# Surrogate Modeling with Python™

Effect of protruding a bulbous bow

Alberto Cominetti

February 2, 2017

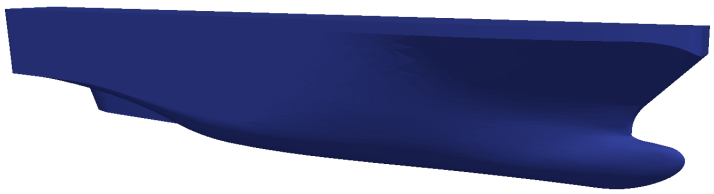
**Abstract**

*The aim of this work is to develop some examples of surrogate modeling in a python environment. To give this work a practical application, a one-variable shape optimization problem is presented. Given a ship geometry, its bulb is elonged or shortened and OpenFOAM simulations are performed to collect a set of data. Here focus is placed on surrogate techniques rather than on hydrodynamic aspects. Three different methods are presented: polynomial regression, radial basis function approach, kriging interpolation. Moreover, best model must be chosen, so two techniques to do this are presented: Cross-Validation and Maximum Likelihood Estimation.*

# Contents

# 1 Collecting the training data-set

Purpose of this work is find the better bulb configuration in order to reduce ship resistance. To simplify the code implementation only longitudinal deformations are applied to bulb geometry. Geometry considered is a 1:20 scale-model of the real ship ($L_{PP} = 4[m]$).



Using *CAMILO* shape morpher, bulb's nose is displaced forward and backward. Seven nose displacements are performed:

$$\underline{X} = \{-0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5\} \quad [m] \tag{1}$$

Then for each of these geometries an OpenFOAM solution is founded.

$$\underline{R} = \{70.74, 68.88, 67.38, 68.24, 65.67, 64.13, 64.86\} \quad [N] \tag{2}$$

OpenFoam solution set-up is inspired by Duisburg Test Case *interFoam* validation case. Case set-up main features are:

- Model scale $1:20:\ L_{PP} = 4[m]$
- Model velocity $= 2.301[m/s]$ ($20[knots]$ real scale)
- $T = 0.265[m]$, $\alpha = 0[deg]$
- $Cells \sim 800K$
- interFoam solver *(0-dof, VoF)*
- Quasi-steady solution *(Local Time Stepping)*

Mesh and dictionaries have been validated doing a comparison with available experimental data [TESI].

In order to give a better generalization about bulb deformations, a parameter described in literature [2] is used. This is called *length parameter* and is the protruding length of the bulb $L_{PR}$ (distance between noise and ship forward point) adimensionalized by the length between perpendicular $L_{PP}$ (distance between backward and forward points) that is a main parameter of the ship.

$$C_{PR} = \frac{L_{PR}}{L_{PP}} \tag{3}$$

Concluding, a data set is collected; this is called *training data-set* and upon it surrogate will be constructed.

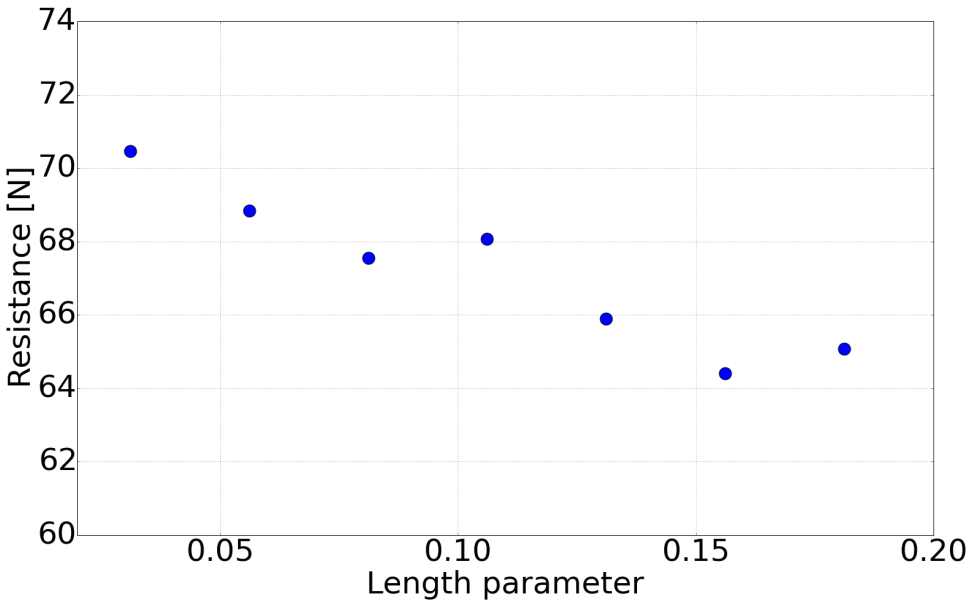



Figure 1: Representation of training data-set

## 2 Surrogate: why and what is?

Before dealing with surrogate building methods, it is necessary to explain what a surrogate is and which advantages it takes.

In this instance, interest is focused on finding the optimal bulb geometry, that is the one that minimizes hydrodynamic resistance.

Looking at Figure 1, optimization means finding the "lower" point. The problem is that only seven discrete points are known, and each of these involve hours of computation (11-hour running in parallel on 8 cores). Furthermore it is important to remember that a priori, resistance behaviour is unknown and, except some physical intuitions, practically unpredictable. Then using gradient-based methods or genetic algorithms can become heavily expensive, especially if the starting point is bad defined.

Here advantages of surrogate modeling comes into play.

If behaviour is unknown a first thing to do is exploring the variable design space, making some observations. This is exactly what have been explained in the section before.

This step give some insights of the problem. For example, the presence of two local minimums and that the optimum probably will be around $C_{PR} = 0.15$.

At this stage different choices can be made. For example, thanks to the information given by design space exploration, gradient-based and GA can be used more easily.

Otherwise the entire behavior can be reconstructed, passing from some discrete points to a continuous function called *Surrogate Model* (also known as *meta-model* or *response surface*).

This means that minimum research or any other optimization scopes, won't be made using an expensive high-fidelity model, but on a surrogate of it.

In other words, high-fidelity model (i.e. CFD simulation) will be used only to calculate a certain number of outcomes. Then these values will be used to built a meta-model (or surrogate) that predicts responses in other design points.

So, once a surrogate model is built, evaluating the quantity of interest becomes inexpensive, and any kind of optimization, calibration and investigation can be made.

# 3 Techniques implemented in Python

There is a wide variety of surrogate building techniques, some have very different approach.

Their goal is to learn from the training data, that is searching across the space of conceivable functions $\widehat{f}$ the one that better fits the training data. The bad news is that this space is infinite, the good one is that the overwhelming majority of these functions would be practically useless at predicting responses at new sites. On the manner such nonsensical predictor

are filtered out lies the variety of surrogate modeling approaches.

The one adopted in this work is to *hard-wire the structure of* $\widehat{f}$ *into the model selection algorithm and search over the space of its parameters to tune the approximation to the observations* [3].

Hence in each of the following techniques a structure is chosen and its *structural parameters* computed to fit the data; after that *tuning parameters* are varied to find the more "likely" surrogate.

In this work only three approaches are presented: *polynomial least square regression*, *radial basis functions approach* and *Kriging interpolation*.

## 3.1 Least square Polynomial Regression

$$\widehat{f}(x, m, \underline{\mathbf{w}}) = w_0 + w_1 x + w_2 x^2 + ... + w_m x^m = \sum_{i=0}^{m} w_i x^i \tag{4}$$

This is the structure of a polynomial response curve: polynomial coefficients $\underline{\mathbf{w}}$ are the *structural* parameters and polynomial degree $m$ is the *tuning* parameter.

Supposing polynomial degree is known, let's find the polynomial coefficients $\underline{\mathbf{w}}$. Applying polynomial equation to each training points a linear system can be constructed as follows:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & ... & x_1^m \\ 1 & x_2 & x_2^2 & ... & x_1^m \\ ... & ... & ... & ... & ... \\ 1 & x_n & x_n^2 & ... & x_n^m \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ ... \\ w_m \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ ... \\ R_n \end{bmatrix} \tag{5}$$

Writing in a more concise form:

$$\underline{\underline{\Phi}} \cdot \underline{w} = \underline{R} \tag{6}$$

where $\underline{\underline{\Phi}}$ is called *Vandermonde matrix*.

Generally, to solve this system a *least-squares analysis* is employed. This means finding the vector $\underline{w}^*$ that minimizes the residuals $||\underline{\underline{\Phi}}\ \underline{w} - \underline{R}||$. Using the theory behind least-squares, the more "likely" estimation of $\underline{w}$ can be calculated as:

$$\underline{w} = (\underline{\underline{\Phi}}^T \underline{\underline{\Phi}})^{-1} \underline{\underline{\Phi}}^T \underline{R} = \underline{\underline{\Phi}}^+ \underline{R} \tag{7}$$

Where $\underline{\underline{\Phi}}^+$ is called *Moore-Penrose pseudo-inverse* of $\underline{\underline{\Phi}}$.

Once coefficients have been determined, surrogate can be plotted.

Unfortunately, in building surrogates, order of the polynomial is usually unknown a priori. One could point out that the higher polynomials fits well

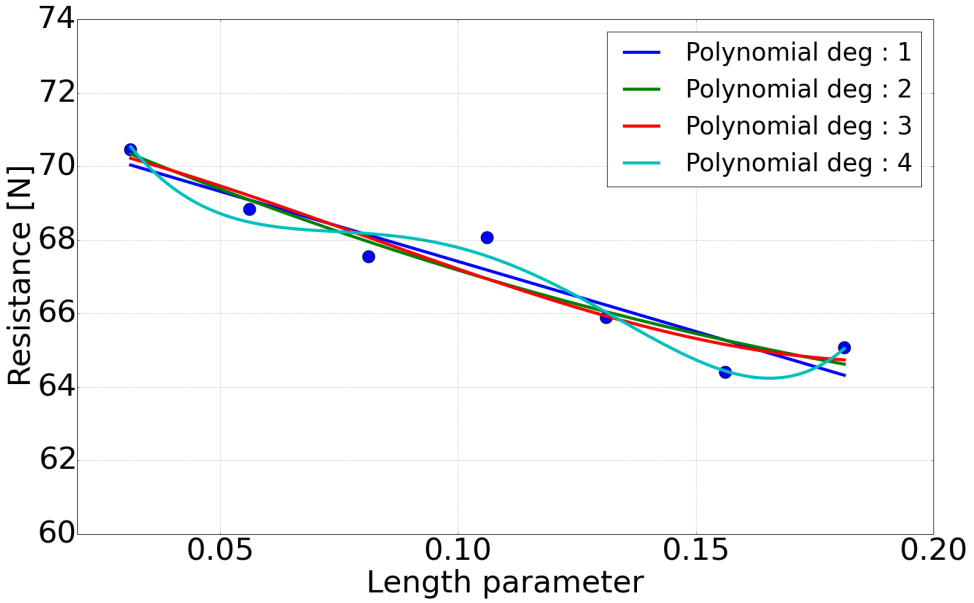



Figure 2: Polynomial fitting of different order (1,2,3,4)

the data, and this is true. But in surrogate modeling danger of *overfitting* the noise must be kept in mind.

Then another method must be used to *tune* the polynomial degree $m$ to best fit the data. Here *Leave-One-Out-Cross-Validation* is used.

First of all, it is necessary to measure the model accuracy. This can be done by defining a *loss function* representing the difference between model and observations. Obviously to do this other design points, out of the training data set, need to be used; these are called *Test Data*.

As previously said, observations could be very expensive and time-consuming, then have a set of data used only for testing may be not convenient.

A way to overcome this limit is splitting the available data set into $q$ subsets and building the model without considering one of these; this will be used later as test data-set.

Rotating the left-out subset, $q$ loss functions can be computed and that can help finding the better parameters.

This procedure is called *"q-Fold Cross Validation"*, when $q = n$ it is called *"Leave-One-Out Cross Validation"*. The latter is used in this instance, and its result are plotted in Figure 3. It seems that the best fitting is the linear trend.

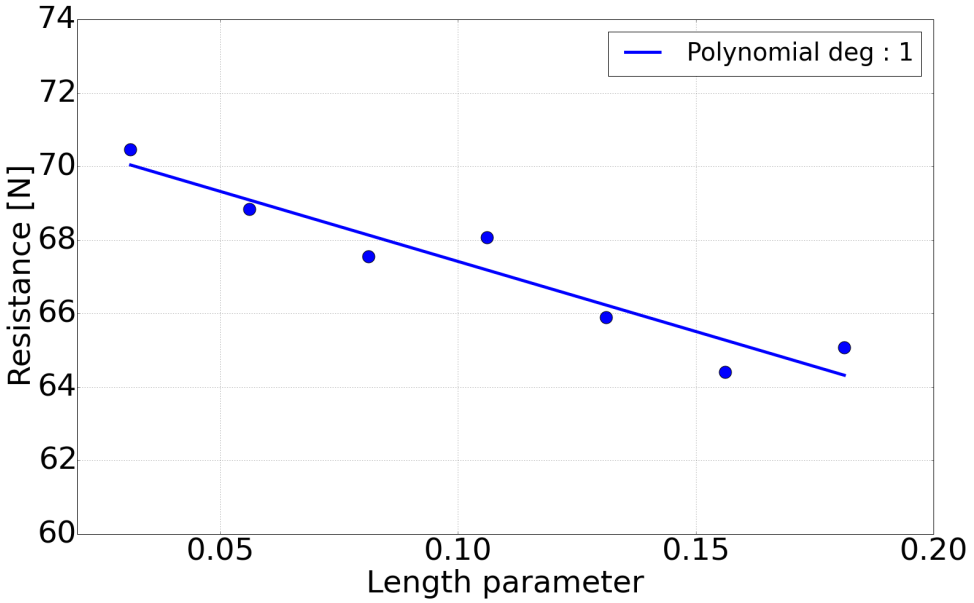



Figure 3: Polynomial fitting of different order (1,2,3,4)

## 3.2 Radial Basis Function Approach

$$\widehat{f}(x) = \underline{\mathbf{w}}^T \underline{\Psi} = \sum_{i=1}^{n_c} w_i \psi(\|x - c_i\|) \tag{8}$$

This is the structure used by Radial Basis Function approach: the essence is to represent a continuous smooth function as a combination of simple basis functions $\psi_i$, defined in $n_c$ centers $\underline{\mathbf{c}}_i$ and with their own weight $w_i$.

There is a wide variety of basis functions, here are some of this:

| | | | |
|---|---|---|---|
| Linear | $\psi(r) = r$ | Gaussian | $\psi(r) = e^{-r^2/(2\sigma^2)}$ |
| Cubic | $\psi(r) = r^3$ | Multiquadric | $\psi(r) = \sqrt{r^2 + \sigma^2}$ |
| Thin plate spline | $\psi(r) = r^2 \log(r)$ | Inverse multiquadric | $\psi(r) = \dfrac{1}{\sqrt{r^2 + \sigma^2}}$ |

Functions on the left are *fixed basis*, while those on the right are *parametric basis*. The difference is the presence of $\sigma$ that could be used as *tuning parameter*.

Now let's see how to estimate the weights $w$ for a *noise-free* case.

$$\widehat{f}(x_j) = \sum_{i=1}^{n_c} w_i \psi(\|x_j - c_i\|) = y_j \quad , \quad j = 1, ..., n \tag{9}$$

Herein lies the beauty of radial basis function approximation [3]. The above equation is a linear system in term of weights $w_i$ and if there is an RBF center for each training data ($n = n_c$) it has an unique solution. Even an highly non-linear response can be modeled by solving a linear system:

$$\underline{w} = \underline{\underline{G}}^{-1} \underline{R} \tag{10}$$

where $\underline{\underline{G}}$ is called *Gram matrix*.

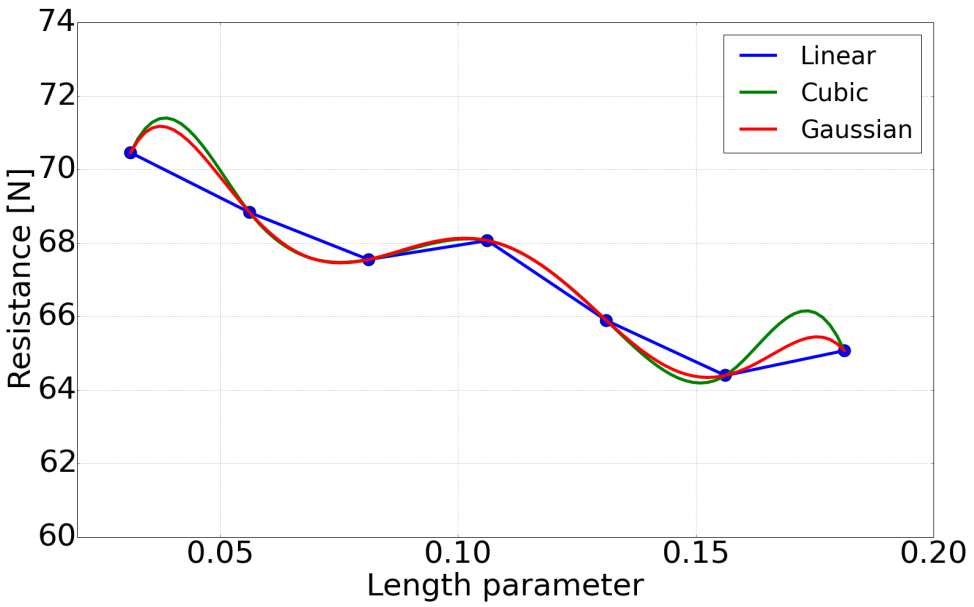



Figure 4: Radial basis function interpolation with linear, cubic or gaussian ($\sigma^2 = 0.05$) basis

There is a fundamental difference between this approach and the previous: interpolation includes training observations in the surrogate and predict others from these, regression try to learn information from such points and predict a general behavior. This represents an important limit of interpolation techniques when data are corrupted by noise. However there are some tricks that can help facing the noise.

For example, as suggested by Forrester et al. [3], it is possible to introduce an added model flexibility in the form of the *regularization parameter* $\lambda$ (Poggio & Girosi, 1990). This allow surrogate not to pass through the training points. Now weights are computed by solving:

$$\underline{w} = (\underline{\underline{G}} + \lambda \underline{\underline{I}})^{-1} \underline{R} \tag{11}$$

Also $\lambda$ is an unknown parameter, ideally related to noise variance, and can be fixed or estimated.

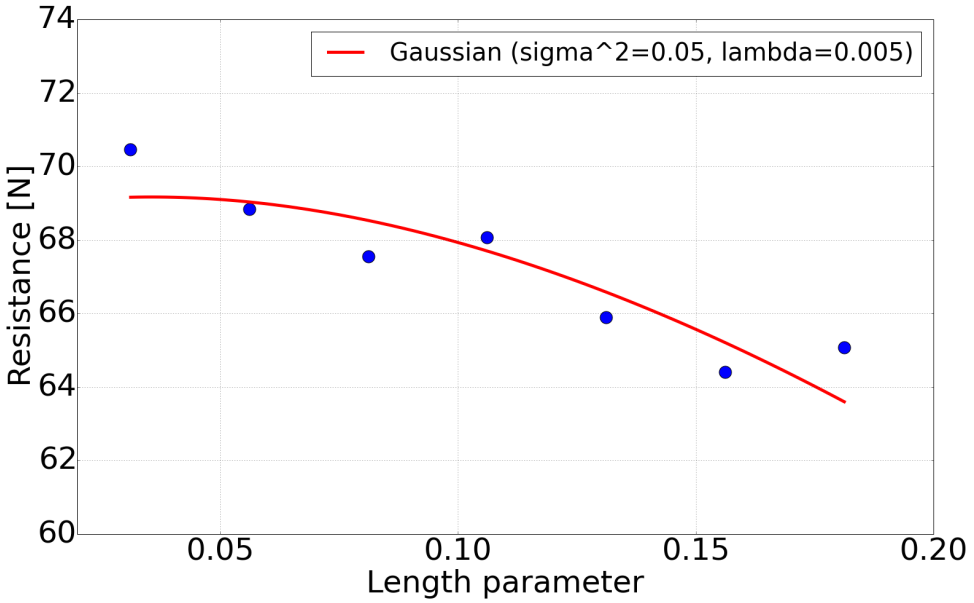



Figure 5: Radial basis function interpolation with gaussian basis ($\sigma^2 = 0.05$) and added model flexibility ($\lambda = 0.005$)

## 3.3 Kriging Interpolation

$$\widehat{f}(x) = t(x) + \varepsilon(x) \tag{12}$$

Kriging surrogate is the sum of a *trend function* $t(x)$ and a *gaussian process error model* $\varepsilon(x)$.

As suggested by *"Dakota v6.5 Theory Manual"* building a Kriging model involves mainly three steps:

- choice of a trend function

- choice of a correlation (kernel) function
- estimation of correlation parameters

Trend function could be a known costant (*simple kriging*), a general polynomial obtained with least square regression (*universal kriging*) or an unknown constant value (*ordinary kriging*).

As in many other Gaussian processes, a Bayesian approach is used, so observed responses will be viewed as if they are from a stochastic process. [3].

Under this assumption, each response can be defined by its *expected value* and *covariance function*.

$$E(R(x)) = \underline{f}^T(x) \cdot \underline{\beta} \tag{13}$$

$$Cov(R(x), R(x^*)) = \lambda \cdot kernel(x, x^*|\theta) \tag{14}$$

where in the first one $\underline{f}^T(x)$ are trend function basis and $\underline{\beta}$ their weights; in the second one $x^*$ is a point outside the training data-set, $\lambda$ is the process variance, and $\theta$ is the correlation function parameter.

Collecting all responses, an *observed "random" vector* is defined:

$$\underline{R} = \{R(x_1), \ ... \ , R(x_n)\} \tag{15}$$

By definition, the joint distribution of $\underline{R}$ satisfies [4]:

$$\underline{R} \sim N_n \left(t(x), \lambda \underline{\underline{K}}\right) \tag{16}$$

where $\underline{\underline{K}}$ is the $n \times n$ matrix of correlations between training points (kernel function applied between all training points).

Assuming to know trend and kernel functions parameters it is possible to compute *conditional expected value* and *conditional variance* of the process at an *untested location* $x^*$:

$$E(R(x^*)|R(x)) = \underline{f}^T(x^*)\underline{\beta} + \underline{k}_*^T \underline{\underline{K}}^{-1}(\underline{R} - \underline{\underline{F}} \ \underline{\beta}) \tag{17}$$

$$Var(R(x^*)|\underline{R}) = \lambda \left(1 - \underline{k}_*^T \underline{\underline{K}}^{-1} \underline{k}_*\right) \tag{18}$$

where $\underline{k}_*$ is the vector of correlations between untested point and training points, and $\underline{\underline{F}}$ is the $n \times q$ matrix of all $q$ trend basis functions at training points.

In this work the *squared exponential* is chosen as *correlation function*:

$$kernel(x, x^*|\theta) = e^{-\theta(x-x^*)^2} \tag{19}$$

As can be seen by its structure, this function correlates the outputs depending on the distance between inputs: very close inputs are strong correlated ($\sim 1$), while distant inputs are weak correlated ($\sim 0$).

This correlation can be tuned by varying the parameter $\theta$: augmenting its value will enhance correlation, zeroing it will make outputs independent between each other. Here, $\theta$ is a scalar because only one design variable is considered; if for example also deformations in $z$ direction are considered, two different $\theta$ can be defined.

Another parameter that could be used for tuning is the exponent; it could be varied in range $[0, 2]$ [4], but not to complicate the implementation this step has been avoided.

In this work only *Simple Kriging* and *Ordinary Kriging* are presented. A result of *Simple* approach is presented Figure 6.

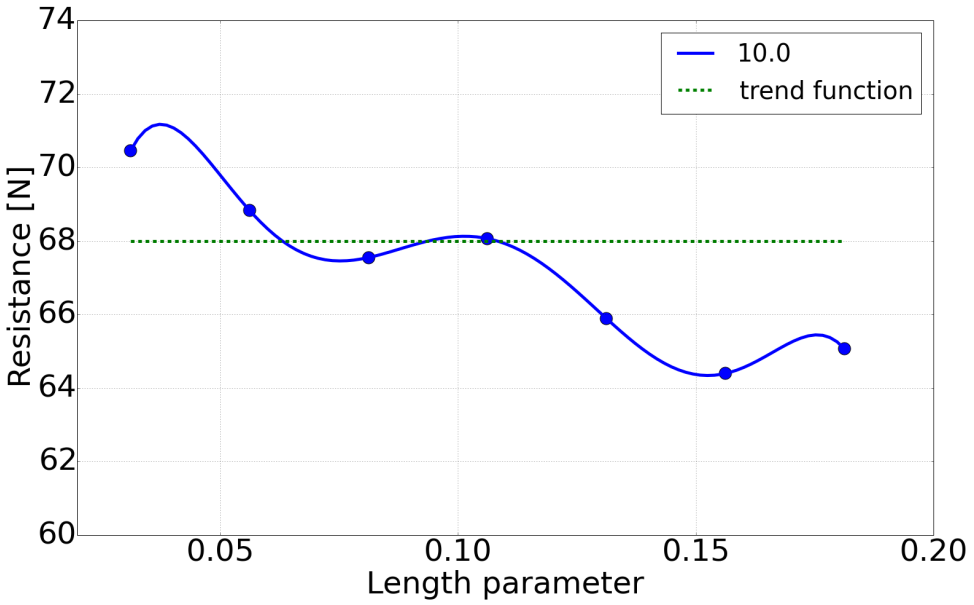



Figure 6: Simple Kriging interpolation with fixed constant trend function ($t(x) = 68$) and $\theta = 10$

Curve shown in figure seems to reasonably fits the data, but it is unknown if it fits "best" than others. Then a criterion to estimate how "likely" is the surrogate is required, in order to estimate kriging parameters: trend function coefficients $\underline{\beta}$ and tuning parameter $\theta$.

### 3.3.1 Tuning

Different criteria can be used to estimate parameters, but in this context only one them is presented: *Maximum Likelihood Estimation*.

The aim is to find the most "likely" multivariate normal distribution that fits the training data. Having parametrized mean and covariance functions, optimization is reduced to their tuning parameters $\underline{\beta}$ and $\theta$ [4].

Choosing a polynomial degree, a polynomial trend function can be found using a least-squares analysis as in the previous section. Also in this case *Leave-One-Out-Cross-Validation* could have been used to find the order, but not to add complexity, it has been avoided and a zero-order has been choosed.

Now only $\theta$ remains to be computed.

To discriminate which is the optimal value, an measure of how "likely" is the surrogate need to be introduced.

This can be accomplished by the *likelihood function*, that is the already presented equation (16). Making the logarithm:

$$\log f(\underline{R} \mid \underline{X}, \lambda, \theta) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log(\lambda^n|\underline{\underline{K}}| - \frac{1}{2\lambda}(\underline{R} - \underline{\underline{F}}\,\underline{\beta})^T\underline{\underline{K}}^{-1}(\underline{R} - \underline{\underline{F}}\,\underline{\beta}) \tag{20}$$

taking the negative and neglecting terms that don't influence optimization:

$$NLL \quad \propto \quad m\log\lambda + \log|\underline{\underline{K}}| + \frac{1}{\lambda}(\underline{R} - \underline{\underline{F}}\,\underline{\beta})^T\underline{\underline{K}}^{-1}(\underline{R} - \underline{\underline{F}}\,\underline{\beta}) \tag{21}$$

Using properties of *Negative Log Likelihood NLL* gradients, it is possible to calculate an optimum $\lambda$ for each iteration over $\theta$ . Hence the problem reduces to find $\theta$ that minimizes *NLL*.

Effect of $\theta$ can be seen in Figure 3.3.1, where four different surrogates are plotted ($\theta = 10^1, 10^2, 10^3, 10^4$).

As indicated by *NLL* diagram (Figure 7) most "likely" surrogates are for $\theta \sim 10^3$. Then case $\theta = 10^3$ will be used in the following for bulb shape optimization.

# 4 Conclusions

## 4.1 Comparison between methods

Since the practical purpose is to find the minimum, the best surrogate must be choosed. At first glance it can be seen that, even if they give a "clear" representation, first order polynomial (green curve) and Gaussian radial basis function with added flexibility (cyan curve) don't have any minimum.

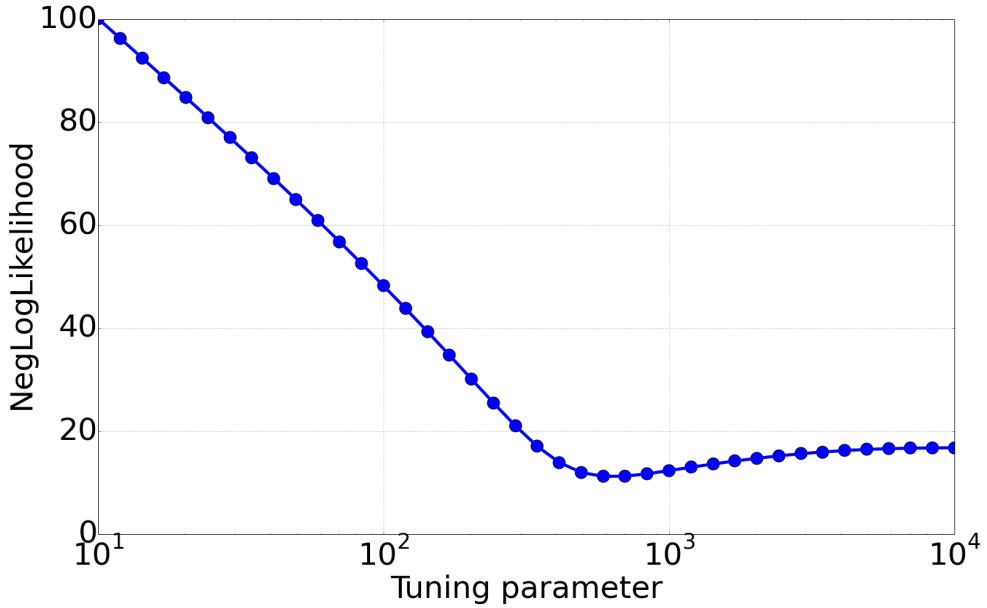



Figure 7: Negative Likelihood versus $\theta$ parameter in Ordinary Kriging interpolation (unknown constant trend function)

Hence remains the choice between 4-order polynomial, Gaussian RBF and Ordinary Kriging.

To help in this other two cases are computed near the global minimum: $0.35[m]$ and $0.45[m]$ elongations.

From this validation, best fitting surrogate seems to be the Gaussian radial basis function, but also Ordinary Kriging catch well the behaviour.

Since difference in Resistance value are neglectable one the two model can be selected (It was expected that have similar beahvior since Kriging is a "more sophisticated" version o Gaussian RBF).

A flaw of Gaussian RBF is the presence of two peaks near the edge of the design space. Probably these are non-physical and are caused by its mathematical definition.

## 4.2 Physical Intuitions

Some physical considerations can be made around results obtained.

First one is about *hydrodynamic*.

The decreasing behavior of ship resistance with bulb elongation was expected. In fact the bulb main effect is to modify the pressure field around

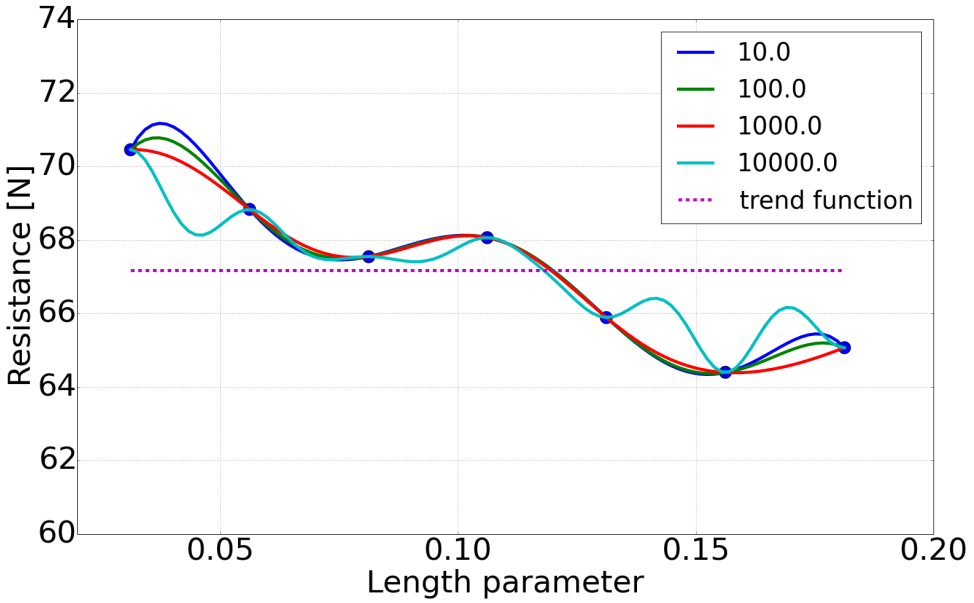



Figure 8: Ordinary Kriging surrogates for four different values of tuning parameter $\theta = 10^1, 10^2, 10^3, 10^4$

the hull by negative interfering with hull wave system. Hence a protrusion makes this resistance, called *wave resistance*, falling down.

On the other hand, longer bulb means more surface, then more viscous resistance. This could explain resistance's rise at the right limit of design space. To better understand this fact, pressure and viscous forces on the hull can be split and plotted.

Figure 11 raise an important consideration. While pressure force behaves as expected, viscous force has a "strange" behavior.

This trend of viscous force create a "plateau" that splits the design space in two minimum regions. One hazardous hypothesis is that this could be related to wave length since we are talking about interference, but this effect should have been seen on pressure component. A more realistic version is that, since an high velocity ($20[kn]$) is tested, some problems in solution arise.

The second consideration is about *Design*.

In fact the minimum found near $C_{PR} = 0.15$, corresponds to an elongment of about $0.4[m]$, that is 10% of ship characteristic length.

The basic hypothesis of fixed ship, that allows to use a quasi-steady

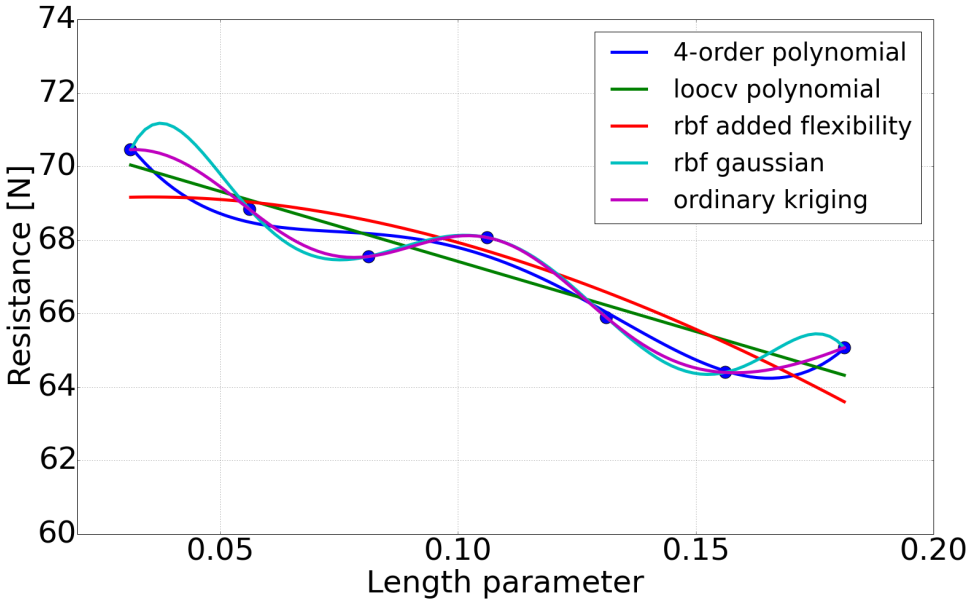



Figure 9: Comparison between different surrogate building methods.

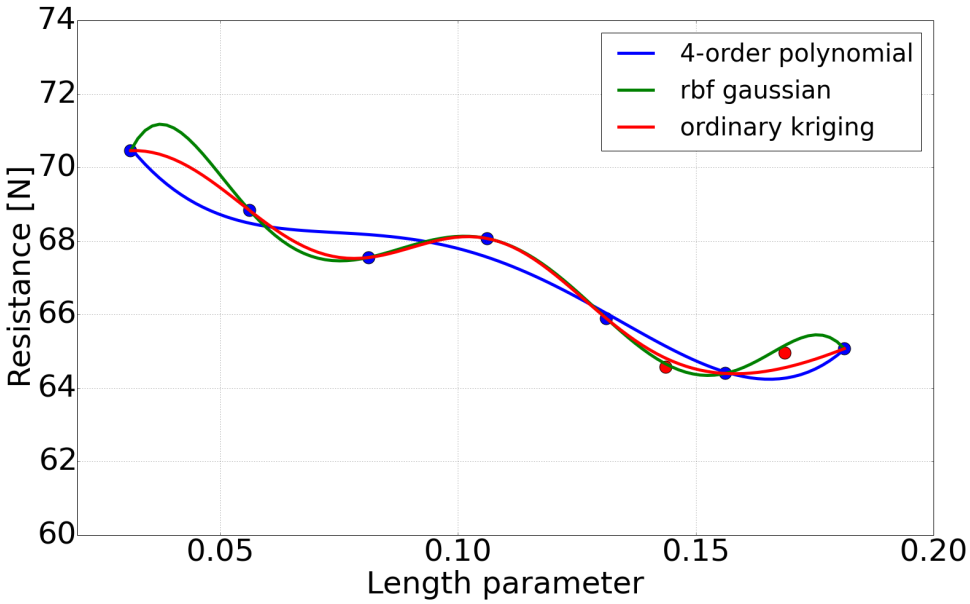



Figure 10: Validation test with two test points near global minimum

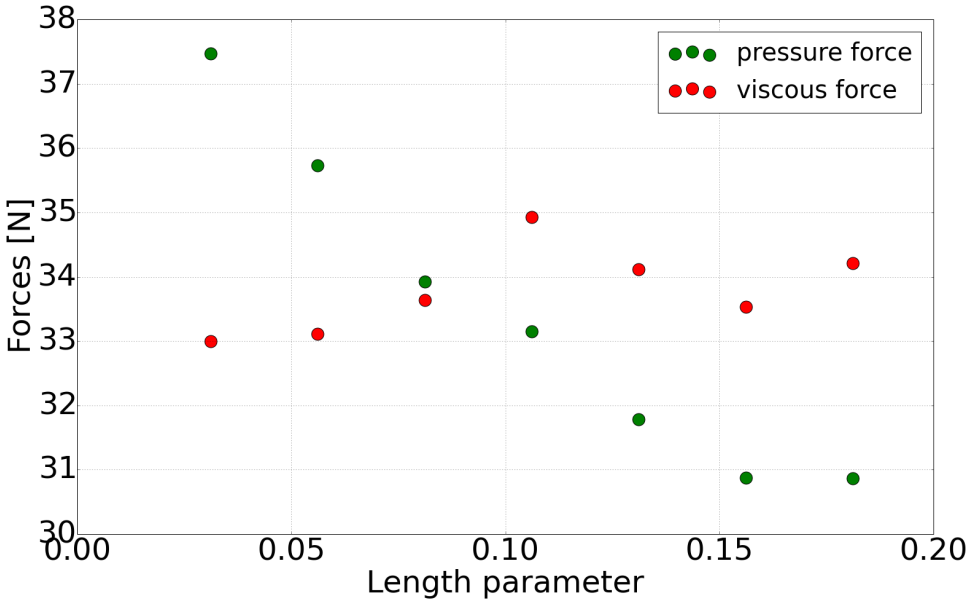



Figure 11: Effect of bulb elongation on pressure and viscous resistance

solver, becomes too strong to be accepted. Then an optimal solution must be searched for smaller elongations.

However, more accurate simulations should be done to give application and reality to these considerations.

As said in the introduction, aim of this work hasn't been neither the accurate study of ship's hydrodynamic, nor the achievement of the optimal feasible bulb. Rather the main purpose has been to give an example of how surrogate modeling can easily extrapolate information about physical behavior from a data set of responses. In this sense surrogate modeling represents a solid basis for further investigations when dealing with expensive optimization problem.

# References

[1] *https://github.com/OpenFOAM/OpenFOAM-4.x/tree/master/tutorials/multiphase/interFoam/ras/DTCHull*

[2] Kracht (1978), *"Design of Bulbows Bow"*, SNAME Transactions Vol.86

[3] Forrester, Sobester & Keane (2008), *"Engineering Design via Surrogate Modelling: a practical guide"*, Wiley, Southampton, UK

[4] McFarland (2008), *"Uncertainty analysis for computer simulations through validation and calibration"*, PhD Thesis in Mechanical Engineering, Nashville, Tennessee

[5] Chang & Zak (2001), *An Introduction to Optimization*

[6] , Wiley Rasmussen & Williams (2006), *Gaussian Processes for Machine Learning*, MIT Press