A data-driven reduced order modeling framework for shape optimization of marine propellers

Anna Ivagnes[1], Nicola Demo[1], Gianluigi Rozza[1], Gianluca Gustin[2] and Gianpiero Lavini[2]

[1] *Mathematics Area, mathLab, SISSA, International School of Advanced Studies, Trieste, Italy*
[2] *FINCANTIERI S.p.A., Trieste, Italy*

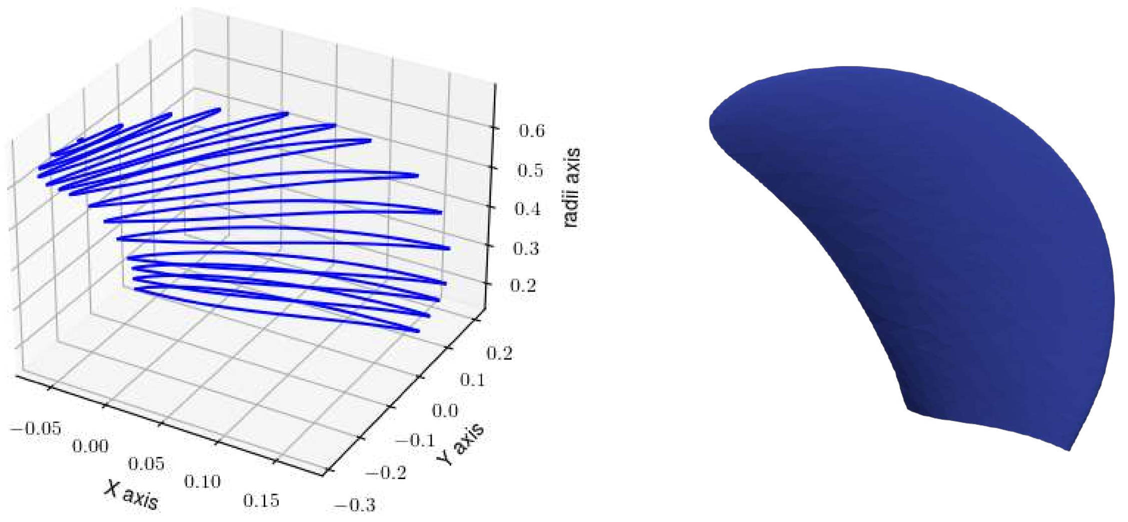
Introduction

The project aims to develop a propeller of **optimal** shape for cruise ships.
The shape optimization problem is faced exploiting **data-driven Reduced Order Models** (ROMs) to reduce the computational effort of high-fidelity fluid dynamic simulations of open water tests.

Goals of the project:

- ⋆ improve the **efficiency** of the propeller's blades;
- ⋆ avoid the **cavitation** phenomenon, i.e. the formation of vapor-filled cavities;
- ⋆ reduce **noise**, **vibration** and **consumption**.

1. Blade parametrization and deformation



Blade **geometric parameters**:

- ⋆ rake
- ⋆ skew
- ⋆ pitch
- ⋆ section parameters:
  - radius
  - camber
  - thickness
  - chord length

Blade deformation: design of 200 deformed blades

Deformed parameters:

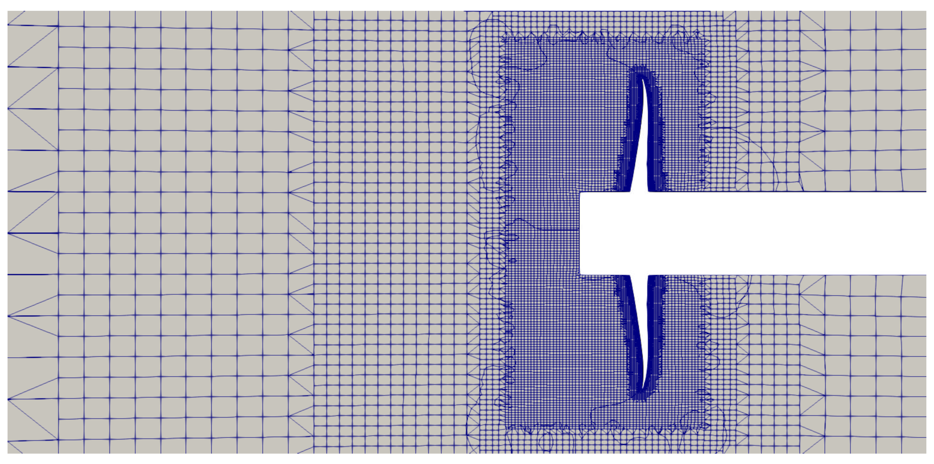
- ⋆ pitch (±10%)
- ⋆ camber (±10%)
- ⋆ thickness (±30%)
- ⋆ chord length (±30%)

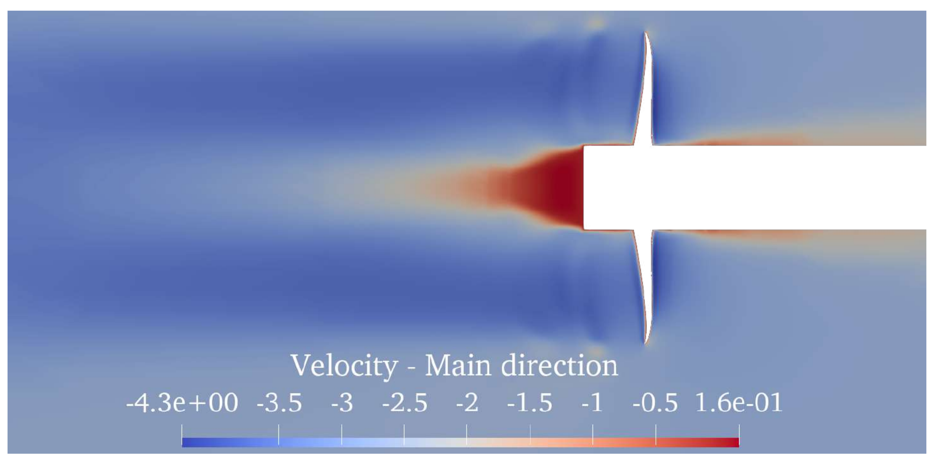
2. Setting of Full Order Model (FOM)

- **Type of simulation:** open water tests, with different inlet velocity v.
- **Mesh:** found in an iterative way in order to validate with experimental measures.

FOM mesh made with *snappyHexMesh* → **iterate** until : $\mathbf{err}_{k_T} \leq 1\%$, $\mathbf{err}_{k_Q} \leq 3\%$ → Validation with **experimental** results

Final mesh: ∼ 6 M cells.





- **Moving mesh technique:** Moving Reference Frame (MRF).
- **Turbulence modeling:**
  - ⋆ Reynolds Averaged Navier-Stokes (RANS) equations
  - ⋆ $\kappa - \omega$ SST model
  - ⋆ $\gamma - \mathrm{Re}_\theta$ turbulent transition.

Definition of forces/moments acting on blades

Thrust force:

$$T = T_{pressure} + T_{viscous}$$
$$T_{pressure} = \rho \int_{S_{bl}} p\mathbf{n} dA$$
$$T_{viscous} = \rho \int_{S_{bl}} \mathbf{\Sigma n} dA$$

Thrust coefficient:

$$k_T = \frac{T}{\rho n_{rps}^2 D^4}$$

Torque momentum:

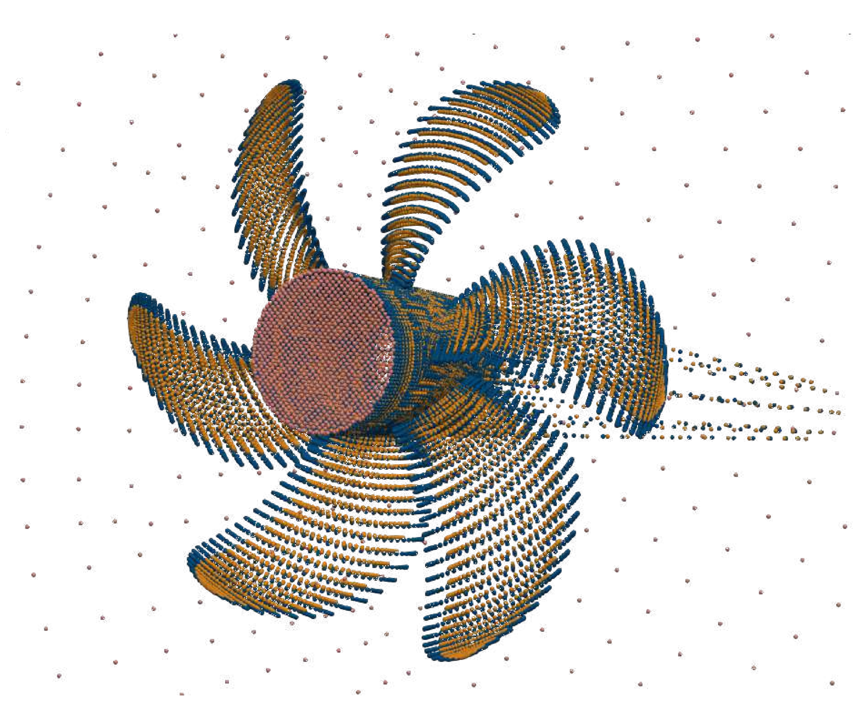
$$Q = Q_{pressure} + Q_{viscous}$$
$$Q_{pressure} = \rho \int_{S_{bl}} p\mathbf{n} \times \mathbf{r} dA$$
$$T_{viscous} = \rho \int_{S_{bl}} \mathbf{\Sigma n} \times \mathbf{r} dA$$

Torque coefficient:

$$k_Q = \frac{Q}{\rho n_{rps}^2 D^5}$$

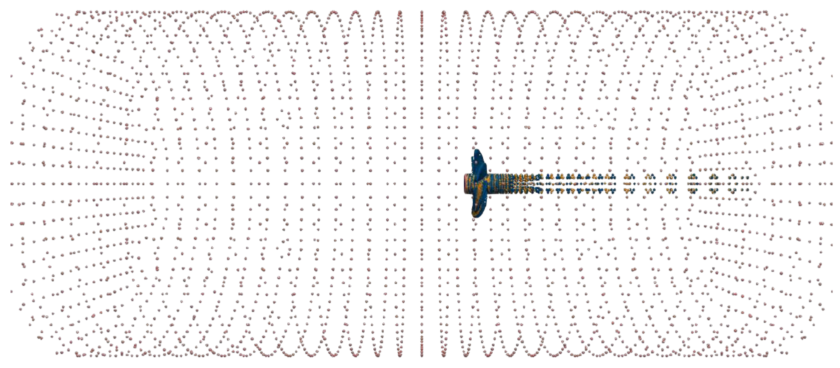
- ⋆ n_{rps}: rounds per second;
- ⋆ S_{bl}: blades surface;
- ⋆ $\mathbf{n}$: normal to surface;
- ⋆ D: propeller diameter;
- ⋆ $\mathbf{r} = (x, y, z)$: position vector;
- ⋆ $\mathbf{\Sigma}$: wall shear stress tensor.

3. OFFLINE stage: FOM simulations

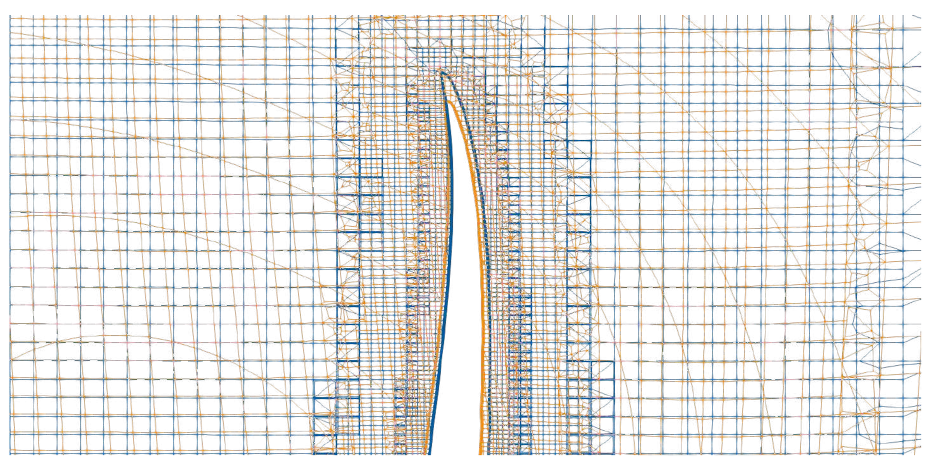


Mesh deformation with **Radial Basis Function (RBF)** interpolation technique from:

- **Undeformed control points** (undeformed propeller and boundaries)
- **Deformed control points** (deformed propeller and undeformed boundaries)



Slice of undeformed (—) and deformed (—) mesh:



a. Reconstruction of internal mesh.
Computational time: ∼ 2 h 15 min.

b. Simulation of a large number of open water tests with deformed blades and mesh.
Computational time to run one simulation: ∼ 48 − 72 h.

4. ONLINE stage: Shape Optimization exploiting ROM

Starting population of **parameters**

↓

Shape Optimization using **Genetic Algorithm**

Selection of the best individuals: **ROM** → prediction of **pressure** p and **wall shear stress** $\mathbf{\Sigma}$ → optimize the efficiency $\eta = \frac{Tv}{2\pi n Q}$

Mate/Crossover → **Mutation**

iterate for each **generation**

Remark: the Reduced Order Model is applied only on the **blades**. **Computational time**: ∼ 5 min.

↓

Optimal set of parameters

↓

Validation of efficiency in OpenFoam

5. Computational science and engineering softwares: `mathlab.sissa.it/cse-software`

BladeX
`github.com/mathLab/BladeX`
`mathlab.github.io/BladeX`

BladeX is a Python package for geometrical parametrization and bottom-up construction of propeller blades.

PyGeM
`github.com/mathLab/PyGeM`
`mathlab.github.io/PyGeM`

PyGeM is a Python package using Free Form Deformation, Radial Basis Functions, and Inverse Distance Weighting to morph complex geometries.

EZyRB
`github.com/mathLab/EZyRB`
`mathlab.github.io/EZyRB`

EZyRB is a Python library for data-driven (non-intrusive) model order reduction with POD with interpolation.

References

[1] N. Demo, G. Ortali, G. Gustin, G. Rozza, and G. Lavini. An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques. *Bollettino dell'Unione Matematica Italiana*, Nov 2020.

[2] N. Demo, M. Tezzele, A. Mola, and G. Rozza. Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing. *Journal of Marine Science and Engineering*, 9(2), 2021.

[3] M. Tezzele, N. Demo, A. Mola, and G. Rozza. An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics. In M. Günther and W. Schilders, editors, *Novel Mathematics Inspired by Industrial Challenges*, number X in Mathematics in Industry. Springer International Publishing, 2022.

[4] M. Tezzele, N. Demo, G. Stabile, A. Mola, and G. Rozza. Enhancing CFD predictions in shape design problems by model and parameter space reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1):40, Oct 2020.