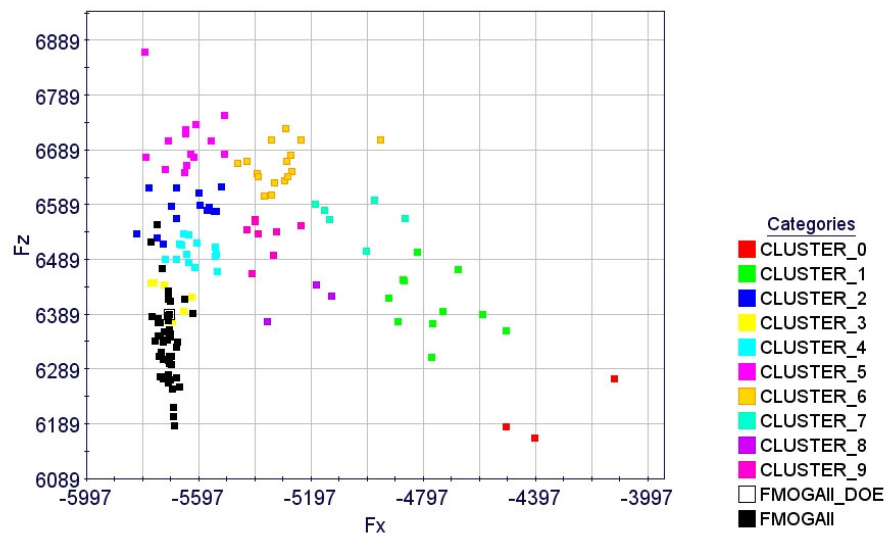


Figure 8 - FMOGAI results obtained in the optimal cluster

### Analysis of the results obtained: SOM Analysis

To better illustrate the influence of input variables on the design objectives for the optimal solutions found, and help the designer in the choice of the definitive design, a SOM (Self Organizing Map) analysis [11] is performed.

SOM is a non-linear projection of Multivariate data (large number of variables) on a two-dimensional regular grid, that allows to mine multiple local correlations and hidden structures in complex datasets. Each cell unit of the SOM can display by a colour scale the values of each variable (in a different SOM map), corresponding to the same local region or cluster of the variable space, thus enabling the user to find out local correlation values between input and output variables. In other words, SOM tool enables the designer to see in a multi-dimensional chart how to modify values of input variables to reach the different target values, here for both Fx and Fz forces.

A SOM is created on the samples belonging to the optimal cluster including the latest evaluated 50 points obtained from the second phase optimization. In this way, two groups of solutions having minimum value (blue scale) of Fx and Fz respectively are identified, as showed in Figure 9. Two groups of SOM units are in fact indicated on the top and bottom right side of the Fx and Fz maps respectively, identifying two precise areas of the design space and therefore two precise groups or clusters of data (two sail solutions belonging to the two different groups are also represented in fig.9). It can be pointed out that minimising Fx and Fz are conflicting objectives, since the two optimal regions are differently located, thus a trade-off solution must be found by the sail designer according to the sail specific requirements.

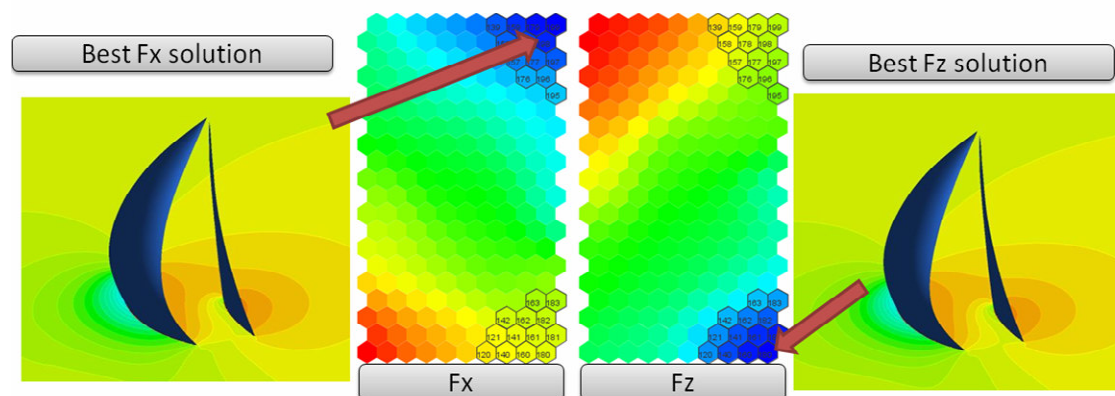


Figure 9 - SOM component maps for objectives illustrating trade-off

Therefore, if we visualize the SOM units corresponding to the best values of the two design objectives on the maps relative to the input variables, we may determine and compare the values of the input variables which can minimise Fx and Fz respectively.

Using the SOM tool, the designer can in fact visualize and estimate the variation of the input variables when moving from the optimal solution for driving force Fx (top right side of the maps) to the best solution for heeling moment Fz (bottom right side of the maps).

In this regard, Figure 10 shows how each input variable changes when moving from best solution for Fx to best solution for Fz, both using SOM maps (on top) and Cluster Parallel Coordinates Chart (on bottom) representation.

We may find for instance that the input variables having more effect on the two design objectives (having highest variation when moving from best Fx to best Fz solution) are the 4 camber parameters, in particular the ones relative to section at 25% and 75% of total height. Consequently, high camber values optimize the driving force, whereas low camber values minimise the heeling moment.



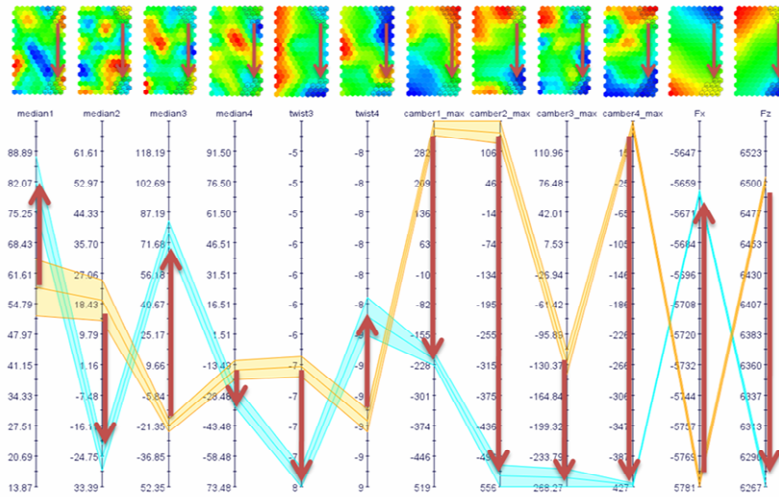


Figure 10 - SOM chart: variation of input variables from best solution for Fx to best for Fz

In both the optimal solutions proposed of fig.9, the performance of the baseline sail model are improved for both Fx and Fz as showed in Figure 11, which reports a comparison between the different models considered.

In a similar fashion, each sail designer can arbitrarily select the final flying shape of the sail according to his preferences and requirements.

	Fx [kN]	Fz [kN]
<b>Baseline</b>	-5.47	6.62
<b>Best Driving Force</b>	-5.76 <b>(+5.3%)</b>	6.39 <b>(-3.5%)</b>
<b>Best Heeling Moment</b>	-5.68 <b>(+3.8%)</b>	6.18 <b>(-6.6%)</b>

Figure 11 - Performance results summary: baseline and optimal solutions

### 3. SECOND APPLICATION: HARMONIC OPTIMIZATION OF CRANKSHAFT

#### Problem description

In this second application, the purpose is to optimize the web shape of a 4-stroke, 4 cylinders engine crankshaft, whose main data [12] are: engine power=50KW@2000rpm, crank radius=32mm, rod length=150mm and piston diameter=85mm. The forces scheme acting on the rod (for design crank speed of 2000rpm) can be visualised in fig.12, from which it has been decided to skip the effects of radial (normal) force.

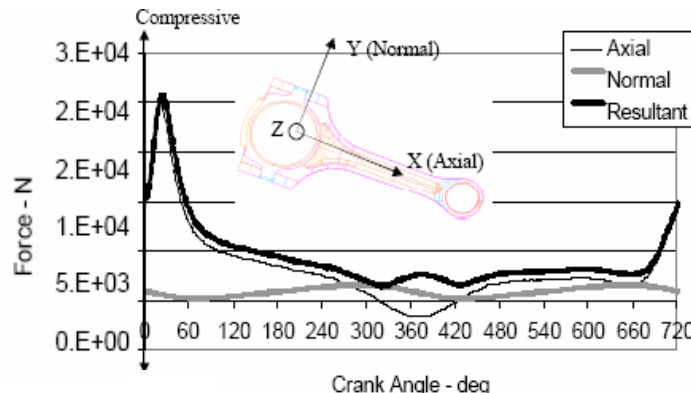


Figure 12 - Forcing loads on the connecting rod of the crankshaft

In particular, the web shape of the crankshaft is to be parameterized (see next section), in order to minimize the maximum stress deriving from an harmonic analysis [13], minimizing at the same time the mass of the crankshaft, and keeping a constraint on the rotation balance (i.e. on the reaction forces computed from static analysis under centrifugal load).

First of all, a modal analysis of the unconstrained crankshaft is performed, in order to find the first natural vibration modes: the analysis is made in Abaqus, with the model defined without constraints.

Second, the forcing loads applied with different phases on each crank-pin (derived from fig.12 and represented schematically for each pin in fig.13 left) have to be transformed by FFT (Fast Fourier Transform) in amplitude vs frequencies domain. In fact, each signal in the time domain can be decomposed in a sum of infinite harmonic components, each one of them characterized by a frequency, an amplitude and a phase (fig.13 right reports amplitude vs frequency transform of the load signal).

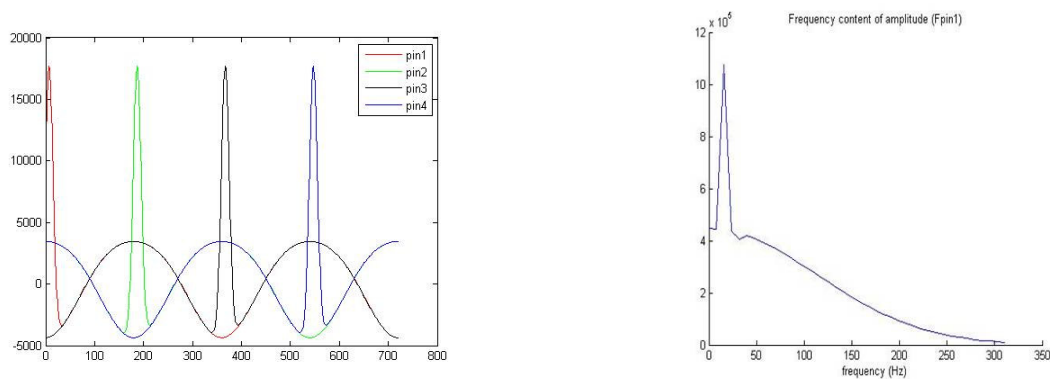


Figure 13 – FFT transform (right) of pin forcing loads (left)

Also the phase of the forcing loads can be obtained by the FFT, but in this case, since the forces on the 4 pins are equally phased, the phases of the harmonic components just have differences of  $\pi/2$  following the ignition order of the pistons (fig.14 left, vectors indicate the different phase of the harmonic components on the 4 pins).

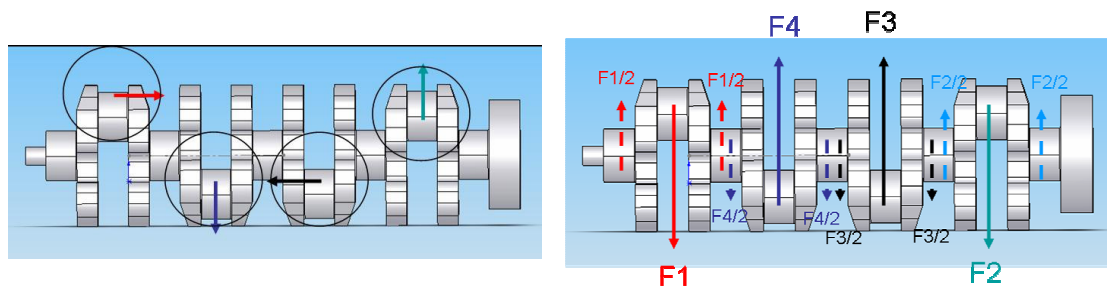


Figure 14 - Phase of the load harmonic components on the 4 pins (left) and reaction load on bearings (right)

We still need to define the harmonic components of the reaction loads on the bearings. As approximation, we can consider that each pin load is balanced equally by the two closest bearings, therefore each bearing is loaded by the sum of two half pin loads, except of the first and last that present only one component (fig.14 right).

In analogy, also the harmonic components of the reaction bearing loads are obtained by the sum of the relative harmonic components of pin forcing loads, and therefore all the loads on the crankshaft can be now expressed in terms of harmonic components, including phase and amplitude for each harmonic frequency.

Since only the harmonic components corresponding to the natural frequencies of a system can produce a large amplification effect of deformations/stresses (resonance), the harmonic analysis will be performed on the crankshaft applying only the harmonic components corresponding to the first natural frequency.

All the forcing loads and reactions are therefore applied on the pins and bearings specifying amplitude and phase corresponding to the selected frequency, and as result the stress field may be obtained using Abaqus steady state dynamic solver. An empirical structural damping coefficient equal to 3 is considered.

The harmonic analysis can be repeated also for the second natural frequency, but since the stresses found are less than  $1/10^{\text{th}}$  of the first frequency ones, their effect may be neglected.

### Model morphing and workflow definition using $\mu$ etaPOST

The mesh of the crankshaft model is completely defined in ANSA (over 800,000 mixed elements), and morphing boxes are created in order to control the shape of the crankshaft webs. For this purpose, six blocks are created around one half of the web (fig.15 left) and are fit to the geometry. The control points are defined as reported in fig.15 left (5 parameters) and in figure 15 right (angle parameter), allowing to modify width, height and angle of the different geometrical parts composing the web.

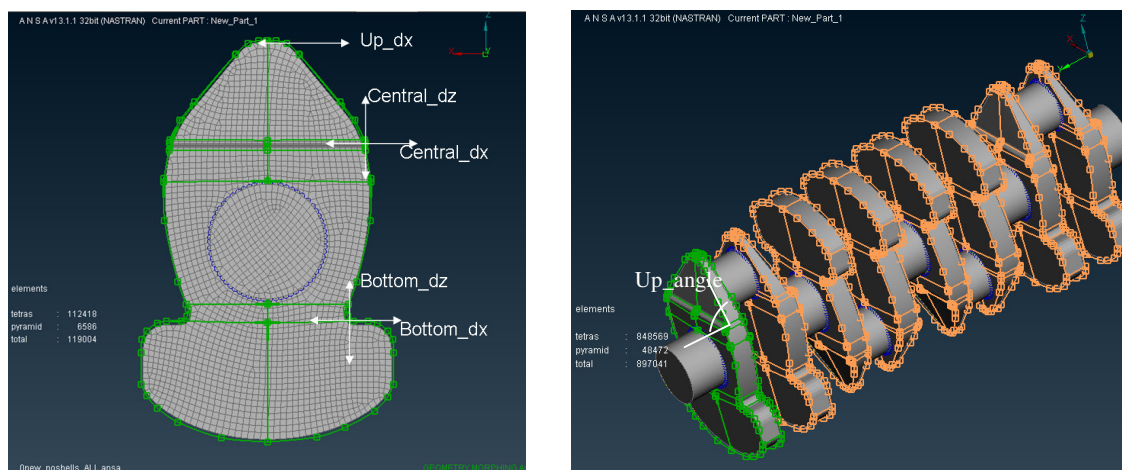


Figure 15 - Morphing boxes around web with relative control points (left) and links around crankshaft (right)

Note that the nodes around the base of pins and cranks are defined as nested elements with none DOF (blue in the picture), and links are defined to extend by symmetry the morphing boxes to the remaining parts of the crankshaft (fig.15 right). All the boundary conditions for Abaqus simulations are defined in ANSA model as described in previous section.

At this point, the ANSA model can be integrated in modeFRONTIER workflow (fig.16) using the direct ANSA interface as described in chapter 2. Besides the input variables defined above, also material or other model properties can be parameterized: in our case, Young modulus is considered.

The output files exported from ANSA node (.inp file) are linked this time to three different SH shell nodes, each one of them launching in batch mode a different Abaqus analysis. The first is a *modal analysis*, from which the first natural frequency obtained is used to extrapolate by the calculator node available in the workflow (*interp* function) the corresponding amplitude of the forcing loads from the relative file (fig.13 right). The second analysis is therefore an *harmonic analysis* performed on the same Abaqus .inp file, after having updated it with the forcing amplitudes obtained by the procedure just described. Finally, a *static analysis* is performed on the constrained crankshaft under centrifugal force, as explained in previous section, to evaluate the balance of the crankshaft.

In fig.16 it is illustrated then how the output data relative to harmonic and static analysis (respectively maximum stress amplitude and maximum reaction force) are extracted from the .odb output files produced by Abaqus, through the  $\mu$ etaPOST direct interface. From the interface it is just necessary to specify the name of the output file (in this case .odb) and the

name of the session file created in  $\mu$ etaPOST to define any response parameter. The latter can be extracted from the available list, as represented in fig.16.

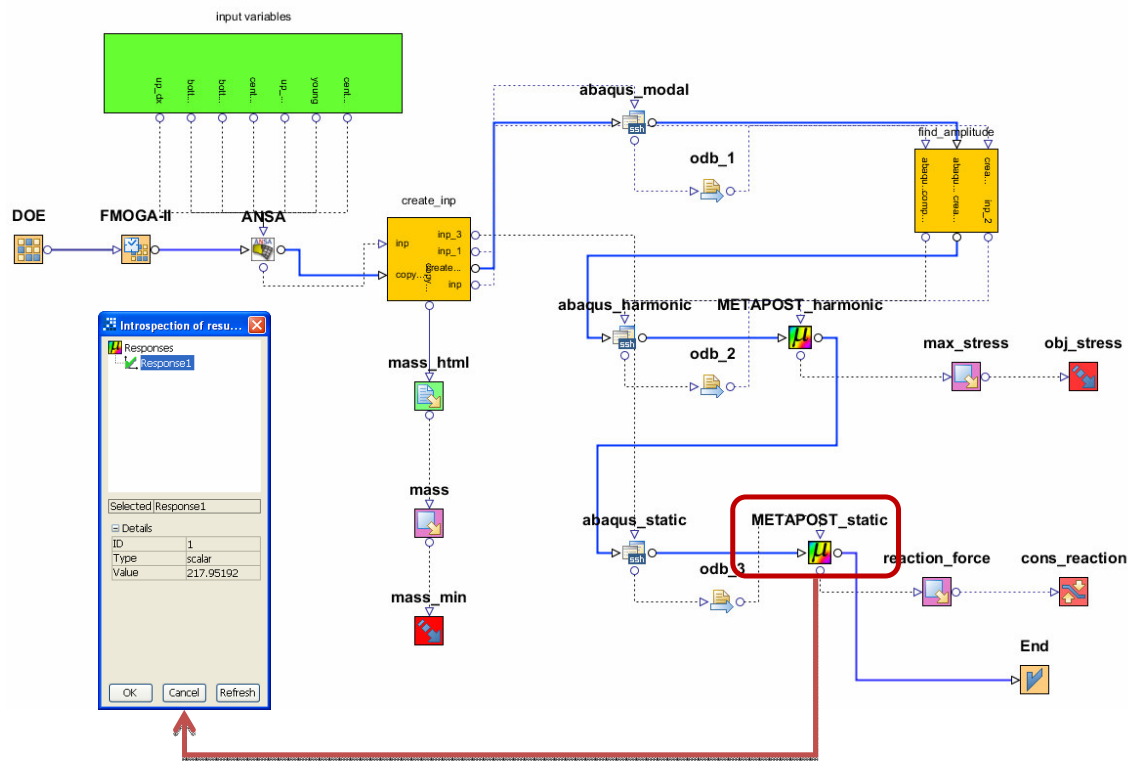


Figure 16 - modeFRONTIER workflow and  $\mu$ etaPOST interface

At that point, the workflow can be completed just by defining the objective nodes relative to mass minimization (extracted from ANSA deck command), to the harmonic analysis maximum stress minimization, and a constraint on the reaction forces on the pins, to be less than the original configuration values.

#### Optimization setup using MOGT (Game Theory) algorithm

The algorithm selected for this application is MOGT (Multi Objective Game Theory) [14]. This class of algorithms is particular efficient to find by a reduced number of simulations good compromise solutions between contrasting objectives (like our case), improving efficiently a baseline configuration.

The Competitive Game algorithm implemented in modeFRONTIER follows different steps: first, the variable space is initially decomposed randomly, and then each player, starting from a common original design, launches a mono-objective optimisation algorithm (Downhill Simplex [5]), to improve the objective assigned to it.

After a certain number of Simplex iterations, each player finds the best configuration (and set of variables) for its objective, and then the search continues with a new step, for which each player starts a new Simplex sharing the optimal variables found by the other players.

The problem of the variables space decomposition is very important [16], since it influences the results of the equilibrium point and thus the optimisation results. For this reason, an adaptive strategy has been applied in the MOGT algorithm, using statistical analysis and in particular the t-Student coefficient to decide, at the end of each player step, if a variable is statistically significant for the player to which it is assigned or not, and, in the latter case, i.e. if the significance percentage is lower than an assigned threshold, the variable is given to another player in the following step.

As a convergence, not only a (Nash) equilibrium point [15] may be found, but also the optimal variable decomposition between the players; in addition, several not-dominated

points can be found by the algorithm during its search, allowing to finally find a set of compromise solutions between the contrasting objectives (Pareto frontier), with a small global number of simulations required, if compared to any other heuristic algorithm like Genetic Algorithm or Evolutionary Strategies.

### Optimization results and Response Surfaces visualization

The application of MOGT to the problem defined in this paper allowed to find the results reported in fig.17, with an overall number of required simulation designs less than 100.

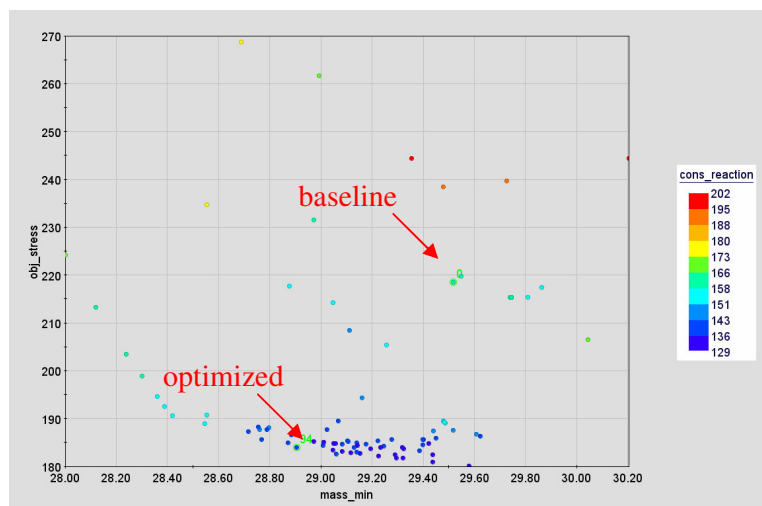


Figure 17 - Optimization results: mass and maximum stress magnitude to be minimized, reaction (colour) constrained (less than 164N)

In ordinates and abscissa of the chart in fig.17 the two objectives to be minimised are reported (respectively, maximum stress of harmonic analysis and crankshaft mass), and each design proposed during the optimization is represented by a point; the colour scale indicates the value of the constraint, which is equal to the static analysis reaction force, to be less than the baseline value of 164N.

Among the Pareto frontier (set of not dominated designs), one design (ID 94) has been arbitrary selected (see also fig.18). All the objectives have been significantly improved (16% reduction of maximum stress amplitude, 2% reduction of mass, and reaction forces on bearings reduced of 15%), confirming the efficiency of the algorithm applied. In particular, from the pictures taken from  $\mu$ etaPOST (fig.18) relatively to the harmonic analysis results, the reduction of maximum stress in the region of pin-web intersection is very clear.

In addition, in order to introduce another modeFRONTIER useful post-processing tool, fig.19 reports an example of RSM (Response Surface Methodology) 3D charts. All the designs data available from the optimization are used to train a Response Surface (several ones are available, see chapter 2: in this case, Radial Basis Function are used), which is a mathematical Meta-Model that can be used to extrapolate, for given input variable values, the output responses.

In fig.19, the response relative to maximum stress amplitude is represented in function of two selected variables, *bottom\_dx* and *central\_dz*, appearing in abscissa and ordinate of the charts, and it is clear how the stress raises significantly if *central\_dx* parameter becomes lower (in fact, optimal web configuration middle part is “shifted” towards the top); at the same time, a panel on the right is available, allowing the user to change manually the values of all the other input parameters, having an immediate update of the responses, and an immediate preview about the responses variation.

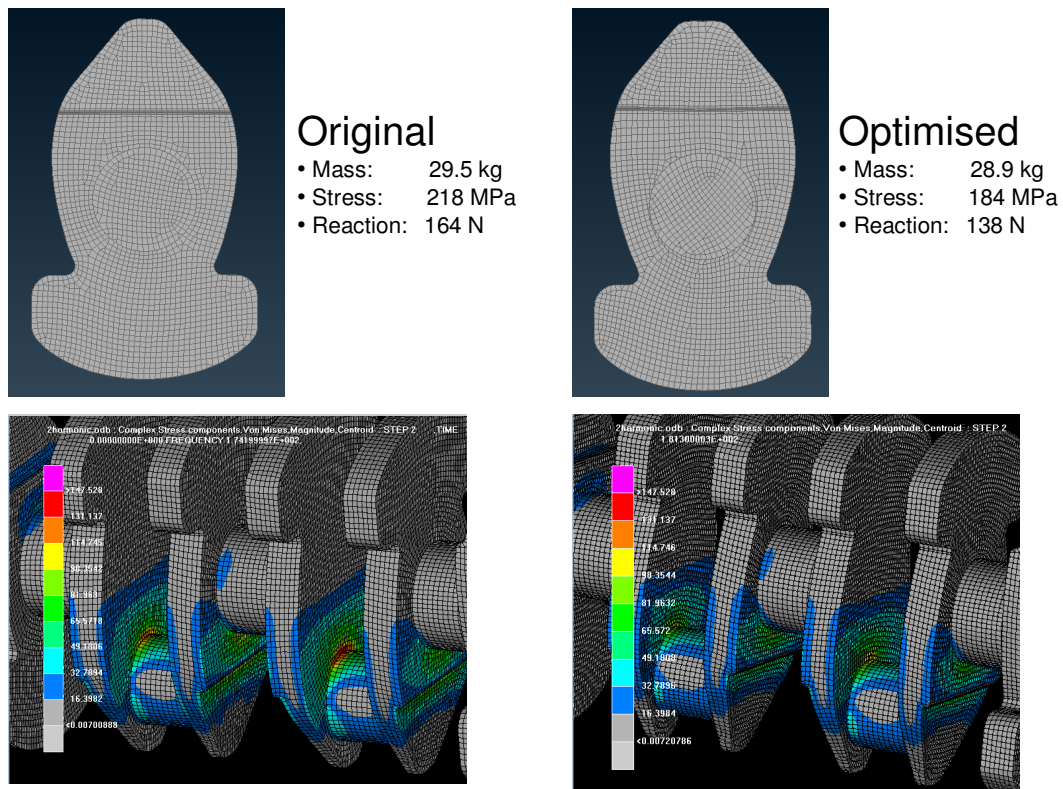


Figure 18 - Comparison between baseline (left) and optimised configuration (right): ANSA model and  $\mu$ etaPOST results

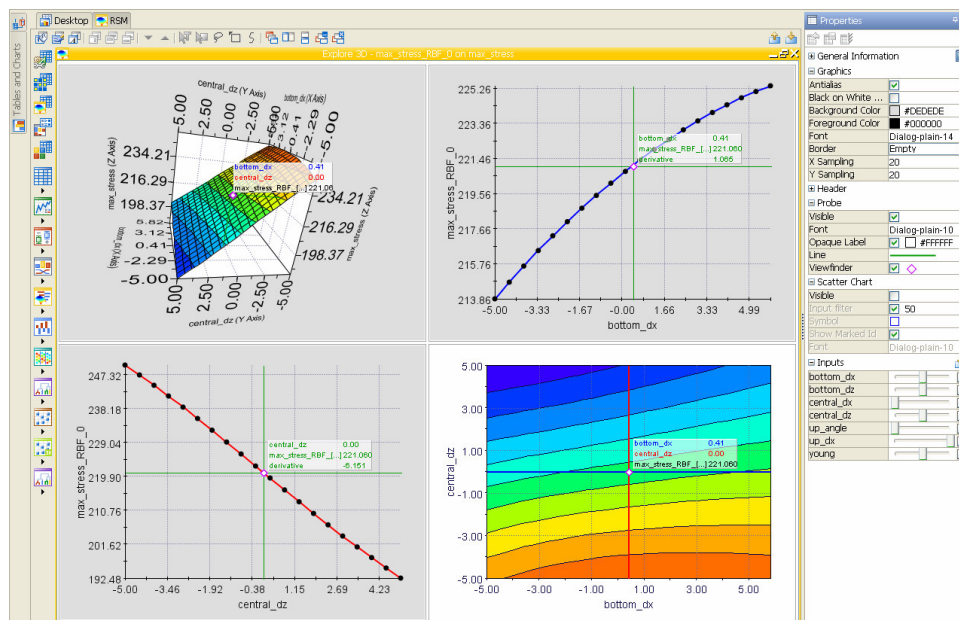


Figure 19 - Response Surface 3D chart in modeFRONTIER: maximum stress response vs morphing input parameters

Finally, fig.20 reports an example of another statistical tool, the Effect chart: for each output variable (here maximum stress) the effects of the single input variables are reported, including magnitude and sign: it is clear that the variable having most (inverse) effect is *central\_dz*, as confirmed by the RSM of fig.19, then *bottom\_dx* is the second variable by

importance with a direct effect (lower values give lower stress results), and then come all the other variables.

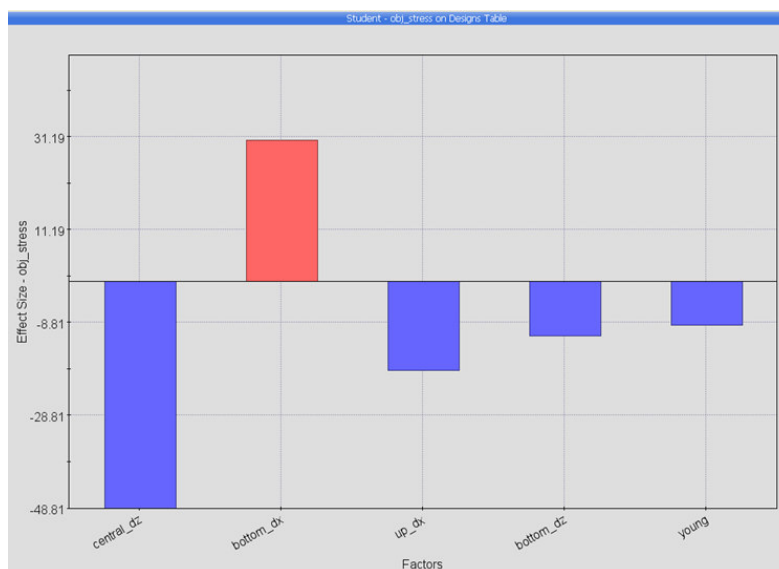


Figure 20 - Effect chart: stress objectives vs input parameters

#### 4. CONCLUSIONS

This paper shows how it is possible to integrate ANSA and  $\mu$ etaPOST software in the multi-objective optimization environment modeFRONTIER, through the dedicated direct interfaces, and how to set up and run a multi-objective optimization for any kind of application.

The efficiency of the procedure is proved by the results obtained in two industrial cases, a CFD application consisting in the maximization of driving force and minimization of heeling moment of a yachting sail, and a structural optimization of an engine crankshaft.

Two different kinds of strategies were adopted: in the first case, characterized by high computational-demanding simulations, a strategy based on statistical analysis (Clustering) and Fast Algorithm based on Response Surfaces (FMOGAI) was applied, in order to reduce the number of necessary simulations to reach the target. SOM (Self Organizing Maps) tool was therefore used to better visualize the multi-dimensional variable space, allowing to identify the variables from which the two objectives depend most, and help the designer in the selection of the definitive solution.

In the second case, a fast and efficient algorithm based on Game Theory (MOGT) was applied, and some useful post-processing tools have been illustrated to better analyze the results.

In both cases, the optimized solutions had significantly improved the baseline configurations, and have been obtained by a reduced overall number of simulation designs.

#### 5. ACKNOWLEDGEMENTS

The authors would like to thank George Korbetis, Serafim Chatzimoisiadis and all the BETA CAE Systems support team for the support in the creation of the ANSA models.

#### REFERENCES

- (1) <http://www.esteco.com>
- (2) K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans Evol Computational Journal, Vol.6, 2002

- (3) A. Clarich, V. Pediroda, C. Poloni, A competitive Game Approach for Multi-objective Robust Design Optimization, AIAA 2004-6511, Chicago, 2004
  - (4) O.L. Davies, The design and analysis of industrial experiments, Oliver & Boyd, 1963
  - (5) Nelder, J. A., Mead, R., A Simplex Method for Function Minimization, Computer Journal, 7, 308, 1965
  - (6) Rousselon N., 2008, Optimization for Sail Design, paper presented at the modeFRONTIER Conference, June 2008, Trieste, Italy
  - (7) I. Foster; C. Kesselman, S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, Intl. J. Supercomputer Applications, 2001
  - (8) Latin Hypercube Sampling: McKay M.D., Conover W. J. and Beckman, R. J. (1979), A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics 21(2), pp. 239-245
  - (9) Edwards, A. L., The Correlation Coefficient, Ch. 4 in An Introduction to Linear Regression and Correlation. San Francisco, CA, W. H. Freeman, pp. 33-46, 1976
  - (10) Kaufman, L. and Rousseeuw, P. J. Finding Groups in Data: An Introduction to Cluster Analysis. New York, Wiley, 1990
  - (11) T. Kohonen Self-Organizing Maps, 3rd ed. Springer, 2001
  - (12) S. Pravardhan. Dynamic Load Analysis and Optimization of Connecting Rod. Msc thesis, University of Toledo, May 2004
  - (13) H. Muller-Slany, F. Brunzel, J.A. Pereira. Structural Damage Detection Based On Highly Accurate Updated Models. In Proceedings IMAC-XIX, Kissimme, Florida, USA, February 5-8 2001
  - (14) A. Clarich, C. Poloni, Multi-objective optimisation in modeFRONTIER for aeronautic applications, including CAD parameterisation, Evolutionary Algorithms, Game Theory, Metamodels and Robust Design, EUROGEN 2007, Jyvaskyla, 2007
  - (15) J.F. Nash, Non-cooperative Games, Annals of Mathematics, (1951)
  - (16) J.W., Wang, J. Periaux, Search Space Decomposition of Nash/Stackelberg Games using Genetic Algorithms for Multi-Point Design Optimisation in Aerodynamics from Domain Decomposition Methods in Science and Engineering, pp. 139-149, CIMNE,2002
-