Development of a Fast Wind Prediction Tool to Assess and Optimize UAV Takeoff and Landing on Vessels

Gonzalo Veiga Piñeiro (PhD Student)
Elena B. Martin Ortega (Full Prof. and Supervisor)

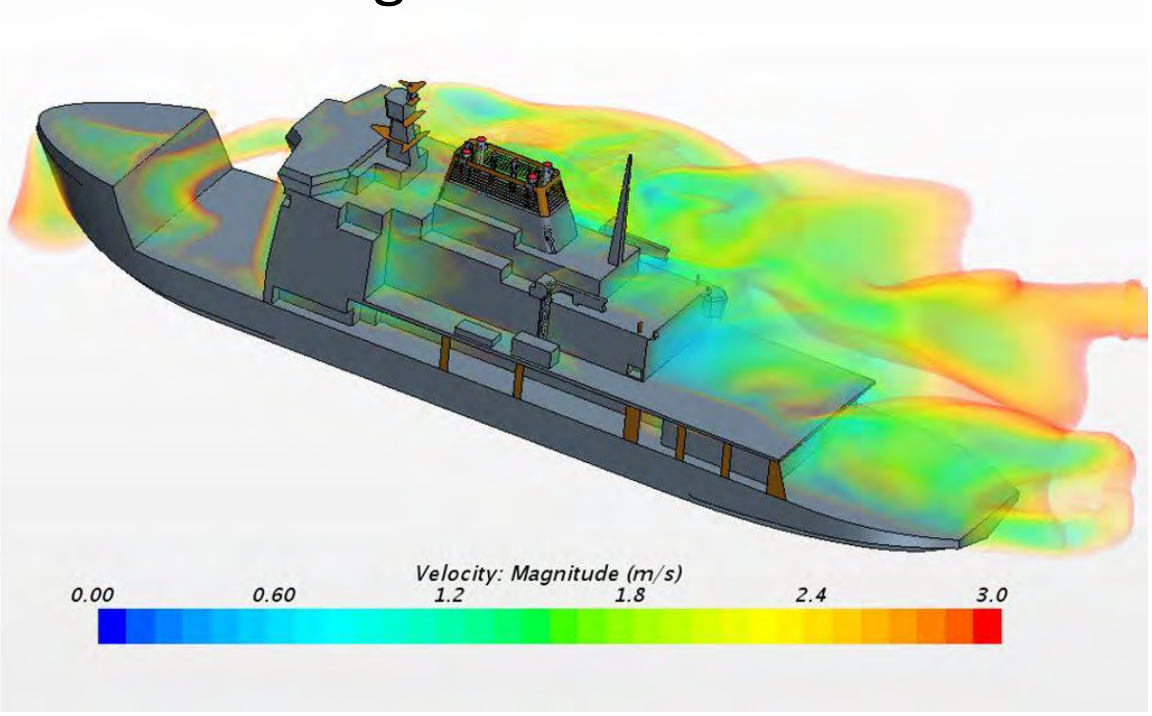
Institute of Physics and Aerospace Sciences (IFCAE)
Universidade de Vigo
Email: gonzalo.veiga@uvigo.gal

Universidade Vigo

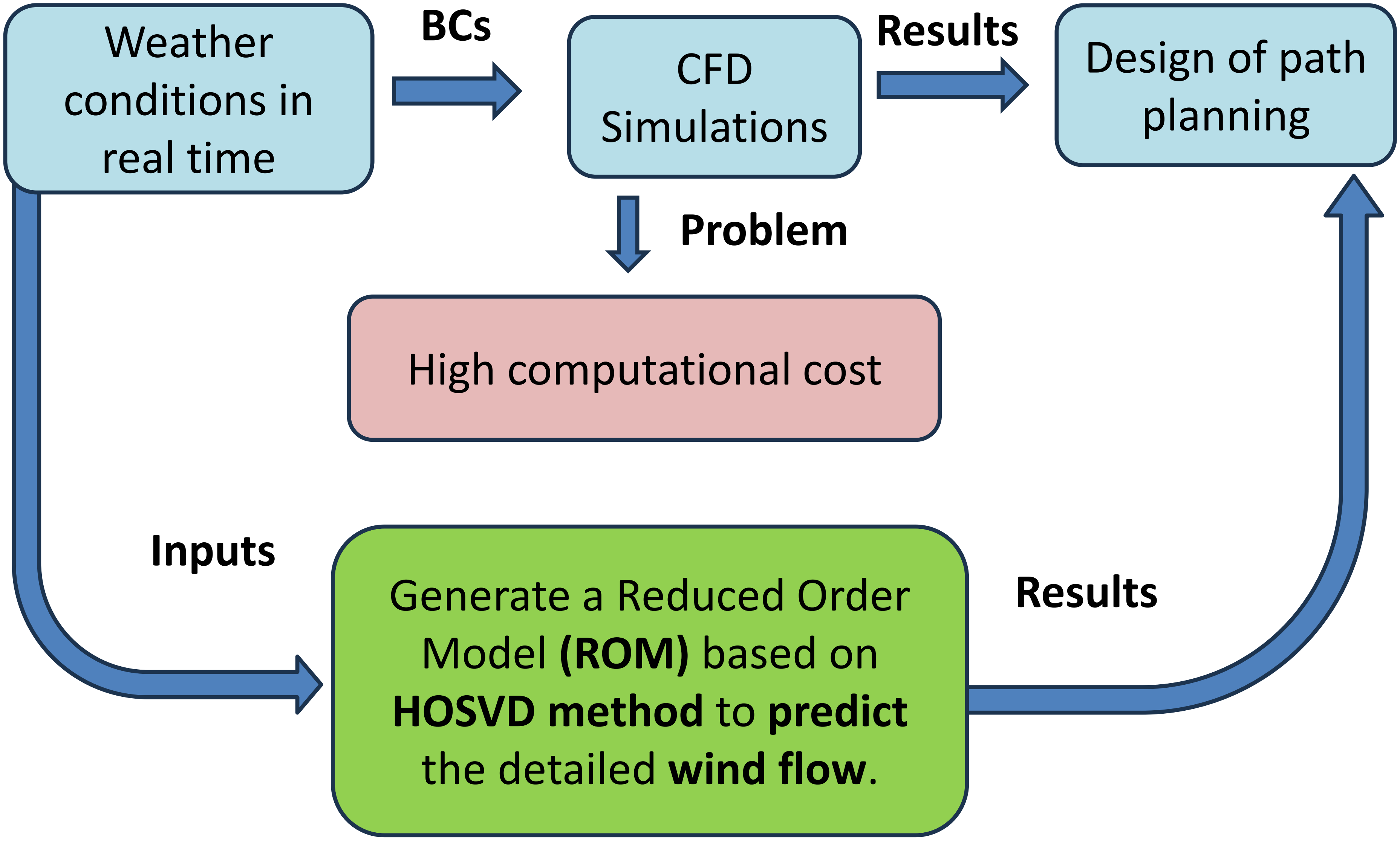
IFCAE
Instituto de Física e Ciencias Aeroespaciais

Introduction

- EAGLE Research Project (TED2021-129757B-C3)
- Path planning design for manteinance operations of wind off-shore turbines
- Predictive tool to understand the behavior of the air
- Takeoff and landing from support vessels
- Ensure a safe flight



Challenge and objectives



Methodology

CFD simulations:
Parametrization of wind BC

↓

Generation of database tensor (BBDD)

↓

HOSVD (Unsupervised ML algorithm) +
Interpolation techniques

↓

Predictive tool

CFD Simulations: Fluid Domain & Mesh

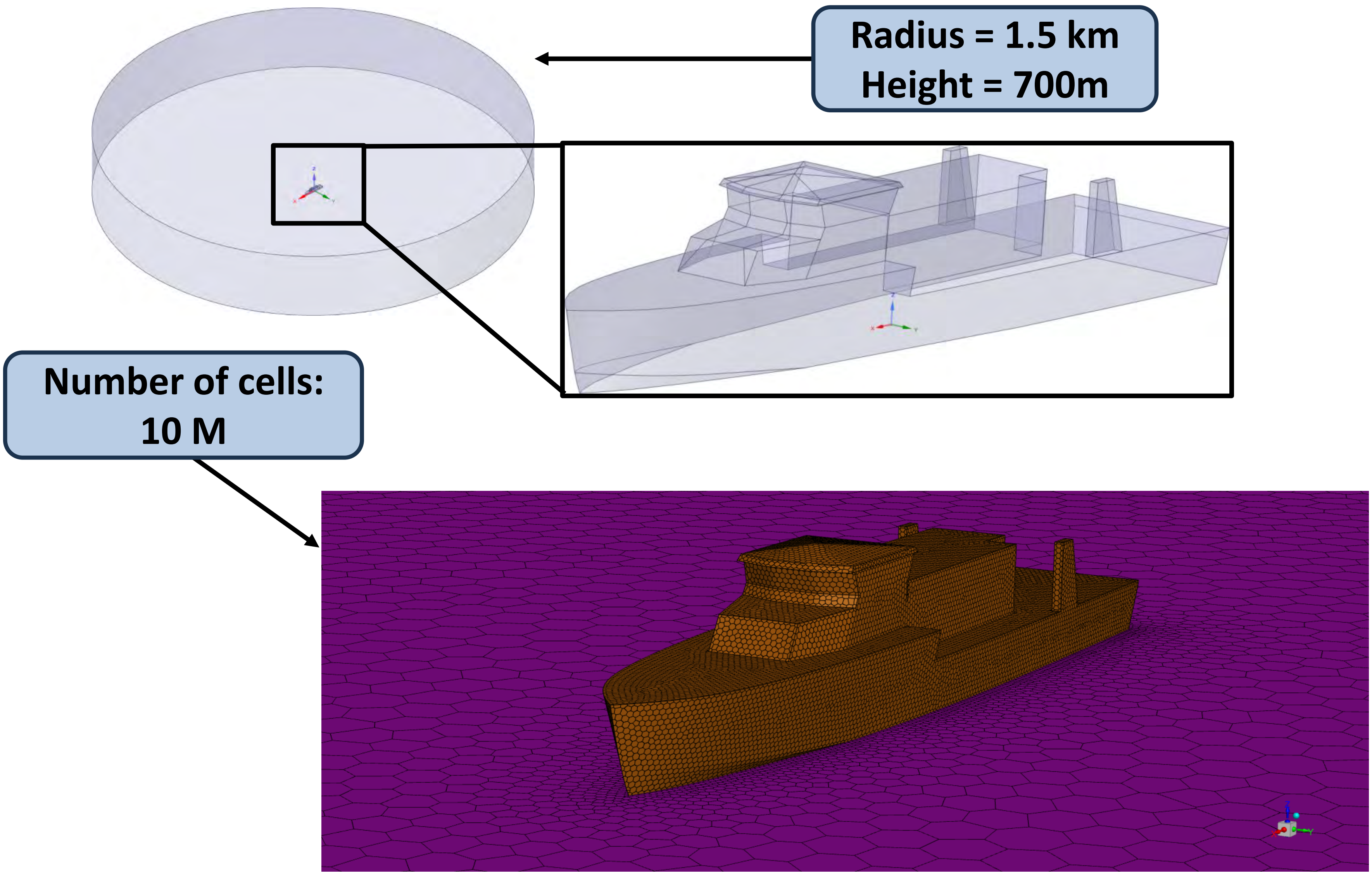


Fig. 2: Fluid 3D domain and mesh

CFD Simulations: Setup

- Software: **OpenFoam (v10.org)**

- RANS Model**: k-epsilon** (validated against wind tunnel test)

$$\nabla \cdot \langle \mathbf{u} \rangle = 0$$

$$\rho_\infty \langle \mathbf{u} \rangle \cdot \nabla k = \bar{\bar{\tau}}^{Re} : \nabla \langle \mathbf{u} \rangle - \rho_\infty \varepsilon + \nabla \cdot \left((\mu + \frac{\mu_t}{\sigma_k}) \nabla k \right)$$

$$\rho_\infty \langle \mathbf{u} \rangle \cdot \nabla \langle \mathbf{u} \rangle = -\nabla \langle p \rangle + \rho_\infty \mathbf{g} + \nabla \cdot (\langle \bar{\bar{\tau}} \rangle + \bar{\bar{\tau}}^{Re})$$

$$\rho_\infty \mathbf{u} \cdot \nabla \varepsilon = C_{\varepsilon_1} \frac{\varepsilon}{k} \bar{\bar{\tau}}^{Re} : \nabla \langle \mathbf{u} \rangle - \rho_\infty C_{\varepsilon_2} \frac{\varepsilon^2}{k} + \nabla \cdot \left((\mu + \frac{\mu_t}{\sigma_\varepsilon}) \nabla \varepsilon \right)$$

- Solver: **simpleFoam** (pressure-velocity segregation approach)

- **GAMG Solver** for **pressure** and **smoothed Gauss-Seidel** for the **rest of variables**

- **Tolerance:** $\mathbf{10^{-4}}$ and **Relaxion Factor: 0.8 (velocity), 0.3 (other variables)**

CFD Simulations: Boundary conditions

Surface	BC Type
Lateral cylinder (Inlet-Outlet)	Inlet-Outlet (ABL Function)
Bottom cylinder (Sea behaviour)	Moving Wall and Marine rugosity (Hersbach et al. [3])
Top cylinder (Undisturbed flow)	Symmetry
Vessel	Wall

- **Hersbach model:** $z_0 = \frac{z_{ref}}{exp(b_n^{fit}) - 1}$, $b_n^{fit} = [(b_n^{\nu})^p + (b_n^{\alpha})^p]^{1/p}$

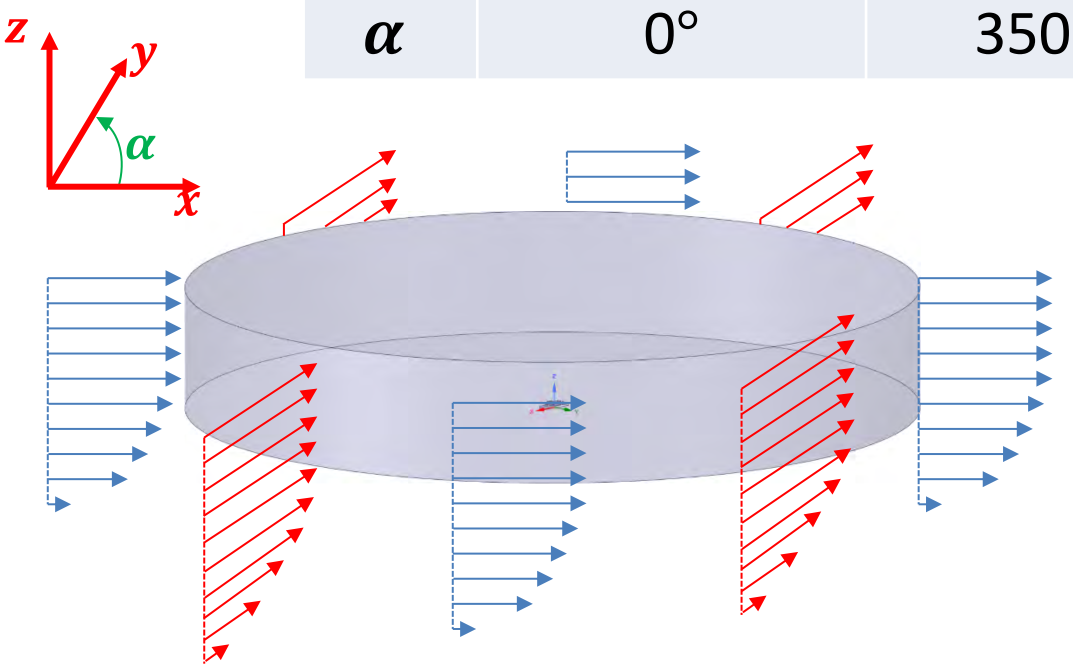
- **Atmospheric Boundary Layer (ABL):**

$$u(z) = \frac{u^*}{\kappa} ln\left(\frac{z-d+z_0}{z_0}\right) \qquad u^* = \frac{u_{ref}}{\kappa} ln\left(\frac{z_{ref}+z_0}{z_0}\right)$$

CFD Simulations: Boundary conditions

- To generate the BBDD, $\boldsymbol{u_{ref}}$ and the inlet flow direction, $dir = (\cos(\alpha), \sin(\alpha), 0)$, were parametrized
- Variables provided by weather forecast server

	Initial value	Final value	Increment
$\boldsymbol{u_{ref}}$	2 m/s	20 m/s	2 m/s
$\boldsymbol{\alpha}$	0°	350°	10°



Example of two different directions: blue ($\boldsymbol{\alpha = 0}$°) and red ($\boldsymbol{\alpha = 40}$°)

CFD Simulations: Numerical BBDD

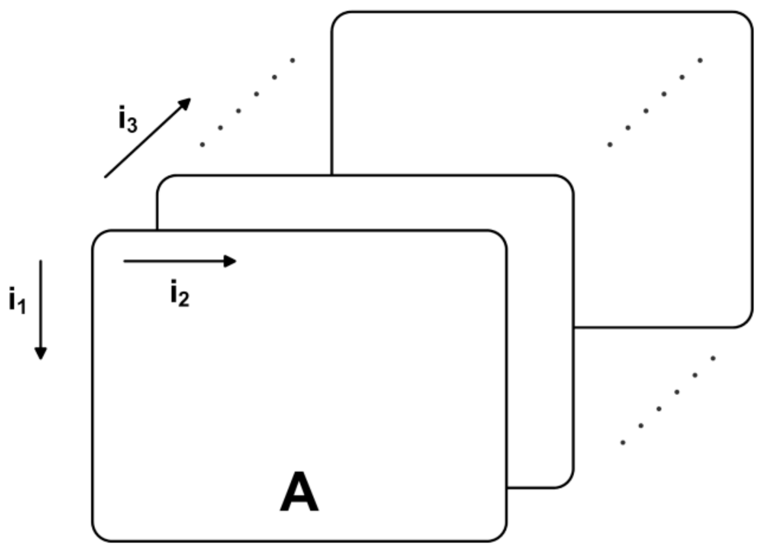
CFD Results

⇩

6 tensors (1 tensor per variable)

- Each simulation → 6 results (**U**, p, k, eps)
- Each result is a file that contains the solution (in column form) in the same order as the ID of the cells

- 1st mode (i1) sweeps spatial information: ID cells
- 2nd mode (i2) sweeps different velocities ($\boldsymbol{u_{ref}}$)
- 3rd mode (i3) sweeps different directions ($\boldsymbol{\alpha}$)



Example. $\boldsymbol{T_p(i,j,k)}$: Pressure calculated in **cell i**, **j-th** inlet **velocity** and **k-th** inlet **direction case**

Wind Predictive Tool

- To generate the Reduced Order Model (ROM), the following techniques have been applied:

  - **High Order Singular Value Decomposition (HOSVD):** is an unsupervised machine learning algorithm based on the POD method: **dimensionality reduction algorithm**

  - **Interpolation techniques**: once the HOSVD is calculated, **cubic spline interpolations** are performed on the mode matrices to predict new cases.

Wind Predictive Tool: HOSVD

- The **High Order Singular Value Decomposition (HOSVD)** method appears as an **extension** of the **SVD** method

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \qquad \text{Eq. (1)}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the **singular values** ordered from largest to smallest ($\boldsymbol{\sigma_1} \geq \boldsymbol{\sigma_2} \geq \cdots \geq \boldsymbol{\sigma_r} \geq \mathbf{0}$)
- This method sorts the system information (or patterns) by its relevance.

- The previous expression can be written in terms of components, which allows to retain only the first $\boldsymbol{k}$ singular values

$$\mathbf{A}_{ij} = \sum_{l=1}^{r} \boldsymbol{\sigma}_l \boldsymbol{u}_{il} \boldsymbol{v}_{jl} \approx \sum_{l=1}^{k} \boldsymbol{\sigma}_l \boldsymbol{u}_{il} \boldsymbol{v}_{jl} \qquad \text{Eq. (2)}$$

Wind Predictive Tool: HOSVD

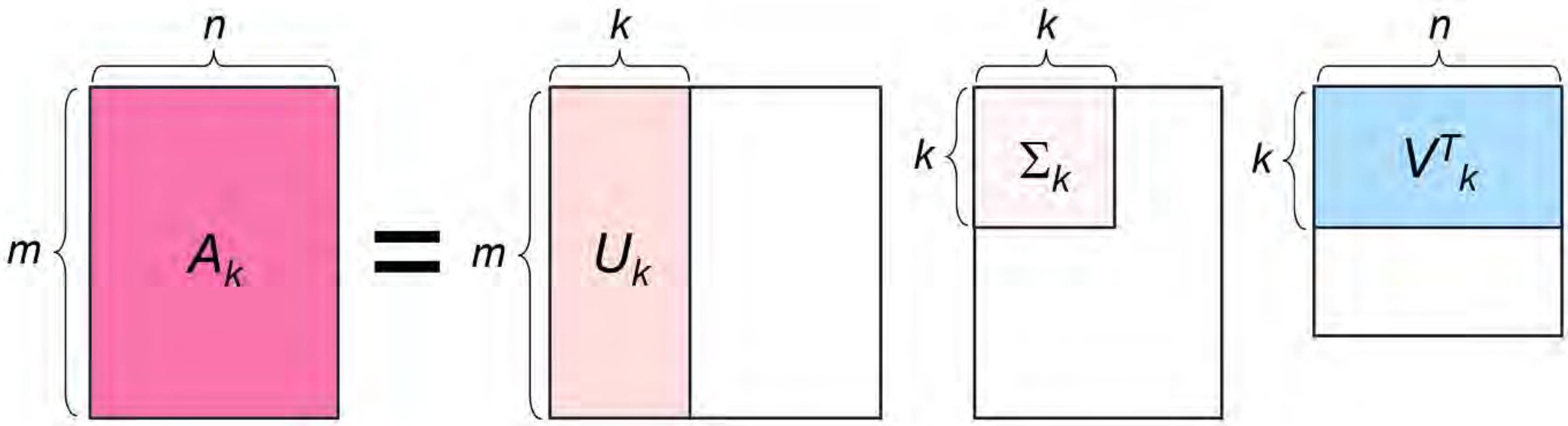


Fig. 5: Graphical example of SVD method

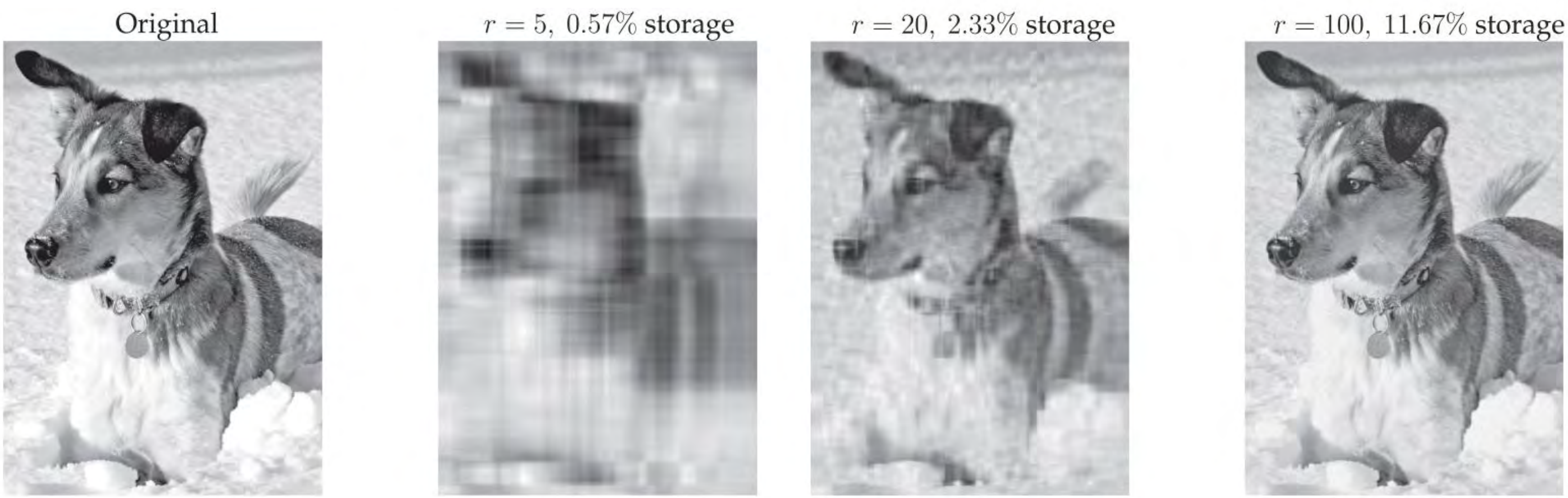


Fig. 6: Figure obtained from Brunton et al. [2]. Image compression of Mordecai the snow dog, truncating the SVD at various ranks r.

Wind Predictive Tool: HOSVD

- If we consider a tensor $\mathbf{T}$ with size $I_1 \times I_2 \times \cdots \times I_N$, the HOSVD allows decomposing the tensor as the following product

$$\mathbf{T} = \boldsymbol{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(\mathrm{N})} \qquad \text{Eq. (3)}$$

- $\boldsymbol{S}$ is the tensor core, $\mathbf{U}^{(i)}$ are mode matrices and $\times_i$ operators are n-mode products (this is the product of tensor by a matrix).

$$\mathbf{T}_{i_1,i_2,\ldots,i_N} = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \ldots \sum_{j_N=1}^{r_N} s_{j_1 j_2 \ldots j_N} u_{i_1 j_1}^{(1)} u_{i_2 j_2}^{(2)} \ldots u_{i_N j_N}^{(N)}$$

$$\approx \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \ldots \sum_{j_N=1}^{k_N} s_{j_1 j_2 \ldots j_N} u_{i_1 j_1}^{(1)} u_{i_2 j_2}^{(2)} \ldots u_{i_N j_N}^{(N)} \qquad \text{Eq. (4)}$$

Wind Predictive Tool: HOSVD

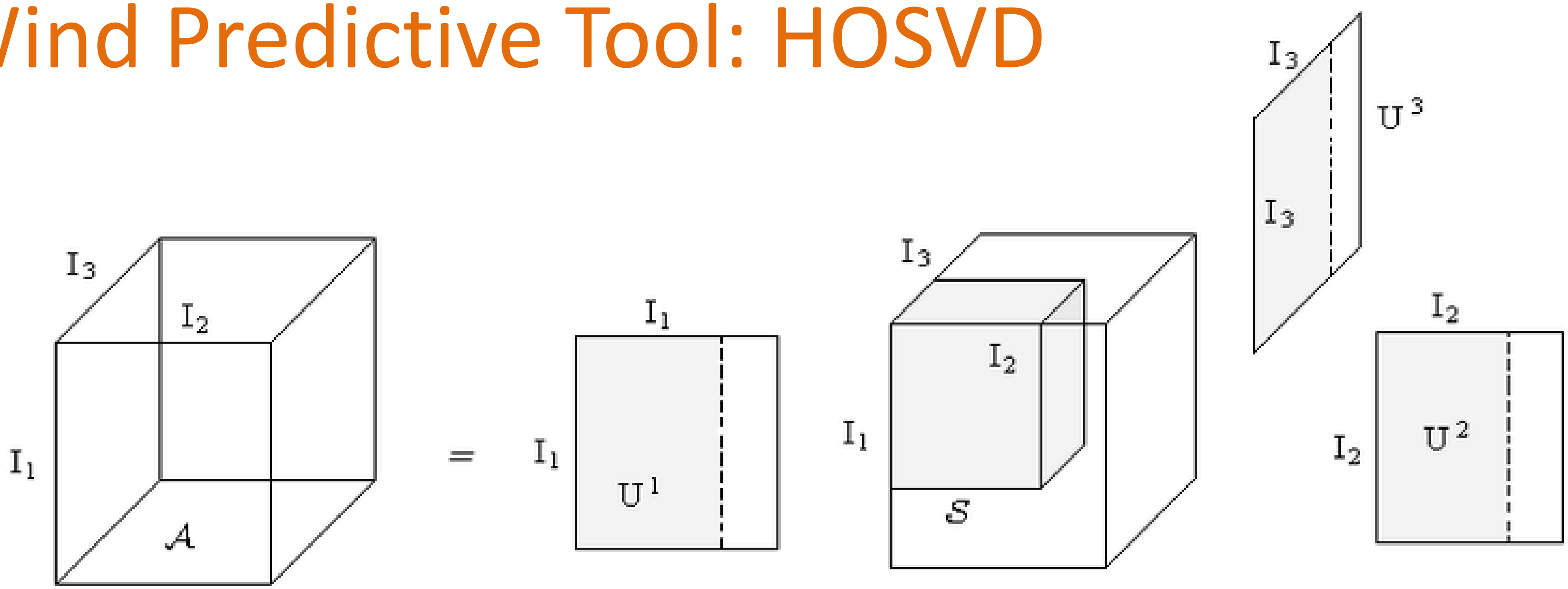


Fig. 7: Graphical example of HOSVD algorithm for 3D tensor

- The columns of $\mathbf{U}^{(i)}$ describes patterns that represent how the information is affected by the: $(i = 1)$ spatial variation (mesh cells), $(i = 2)$ different velocities or $(i = 3)$ different angles.

- The elements of the tensor $\boldsymbol{S}$ weigh the effect that the columns of the different matrices have on reconstructing the solution.

Wind Predictive Tool: Predictions

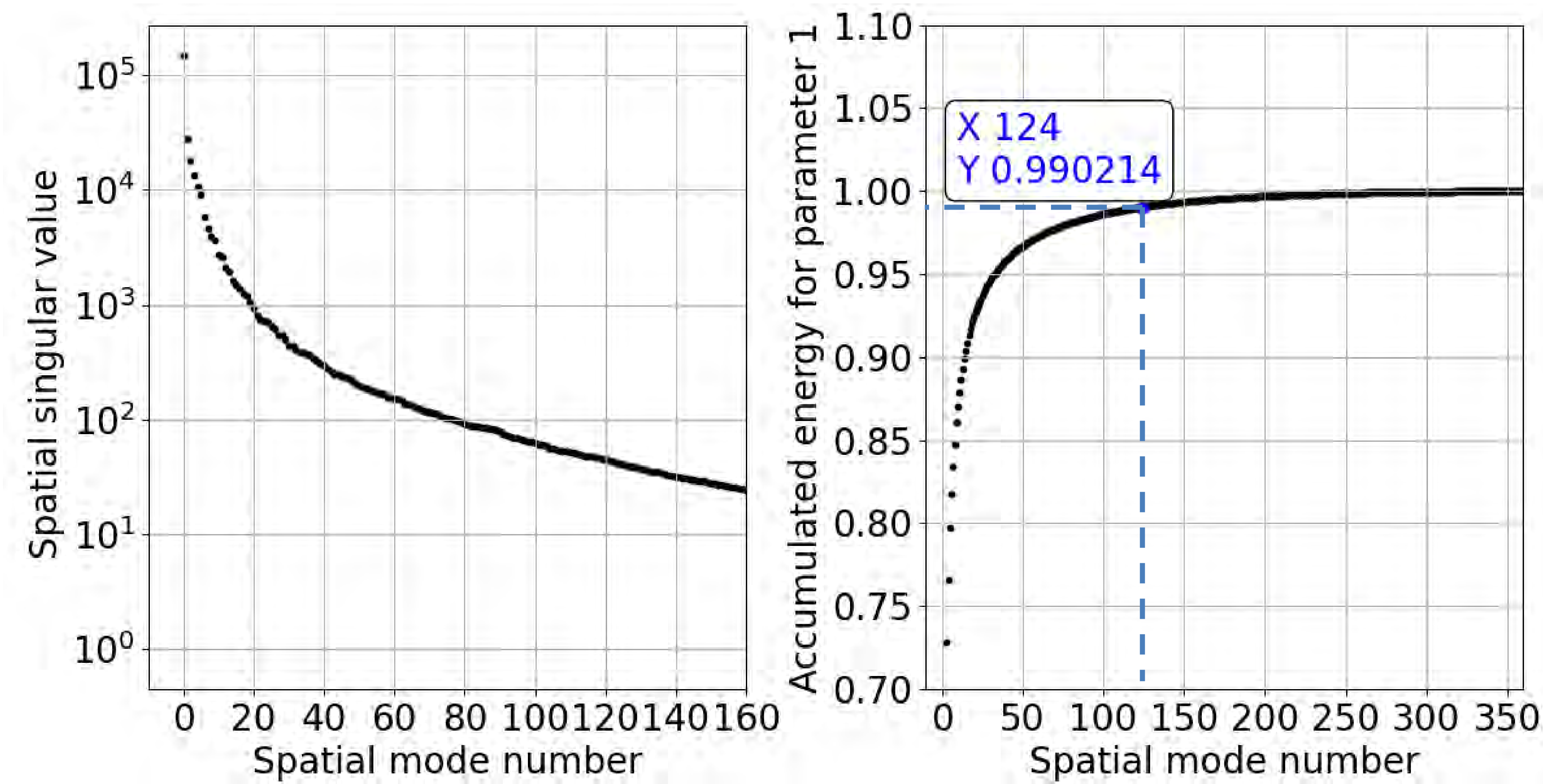
i. Perform a **truncation** based on the retained information
ii. **Interpolate** the mode matrices
iii. **Reconstruct** the solution

i. Truncation. To perform truncation, it's necessary to know the accumulated energy (**A.E.**). This measures the **quality of the approximation** when using a subset of the most significant singular values.

$$\mathbf{A.E.}_{\boldsymbol{n}} = \frac{\sum_{i=1}^{k_n} (\sigma_i^n)^2}{\sum_{i=1}^{r_n} (\sigma_i^n)^2} \qquad \text{Eq. (5)}$$

$\boldsymbol{n}$ is the parameter

Wind Predictive Tool: Predictions



Wind Predictive Tool: Predictions

ii. Interpolation. It is necessary to interpolate the mode matrices, thereby obtaining the **interpolated vectors**.

$$
\begin{array}{l}
u_1 = 2 \longrightarrow \\
\cdots \\
u_4 = 8 \longrightarrow \\
\boxed{u = 9} \longrightarrow \\
u_6 = 10 \longrightarrow \\
\cdots \\
u_{10} = 20 \longrightarrow
\end{array}
\begin{pmatrix}
0.76 & 0.2 & & 0.03 \\
\cdots & \cdots & & \cdots \\
0.6 & 0.5 & & 0.4 \\
? & ? & \cdots & ? \\
0.33 & -0.4 & & 0.23 \\
\cdots & \cdots & & \cdots \\
0.23 & 0.1 & & 0.9
\end{pmatrix}
$$

Necesary compute the interpolated vector $\boldsymbol{i}\mathbf{U}^{(\mathbf{2})}$.
Example: $\boldsymbol{u_{ref}} = \mathbf{9}$ (is not in BBDD).

1D cubic splines must be performed for each column and **evaluate** each spline at the corresponding point.

Wind Predictive Tool: Predictions

iii. Reconstruction. The solution for each cell in the mesh is obtained by applying the following expression. The size of the vector solution ($\boldsymbol{q}_{int}$) is $N_{cells} \times 1$

$$\boldsymbol{q}_{int} = \mathbf{U}^{(1)}\mathbf{S}_{1}\left(\boldsymbol{i}\mathbf{U}^{(2)} \otimes \boldsymbol{i}\mathbf{U}^{(3)}\right)^{\boldsymbol{T}} \quad \text{Eq. (6)}$$

Kronecker Product

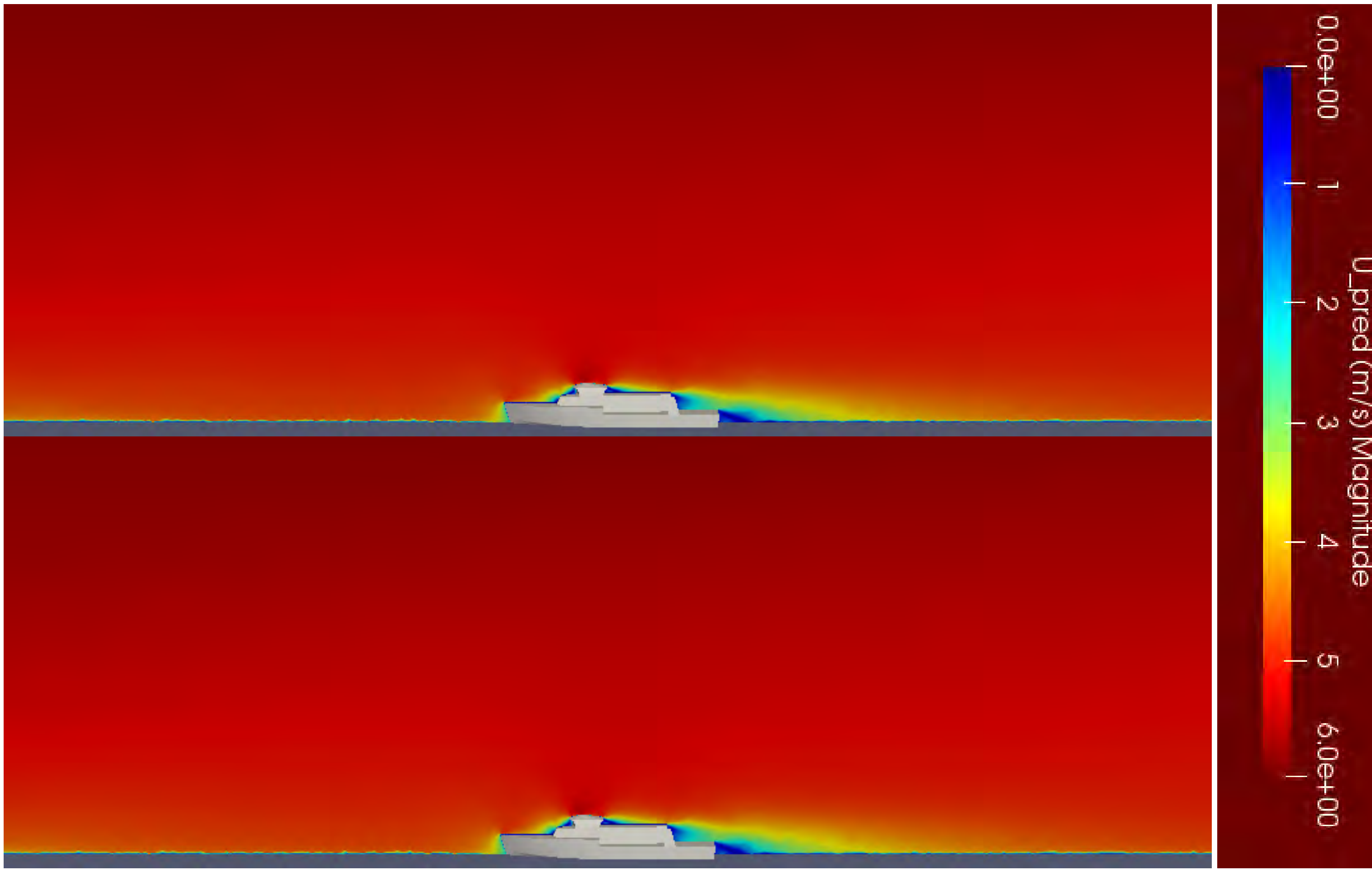
Unfolding matrix of $\boldsymbol{S}$ associated with 1st mode

Results: CFD vs Predictive Tool

- Case: $\boldsymbol{u_{ref}} = \mathbf{9}\ \boldsymbol{m/s}$ and $\boldsymbol{\alpha} = \mathbf{15}°$ (not contained in BBDD)

$\boldsymbol{u_{pred}}\ \left(\frac{\boldsymbol{m}}{\boldsymbol{s}}\right)$

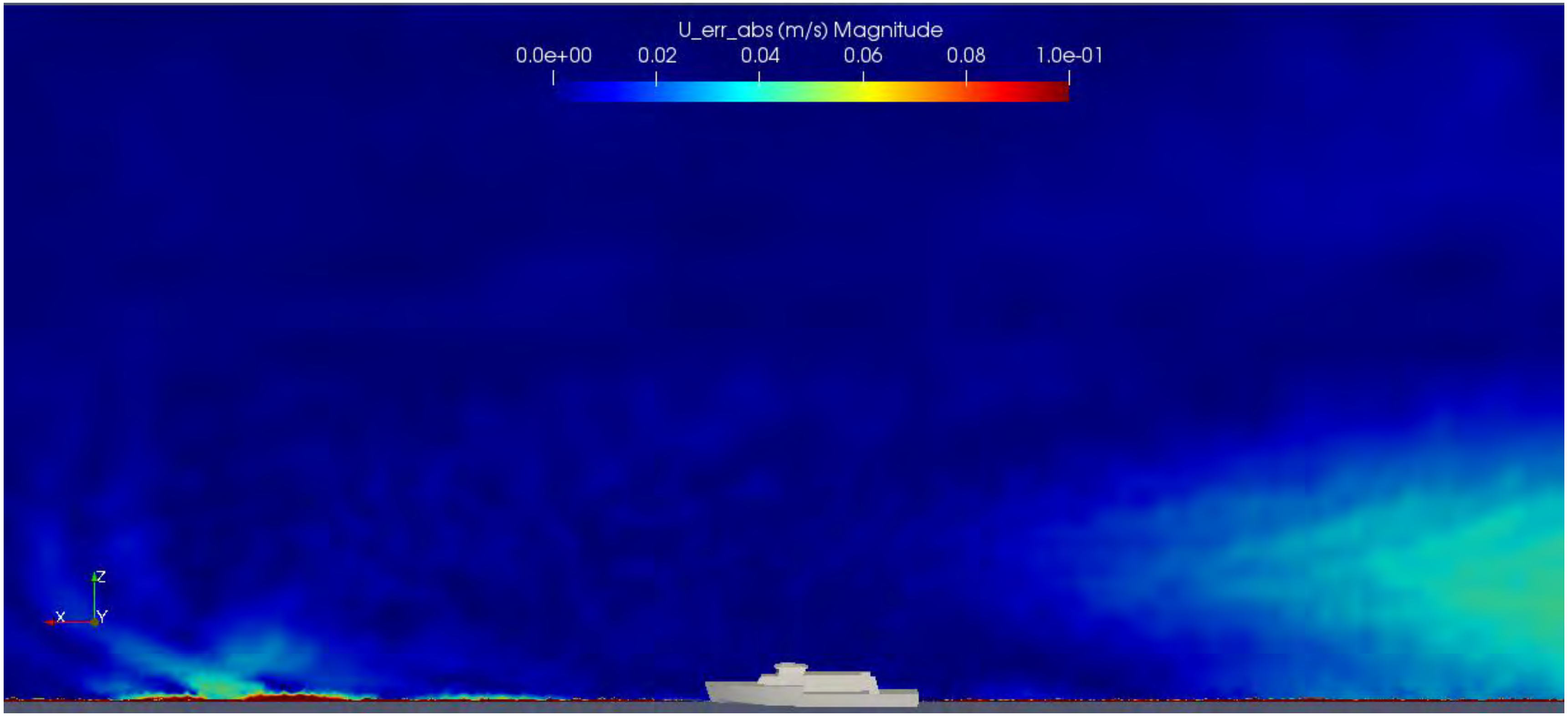
$\boldsymbol{u_{cfd}}\ \left(\frac{\boldsymbol{m}}{\boldsymbol{s}}\right)$



Results: CFD vs Predictive Tool

- Absolute error is computed for each cell:

$$\mathbf{Abs.\,Error} = \left|u_{cfd} - u_{pred}\right|$$



Absolute error $\left[\frac{\mathbf{m}}{\mathbf{s}}\right]$

Results: CFD vs Predictive Tool

- Residual Root Mean Squared Error (RRMSE) of the HOSVD was calculated as:

$$\textbf{RRMSE} = \sqrt{\sum_{n=1}^{N=3} \frac{\left(\sigma^n_{k_n+1}\right)^2 + \left(\sigma^n_{k_n+2}\right)^2 + \cdots + \left(\sigma^n_{r_n}\right)^2}{(\sigma^n_1)^2 + (\sigma^n_2)^2 + \cdots + \left(\sigma^n_{r_n}\right)^2}} \quad \text{Eq. (7)}$$

Retained Modes (σ) [space, velocity, angle]	[360, 10, 36]	[124, 6, 27]	[34,1,15]	[16,1,10]
Acumulated Energy [%]	100 %	99%	95%	90%
Execution time [s]	5.75	1.226	0.2549	0.1
RRMSE [-]	0	0.0146	0.0619	0.1130

- **Time of CFD Simulation: 2hr** → For **99% A.E.,** the **prediction time** is nearly **4800 times faster than CFD**.

Conclusions

- A fast numerical tool to **predict wind flows past a vessel** has been developed **using HOSVD** (unsupervised ML algorithm).
- The tool uses the **weather forecast** as **inputs** for the **predictions in real time**.
- **Prediction time takes less than 1.5 s** (4800 times faster than CFD)
- The **average** and **maximum relative errors** are **4%** and **7%.**
- These type of tools, which can be **applied to different environments** (e.g. urban vertiports), pave the way for automatic generation of UAVs paths.

Authors

1. Gonzalo Veiga Piñeiro (gonzalo.veiga@uvigo.gal)
2. Pablo Dominguez Estevez
3. Enrique Aldao Pensado
4. Gabriel Fontenla Carrera
5. Fernando Veiga Lopez
6. Higinio Gonzalez Jórge (Full. Prof.) (higiniog@uvigo.gal)
7. Elena Martin Ortega (Full. Prof.) (emortega@uvigo.gal)

References

[1] *https://vardmarine.com/computational-fluid-dynamics/*

[2] *S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annual Review of Fluid Mechanics 52 (1) (2020) 477–508. doi:10.1146/annurev-fluid-010719-060214.*

[3] *H. Hersbach, Sea surface roughness and drag coefficient as functions of neutral wind speed, Journal of Physical Oceanography 41 (1) (2011) 247 – 251. doi:10.1175/2010JPO4567.1*

Acknowledgments

Grant TED2021-129757B-C3 funded by MICIU/AEI/10.13039/501100011033 by "European Union NextGenerationEU/PRTR".

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Financiado por la Unión Europea NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

AGENCIA ESTATAL DE INVESTIGACIÓN

Thank you!

Objectives

- Development of a fast (in 1.5 s) wind prediction tool

- **Inputs**:
    - Weather forecast parameters (MeteoGalicia)
    - Vessel location and its orientation

- **Outputs**:
    - Wind velocity
    - Turbulence properties
    - Extension: 1.5 km around the vessel
    - Resolution: less than 1 m

Hersbach model

$$b_n^\nu = -1,47 + 0,93\ln(R), \qquad R = \frac{z_{ref}}{\alpha_M \nu_\infty}(\kappa u_{\mathrm{ref}})$$

$$b_n^\alpha = 2,65 - 1,44\ln(A) - 0,015(\ln(A))^2, \qquad A = \frac{\alpha_{\mathrm{ch}}}{gz}(\kappa u_{\mathrm{ref}})^2$$