University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture
Department of Energy, Power Engineering and Environment

Ship Hull Optimisation On A Desktop Computer

Robert KESER
Vuko VUKČEVIĆ
Vanja ŠKURIĆ
Inno GATIN
Hrvoje JASAK

ABSTRACT

Considering the trends in the modern shipbuilding industry, it is obvious that fuel consumption and exhaust emissions have become more important than ever, therefore ship efficiency improvements are mandatory. In order to tackle this problem, an automatic ship hull optimisation loop is proposed. To reduce the required computational power and time, a linearised free surface solver is used, which needs approx. 75 minutes to reach a converged solution on one Intel Core i5-4570 processor (with 4 cores). Changes to the hull geometry are done by an automatic mesh motion solver and the optimisation loop is controlled with the Dakota toolkit.

Optimisation loop

The proposed optimisation loop is given in Figure 1. The Dakota toolkit is used as a flexible interface between `foam-extend` and integrated optimisation methods.

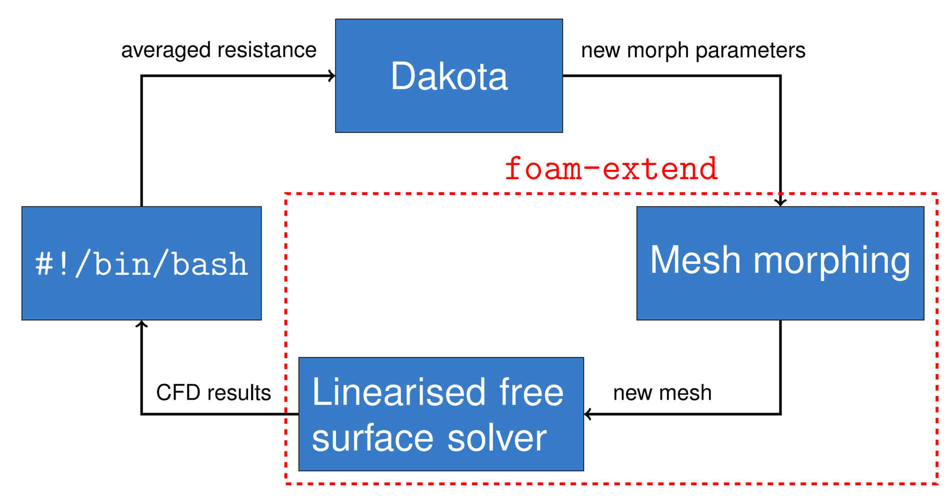

Figure 1: Optimisation loop

The selected optimisation method gives new morphing parameters to the developed automatic mesh motion solver which modifies the discretised ship's hull and gives it to the developed linearised free surface solver. After the calculation, a `bash` script averages the obtained resistance and returns it to the optimisation algorithm.

Mesh morphing

Changes to the hull geometry are done by an automatic mesh motion solver which uses a given transformation function to calculate the displacement for patch (ship's hull) points located inside the user-defined bounding box. In the presented case the transformation function is defined as follows:

$$\Delta y = \left[a_5x^5 + a_4x^4 + a_3x^3 - (a_5 + a_4 + a_3)\,x^2\right]\left[b_5z^5 + b_4z^4 + b_3z^3 - (b_5 + b_4 + b_3)\,z^2\right]$$

where Δy is the displacement in the y direction, x and z are local coordinates inside the bounding box, $a_{\{3,4,5\}}$ and $b_{\{3,4,5\}}$ are coefficients of the given polynomial which are also optimisation parameters. Figure 2 shows the calculated displacement for $a_{\{3,4,5\}} = b_{\{3,4,5\}} = 0.1$.

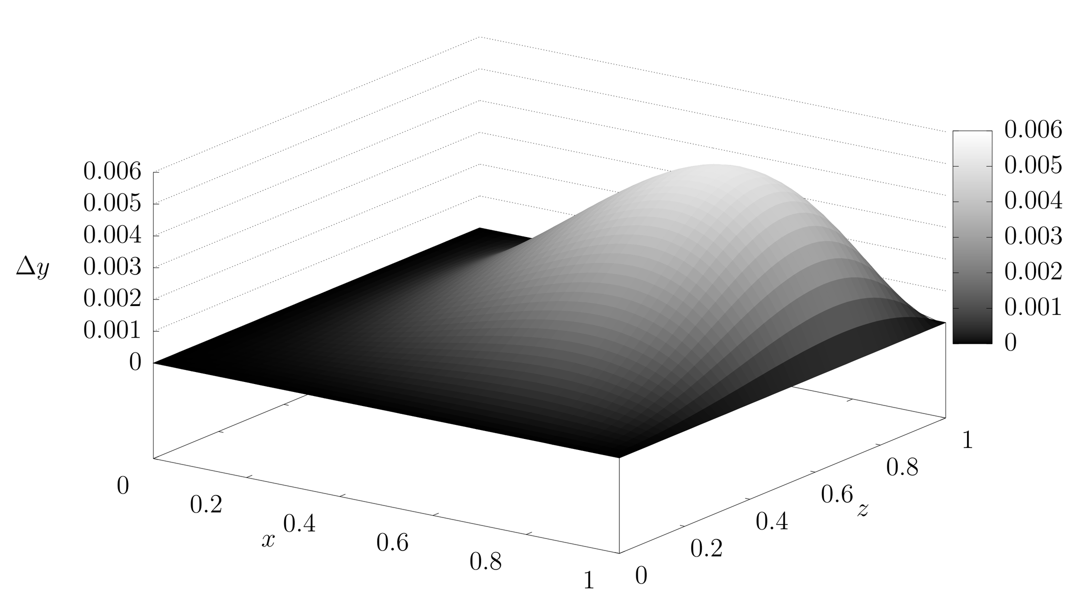

Figure 2: Transformation function

Linearised free surface solver

To reduce the required computational power and time, a linearised free surface solver is used. The solver is based on the method developed by Woolliscroft and Maki with several modifications: quasi-linear convection term, relaxation zones for efficient wave-structure interaction and 6DOF equations. The described solver needs approx. 75 minutes to reach a converged solution on one Intel Core i5-4570 processor (with 4 cores).

Results

Figure 3 shows a comparison of the original and a new morphed mesh deformed with the suggested transformation function.

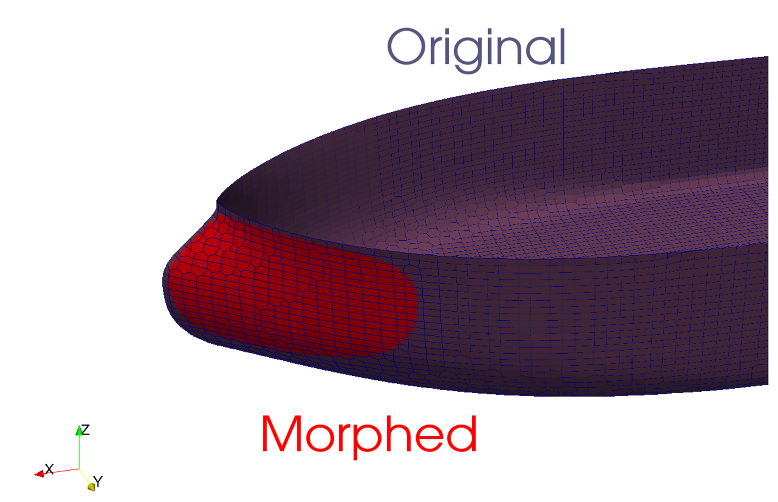

Figure 3: Mesh morphing

The following optimisation results are obtained with the `coliny_cobyla` (Constrained Optimization by Linear Approximations) optimisation method which is a local derivative-free method. Behaviour of optimisation parameters and normalised resistance against corresponding optimisation iterations are given in Figure 4.

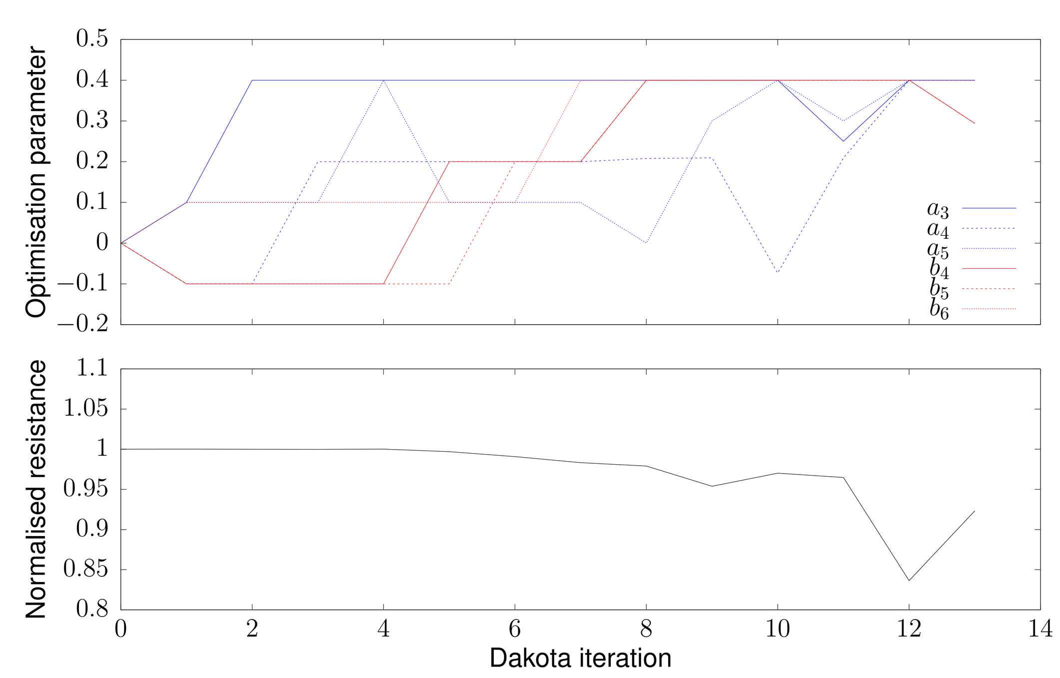

Figure 4: Optimisation

The proposed workflow is very flexible, since the transition to other optimisation methods or different transformation functions can be done in matter of minutes.

12th OpenFOAM Workshop
Exeter, 24th - 27th July 2017

8th Floor CFD@FSB
www.fsb.hr/cfd

Feel free to contact us at robert.keser@gmail.com or take a look at our YouTube channel: 8th Floor CFD@FSB.