

## SHIP DESIGN OPTIMIZATION FRAMEWORK

### George Gabriel Cotoc

“Dunarea de Jos” University of Galati,  
Faculty of Naval Architecture, Galati, Domneasca  
Street, No. 47, 800008, Romania,  
E-mail: [george.cotoc@ugal.ro](mailto:george.cotoc@ugal.ro)

### Florin Pacuraru

“Dunarea de Jos” University of Galati,  
Faculty of Naval Architecture, Galati, Domneasca  
Street, No. 47, 800008, Romania,  
E-mail: [florin.pacuraru@ugal.ro](mailto:florin.pacuraru@ugal.ro)

### Liliana Rusu

“Dunarea de Jos” University of Galati,  
Faculty of Engineering, Galati, Domneasca Street,  
No. 47, 800008, Romania,  
E-mail: [liliana.rusu@ugal.ro](mailto:liliana.rusu@ugal.ro)

### Antti Pösö

NAPA Group,  
HTC Helsinki, Pinta Tammasaarenkatu 3  
FI-00180 Helsinki, Finland  
E-mail: [antti.poso@napa.fi](mailto:antti.poso@napa.fi)

## ABSTRACT

*This study intends to develop a Computational Fluid Dynamics space exploration framework, which creates a bridge between the NAPA software and OpenFOAM 9 simulation software. With the use of the Free Form Deformation function implemented in NAPA, iterations of the KCS hull were automatically generated by changing the length of the bulb. The newly created versions of the model were further analyzed with numerical investigations to determine the ship resistance simulation using RANS equations. The optimization process and the data transfer between the two software packages are monitored by the Dakota optimization software.*

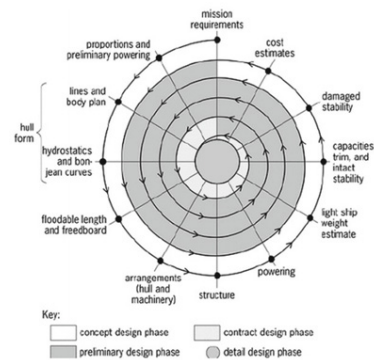
**Keywords:** CFD, RANSE, optimization, design of exploration.

## 1. INTRODUCTION

In the shipbuilding industry, a new ship order is subject to several design constraints. Of the constraints imposed by stakeholders, the most important are those related to cost and time. The cost and time invested in designing and building the vessel must be as low as possible, so that the offer is attractive to ship owners.

Architects and engineers accomplish the design stage by following the principle of design spiral [1] shown in Figure 1. This method is an iterative process that usually starts from basic preliminary information and orbits around a core that represents the real ship with all the approved details. To fulfill the requirements, the involved experts opti-

mize the ship’s performance by using knowledge gathered from older projects, through a process of trial and error.



**Fig.1 – Design spiral**

Besides their knowledge, they are using different software tools to increase productivity and lower the number of possible mistakes that might pass through. In the last decades, these software tools were individual, meaning that they were not syncing and if a change was made in the design spiral, all the subsequent processes must have been updated manually. This inconvenience is solved by developing a bridge between these designing software tools with the help of optimization tools and programming languages.

To make the ship more profitable, optimization can be applied to several areas. One possibility for an optimization study can be to increase the cargo holds volume, or to optimize the structural design. Also, optimizing the hydrodynamic performances would decrease the forward resistance of the ship and implicitly lower the needed energy used in the ship voyage.

One step in reaching these goals is presented in the current study. For the hull surface of the KRISO Container Ship (KCS) [2], the best bulb configuration to lower the forward resistance in calm water was found. The subject of the optimization is a 3,600 TEU container ship that has no full-scale built ship. The geometry modeled in NAPA can be observed in Figure 2 and the dimensions of the ship at scale 1:40 in Table 1. The CFD analyses done with the optimization framework were performed for Froude number 0.26.



Fig. 2 – KCS geometry

Table 1 – Main particulars

Main Particulars	Full scale	Model scale 1:40
LBP [m]	230	5.75
LWL [m]	232.5	5.81
Beam [m]	32.5	0.8125

Depth [m]	19	0.475
Draft [m]	10.8	0.27
Displacement [m <sup>3</sup> ]	52030	0.81
Block coefficient	0.6505	0.6505
Speed [m/s]	12.2456	1.95
Froude No.	0.26	0.26

## 2. NAPA and FFD

The software used to accomplish the hull surface generation, modifications and overall control of the processes is NAPA. For the optimization framework, which is the aim of this study, the Free Form Deformation function present in NAPA is one of the three methods used for the proof of concept. In developing the FFD method in NAPA, the theories from T. Sederberg et al in 1986 [3] and further developed by Coquillart et al in 1990 [4] were applied.

The key advantage of FFD is the faster design of new hull variations which is a lot quicker than the conventional method of manually adjusting the curves. If the original hull quality has been satisfying, it is also possible to keep the original fairness, so no additional fairing is needed for the newly generated hull.

The FFD function built in NAPA applies a local transformation to a specified surface by defining a three-dimensional box around it. For the bounding box, the number and placements of the control points between the corners may also be specified separately in each direction, and intermediary control points can be defined. The change will only have an impact on the portion of the surface that is in the box. Afterwards, the position of the displaced control points is set. By moving the initial set of points to the location of the new point, the surface within the bounding box is deformed. Figure 3 depicts the two bounding boxes used in the FFD function for the metamorphosis of the ship bulb.

### 3. OpenFOAM

The Computational Fluid Dynamics analysis stage of the framework was accomplished with OpenFOAM. Using this software had the advantage that it can be fully controlled with commands inserted in a Linux terminal. Knowing all the steps needed to start the analysis, it was made possible later to create automation scripts that control the flow of input and output data.

In the developed optimization loop, for turbulence closure, the  $k-\omega$  SST turbulence model with wall function formulation is used. The free-surface capture strategy is based on the multi-phase flow approach with high-resolution interface schemes using the Volume of Fluid method.

For parallel computing, domain decomposition is a technique utilized by OpenFOAM. Mesh and field decomposition, parallel application execution, and post-processing of the decomposed case are all steps in the parallel computation process. The function that decomposes the mesh and fields has the goal to divide the domain into manageable pieces while ensuring an economic outcome. When the decomposing stage is done, every patch is assigned to the core processor that will do the calculations. At the end of the parallel calculations, the postprocessing stage must bring the patches back to their original shape. The entire simulation can be post-processed by rebuilding the case, or alternatively, a specific portion of the decomposed domain can be post-processed separately by simply treating each processor directory as a separate case.

### 4. DAKOTA OPTIMIZER

Comprehending and forecasting complicated behaviors in fluid dynamics is done by modeling a simulation environment. Simulators frequently act as virtual prototypes where a set of established system parameters are changed to enhance a system's performance as stated by one or more system per-

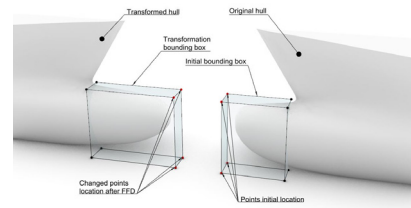


Fig. 3 – Deformation bounding boxes

formance objectives. Executing the simulator, assessing the performance objective(s), and modifying the system parameters in an iterative, automated, and guided manner, are necessary for such virtual prototype optimization or tuning.

Dakota is giving access to a wide variety of iterative capabilities through a single, straightforward interface between Dakota and the involved simulator. It often just takes a few commands to be changed in the Dakota text input file to try a different iterative approach or meta-algorithm, followed by the commencement of the new analysis.

Dakota and the simulation code typically have a loosely connected relationship, through the reading and writing of small data files, and the simulation code exchange information.

The main classes of algorithms provided by Dakota are *Parameter Studies*, *Design of Experiments*, *Uncertainty Quantification*, *Optimization* and *Calibration*. The method *Design of Experiments* (DoE) consists in exploring the parameter space of an engineering design problem, being suitable for this framework.

From the library of algorithms implemented in Dakota, the method *Design Analysis of Computer Experiments* (DACE) is the recent form of DoE. This method uses the sampling tool Latin hypercube sampling [5] and it was implemented in the framework that makes the subject of this study. The computer experiment was done to investigate the design space and to better understand the range of possible outputs for a given bulb length. Finding out which inputs have the

most impact on the output or how input changes affect the output is another goal.

## 5. FRAMEWORK OPERATIONS

The overall framework structure used to design the experimental space exploration loop is shown in Figure 4.

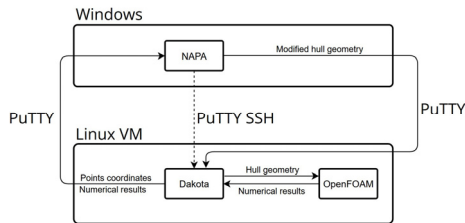


Fig. 4 – Overall experiment structure

OpenFOAM and Dakota are software products that require a Linux operating system. To have them both and NAPA on the same PC, an Oracle Virtual Machine was created having the Ubuntu operating system. Also, PuTTY was installed to transfer commands from NAPA to Virtual Machine. PuTTY has an implemented SSH network protocol that permits a protected connection to the Linux environment. Besides sending commands, is possible to transfer files between the two OS.

To function properly, the exploration loop requires a list of settings to be done before the start of the framework. Firstly, the geometry is stored in NAPA, having a bounding box generated to envelop the bulb. The maximum bounding box dimensions and the number of iterations are mentioned in the file *dakota\_case.in*, along with the sampling method (LHS) which is used by Dakota. The looping is programmed to make 20 iterations for the current study. In the end, the framework will have 20 generated ship bulb configurations with 20 sets of CFD. In this file are also found names for the files used to read the output value from the CFD solver, to write the next parameters used to modify the bulb length and the overall control file *simulator\_script*. The file *simulator\_script* has

three chapters, pre-processing, analysis and post-processing, which contain line codes that control and show the stages of the CFD simulation. The OpenFOAM parameters regarding the fluid speed, mesh discretization, time step and turbulence model are written in the analysis template which is later copied in every generated case. When all the data is set in place, the framework can be initiated.

NAPA offers the possibility to create custom macros. In a general macro, the functions to modify and generate the geometry are written. Also, this code controls the overall processes of the framework and gathers the results at the end. The macro begins by setting up the variable that contains the Ubuntu virtual machine credentials, as in the user, password and Ip. The latter variables are needed to send or copy certain files from the windows environment to Ubuntu VM and backward with PuTTY.

The optimization starts when NAPA is calling Dakota from the VM, and the first case is generated. The following step, which is also the framework's central component, oversees finding the file that provides the coordinates for each point and applying the FFD capabilities to the ship bulb. When the surface is successfully modified, the next step is to export it. The amount of time required to complete the export could range from a few seconds to a few minutes. The time duration depends on the values of the STL export parameters.

Sending the STL file to the Ubuntu VM's Dakota case folder, where the mesh should be placed, is the last action that the macro iteration must take after completing the export. With the STL file in place, the pre-processing stage will begin. The stage tasks consist in generating the mesh boundary and refinements, then subtracting the hull geometry from within using *snappyHex-Mesh*. Next, the job is split for every processor, followed by the actual calculation of them and in the end reassembling them. At the end of the analysis, the CFD results are

located and averaged. A separate script created in python offers the possibility to choose the percentage of the values for the average, and also, to select the type of values needed, such as viscous resistance, pressure resistance and others. The resulting values are sent to Dakota in order to return a new set of coordinates for the next case, and to NAPA.

**6. FRAMEWORK RESULTS**

At the end of all the analysis cases, the objective to study the CFD performance around the hulls with different bulb lengths using OpenFOAM was accomplished. Table 2 presents the total resistance obtained at the end of the analyses. Besides their value, a percentage comparison with hull no. 2 is made, which had the bulb length equal to the original one.

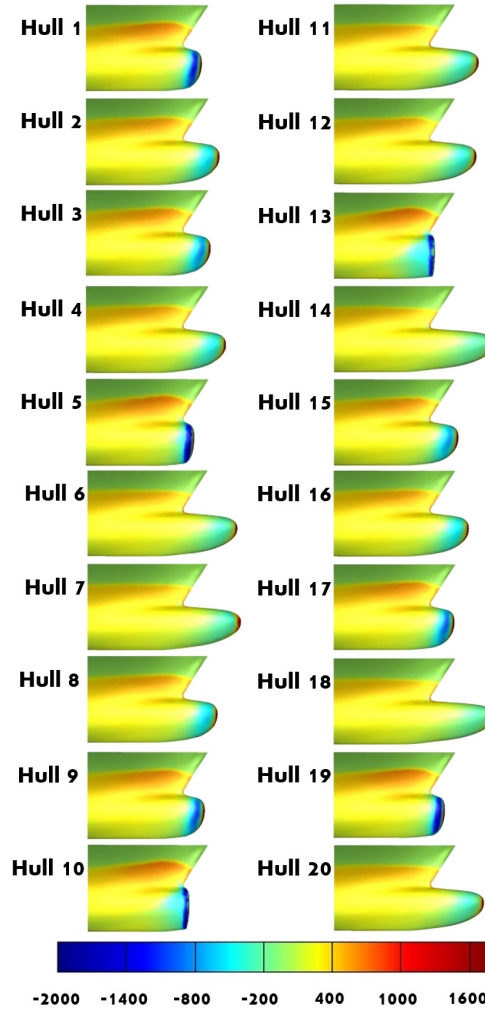
**Table 2 – Total resistance**

Hull	Bulb box X coord. [m]	R [N]	%
No.1	233.63	44.424	1.123 %
No. 2	237.67	43.931	-
No. 3	235.62	43.951	0.048 %
No. 4	239.20	43.740	-0.434 %
No. 5	231.89	47.105	7.225 %
No. 6	242.06	43.908	-0.052 %
No. 7	242.88	45.064	2.580 %
No. 8	237.49	43.680	-0.570 %
No. 9	234.45	44.149	0.497 %
No. 10	230.08	50.732	15.482 %
No. 11	240.72	43.812	-0.269 %
No. 12	240.20	43.404	-1.199 %
No. 13	230.53	51.654	17.581 %
No. 14	244.26	44.120	0.431 %
No. 15	236.01	43.801	-0.294 %
No. 16	238.30	43.745	-0.423 %
No. 17	234.90	44.039	0.247 %

No. 18	244.10	44.120	0.432 %
No. 19	232.71	45.000	2.434 %
No. 20	241.70	43.850	-0.184 %

Correlating Table 2 with the bulb profiles presented in Figure 5, the configurations that have a blunt bulb have an increase in the total resistance of up to 18 %.

Figure 5 represents the hydrodynamic pressure plotted on the hull versions. The pressure variation is drastically changing between the variations with a blunt bulb and with a longer bulb.



**Fig. 5 – Hydrodynamic pressure**

## 7. CONCLUDING REMARKS

This study was undertaken to design a framework that fulfils the need to optimize and explore design space in the early stages of ship design. With the use of NAPA software and the basic programming language built within, the generated macro can connect NAPA to CFD software and optimization software.

To validate the proof of concept for the framework, a study of 20 hulls with different bulb lengths was defined in NAPA. The study began with NAPA creating the loop by calling Dakota. Later, the optimization capabilities of Dakota were combined with the CFD analysis made by OpenFOAM. At the end of the analysis, 20 hull iterations and 20 sets of hydrodynamic information were gained. The numerical results were over-viewed, and the hydrodynamic pressure was plotted on the hull used in the study. In the presented results, the length variations have an influence over the ship's total resistance, created wave system and the pressure on the hull surface.

## REFERENCES

- [1] **Apostolos Papanikolaou**, A Holistic Approach to Ship Design, Springer 2019
- [2] **Van, S.H., Kim, W.J., Kim, D.H., Yim, G.T., Lee, C.J., and Eom, J.Y.**, 1997, "Measurement of Flows Around a 3600TEU Container Ship Model," Proceedings of the Annual Autumn Meeting, SNAK, Seoul, pp. 300-304 (in Korean)
- [3] **Sederberg, T. W. and Parry S. R.**, Free-Form Deformation of Solid Geometric Models, , Proc. SIGGRAPH '86
- [4] **Coquillart, S.**, Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling, , INRIA research document, 1990
- [5] **L. P. Swiler and G. D. Wyss**. A user's guide to Sandia's latin hypercube sampling software: LHS UNIX library and standalone version. Technical Report SAND04-2439, Sandia National Laboratories, Albuquerque, NM, July 20

*Paper received on November 21<sup>th</sup>, 2022*