upscale

Upscaling **P**roduct development **S**imulation **C**apabilities exploiting **A**rtificial inte**L**ligence for **E**lectrified vehicles

D2.1 Requirements for aerothermal simulations reduced order model

Authors

Eysteinn Helgason – Volvo Cars

Erik Sällström – Volvo Cars

Jianbo Huang – Engys

Bhanu Prakash – IDIADA

Luca Miretti - CRF

Month Year

09/2019

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 824306.

Project Details

PROJECT TITLE	Upscaling product development simulation capabilities exploiting artificial intelligence for electrified vehicles
PROJECT ACRONYM	Upscale
GRANT AGREEMENT NUMBER	824306
INSTRUMENT	RIA
CALL	LC-GV-2018
STARTING DATE OF THE PROJECT	November, 1ST 2018
PROJECT DURATION	42 Months

The UpScale Project

The UPSCALE (Upscaling Product development Simulation Capabilities exploiting Artificial inteLligence for Electrified vehicles) goal is demonstrating the feasibility of using AI enhanced CAE methods in EV development processes, such as vehicle aerodynamics, battery thermal modelling and crash simulation and leading the deployment of AI tools for other CAE applications. UPSCALE is the first EU-project that has the specific goal to integrate artificial intelligence (AI) methods directly into traditional physics-based Computer Aided Engineering (CAE)-software and –methods. These CAE-tools are currently being used to develop road transportation not only in Europe but worldwide. The current focus of the project is to apply AI-methods to reduce the development time and increase the performance of electric vehicles (EVs) which are required by the automotive industry to reduce global emission levels. High performance computing (HPC) and CAE-software and –methods play a decisive role in vehicle development process. In order to make a significant impact on the development process, the two most HPC intensive CAE-applications have been chosen as use cases for the project: vehicle aero/thermal- and crashmodelling. When considering total automotive HPC usage, approximately 20% is used for aero/thermal simulations and up to 50% of HPC resources are utilized for crash simulations. By improving the effectiveness of these two areas, great increases in efficiency will lead to a 20% reduction of product time to market. Other novel modelling approaches such as reduced order modelling will be coupled to the AI improved CAE-software and -methods to further reduce simulation time and ease the application of optimization tools needed to improve product quality. Through the combined effort of universities, research laboratories, European automotive OEMs, software companies and an AI-SME specialized in machine learning (ML), the UPSCALE project will provide a unique and

effective environment to produce novel AI-based CAE-software solutions to improve European automotive competiveness.

The UpScale Consortium

PARTICIPANT Nº	PARTICIPANT ORGANISATION NAME	COUNTRY
1 (Coordinator)	**IDIADA AUTOMOTIVE TECHNOLOGY SA (IDIADA),**	Spain
2	VOLVO PERSONVAGNAR AB (Volvo Cars)	Sweden
3	VOLKSWAGEN AG (VW)	Germany
4	CENTRO RICERCHE FIAT SCPA (CRF)	Italy
5	ESI GROUP (ESI GROUP)	France
6	ENGYS LTD (ENGYS LTD)	United Kingdom
7	Kompetenzzentrum - Das Virtuelle Fahrzeug, Forschungsgesellschaft mbH (VIF)	Austria
8	VRIJE UNIVERSITEIT BRUSSEL (VUB)	Belgium
9	ECOLE NATIONALE SUPERIEURE D'ARTS ET METIERS (ENSAM PARISTECH)	France
10	ALGORITHMICA TECHNOLOGIES GMBH (ALGORITHMICA)	Germany
11	F INICIATIVAS I MAS D MAS I SL (F-INICIATIVAS)	Spain

Document Details

DELIVERABLE TYPE	Report
DELIVERABLE Nº	2.1
DELIVERABLE TITLE	Requirements for aerothermal simulations reduced order model
NAME OF LEAD PARTNERS FOR THIS DELIVERABLE	Volvo Cars
VERSION	1
CONTRACTUAL DELIVERY DATE	M12
ACTUAL DELIVERY DATE	M12
DISSEMINATION LEVEL	Public

Revision History

The following table describes the main changes done in the document since it was created

REVISION	DATE	DESCRIPTION	AUTHOR (ORGANIZATION)
V.0	06/09/2019	First complete version of deliverable	Eysteinn Helgason (Volvo Cars)
V.1	24/10/2019	WP2 leader review	Luca Miretti (CRF)
V.2	31/10/2019	Project coordinator	Enric Aramburu (IDIADA)
V.3	31/10/2019	Review	A. R. de Liébana (IDIADA)

Disclaimer

Table of contents

1. Executive Summary

This report describes the work performed in Task 2.1, including subtasks 2.1.1, 2.1.2 and 2.1.3. The results will be used as a basis for future work in work package 2 and 4. This report is a part of work package 2, focusing on AI based design for Aerodynamics.

The main goal of Task 2.1 is to identify the required inputs in terms of variables (geometries, flow fields) and the amount of data that needs to be generated in order to train Reduced Order Models/Machine Learning Models for aerothermal purposes. Acceptable accuracy thresholds are also discussed.

Part of the report focuses on assessment of data/processing requirements. This is done by evaluating which physical fields and geometrical information are required to evaluate the different designs and perform the learning process during the AI phase. This part also includes evaluation of the amount of data needed to be stored and a proposal on how to limit the amount of simulations required in order to guarantee the best trade-off in terms of accuracy versus cost.
Target settings are also defined for the accuracy of the ROM/ML models based on work package 1 as well as on previous work.

The report concludes with a description of the model used as a simplified vehicle in Work Package 2. A modified version of the DrivAer model has been created for this purpose. The original DrivAer model is a simplified vehicle model well known in the automotive industry. It is intended as a bridge between a bluff body (e.g. the Ahmed body) and a fully detailed vehicle. The original DrivAer model includes a simple internal combustion engine. No electrical version of the DrivAer model is publicly available. For the purpose of the work performed in Work Package 2, a parametrized electrical version of the original DrivAer model has been created. The model is presented in the last section of this report.

This deliverable doesn’t deviate from the plan in regard to its content or delivery date.

2. Determination of data/processing and accuracy requirements for reduced order models

In order to determine the data and processing requirements we focus first on the grid resolution. In order to estimate the size of the grid required for a vehicle, a machine-learning model is trained on a 2 dimensional aerofoil. The information obtained from the simulations is used as a basis to estimate the resolution required as well as the storage requirements when applying the same method to a full size vehicle. The next sub-section focuses on the processing requirements. The computational power required for each simulation and the time required to screen through all the 17 parameters on the simplified vehicle that will be used in work package 2 are estimated. Finally, a screening workflow is proposed in order to minimize the amount of simulations required by identifying the most important parameters.

2.1. Input data specification in training a deep neural network

This sub-section describes the application of a deep neural network to a two dimensional aerofoil case. The results obtained from the simulations are used as a basis when estimating the amount of data required for a complete vehicle simulation.

2.1.1. General considerations

To train a good machine-learning model, one of the first steps one has to take is to decide the grid resolution used for generating a training dataset. A few factors that need to be considered are:

1. The grid resolution needs to be high enough to extract all the important features from a Computational Fluid Dynamics (CFD) simulation flow field
2. The grid resolution needs to be high enough to provide all the necessary training features for the machine-learning model

.

From machine-learning perspective, the smaller the dataset, the easier the training process will be. However, if the resolution of a data sample is too small, it will inevitably miss some important features for the models to capture during training. On the other hand, if the resolution of the data samples is too big, one may not even be able to train a model at all.

In this sub-section, we will try to present some rough analysis as for what kind of resolution one may need to train a model for predicting external aerodynamics behaviour around cars.

2.1.2. An aerofoil case: CFD simulation

We consider an aerofoil with shape code ERRER 557. It is a two-dimensional case, and the CFD simulation is run using simpleFoam, which is a segregated finite volume method solver in OpenFOAM that applies the simple pressure correction. We are using unstructured grids with refinement around the aerofoil surfaces. The total number of cells used in the simulation is 41939. The mesh grid is shown in **¡Error! No se encuentra el origen de la referencia.** and a closer-look of the mesh around the aerofoil surface is shown in **¡Error! No se encuentra el origen de la referencia.**.

The bounding box for the computational domain (in meters) is:

$x \in [-5,5]$ and $y \in [-5,5]$, i.e. the flow domain is 10 meters by 10 meters in x and y directions. The simulation is run for 500 iterations with residuals for U, V and P at 0.00043, 0.00043 and 0.0040 respectively. The aerofoil is about 1 meter in length.

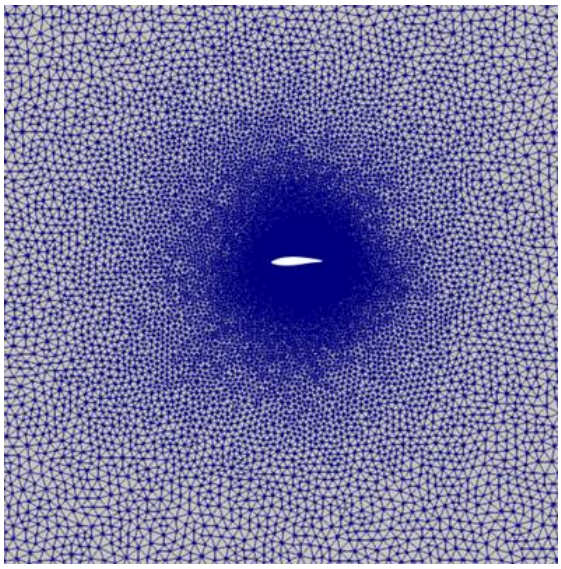

Figure 1. Computational mesh

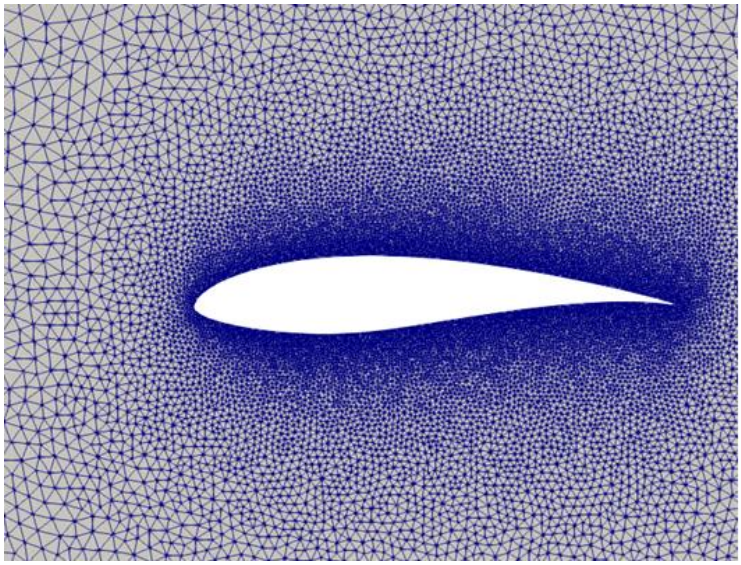

Figure 2. Computational mesh focusing on the area around the airfoil

2.1.3. Sampling results

The dimensions of sampling region in x and y directions are both from -0.8 m to 1.5 m, where $x = 0$ m corresponds to the leftmost part of the aerofoil surface.

To compare the sampling accuracy, three set of sampling resolutions are used: 128x128, 256x256 and 512x512 in x and y directions respectively. The results from the CFD simulations are presented in Figure 1. The results obtained by sampling the pressure to a grid size of 128x128 are presented in Figure 2. The results for a grid size of 256x256 and 512x512 are shown in Figure 3 and Figure 4.

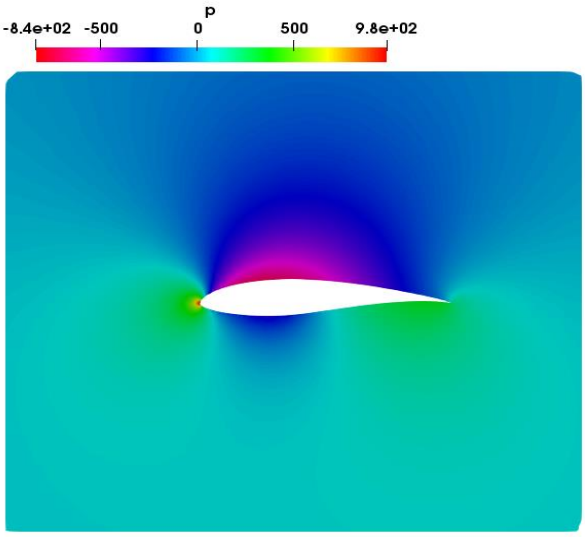

Figure 1. Pressure field obtained from the CFD simulation

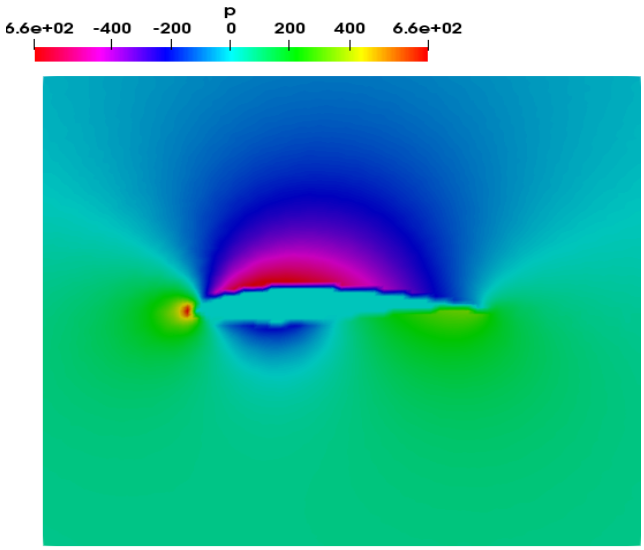

Figure 2. The pressure obtained by sampling with a grid size of 128x128

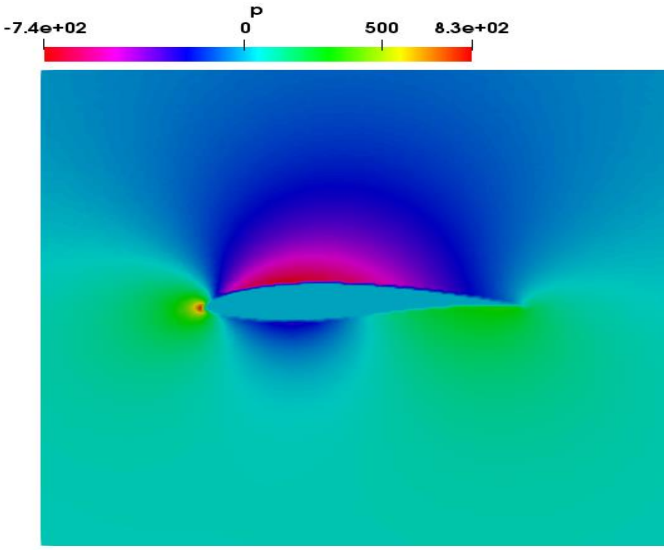

Figure 3. The pressure obtained by sampling with a grid size of 256x256

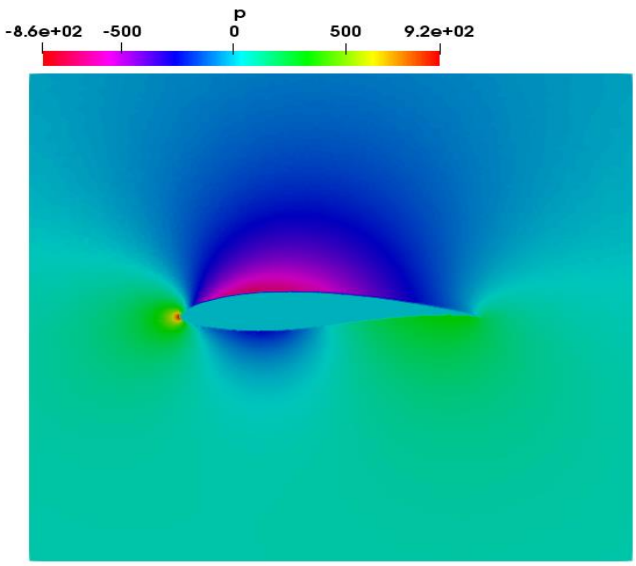

Figure 4. The pressure field obtained by sampling with a grid size of 512x512

The velocity field obtained from the CFD simulation is presented in Figure 5 and the sampled velocity field for a grid size of 128x128 is shown in Figure 6. The sampled velocity field for the remaining grid sizes, 256x256 and 512x512, are presented in Figure 7 and Figure 8, respectively.

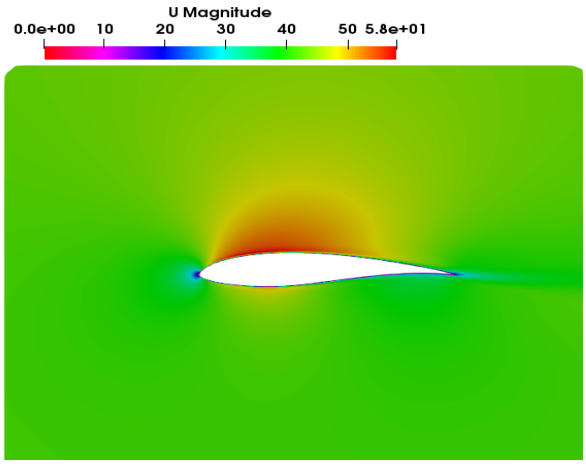

Figure 5. The velocity field obtained from the CFD simulation

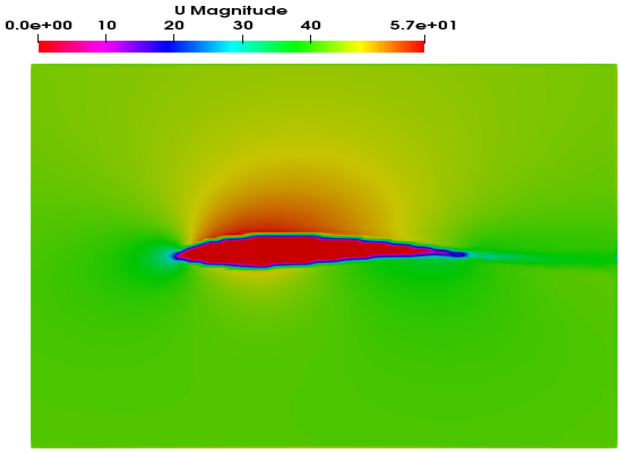

Figure 6. The velocity obtained by sampling with a grid size of 128x128

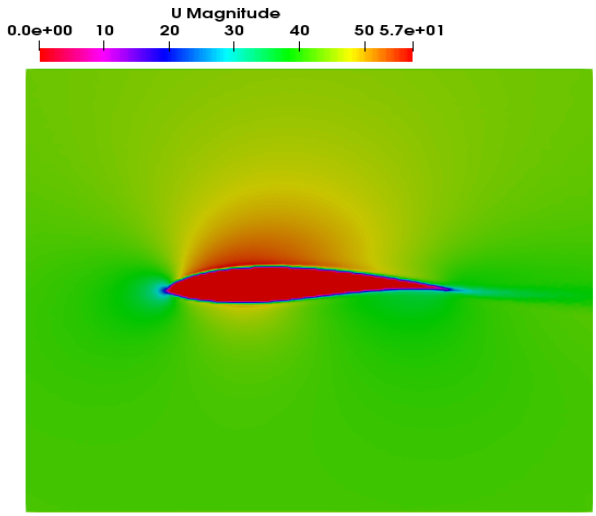

Figure 7. The velocity obtained by sampling with a grid size of 256x256

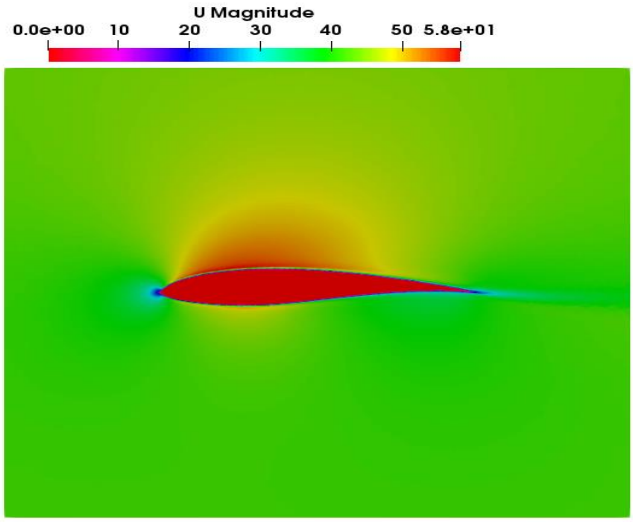

Figure 8. The velocity obtained by sampling with a grid size of 512x512

2.1.4. Analysis of the sampling results

We can draw some conclusions by comparing the sampling results with the output from the CFD simulation:

- The 128x128 resolution is not good enough to smoothly represent the aerofoil surface. It does not capture well the peak velocity, U_{max}, the peak pressure, P_{max}, or the minimum pressure, P_{min} in the flow. The sampled U_{max} is only 86% of the CFD post-processing, and the P_{min} value reqion is far bigger than in the simulation.

- The 256x256 resolution presents a much better result. It resolves the aerofoil surface well, the U_{max} is 96% of the CFD post-processing, and the P_{min} pressure region well-match the CFD post-processing.

- By applying the 512x512 grid size the we expect to slightly improved sampling comparing to 256x256. In practice, it might not be suitable for training the network, as it will greatly increase the training time and memory footprint.

From the above analysis, we can conclude that for the two-dimensional aerofoil case, a sampling grid resolution of 256x256 would be needed to have an adequate representation of the CFD simulations.

2.1.5. Machine-learning prediction

A convolutional neural network (ConvNet) based deep-learning model has been developed at Engys Ltd. The model architecture is similar to that published in [1].

The training dataset is generated using the UIUC database[1] as geometry. A steady-state, incompressible flow solver is used. The upstream, down stream top and bottom boundaries are all set into freestream, with U component chosen randomly, using a uniform distribution (0,100) m/s, while the V and W components are all set into zero.

[1] https://m-selig.ae.illinois.edu/ads/coord_database.html

A two-dimensional model is used. As the training of the ConvNet model requires structured grid data, while the CFD simulation is carried out with unstructured meshes, a k-nearest-neighbour-based sampling method is developed at Engys, and is used to generate the training dataset. The method is fast and generally applicable to any external flow field.

The ConvNet is a U-shaped network with 6 forward layers and 6 backwards layers. The overall accuracy with our dataset is 99.57% for velocity and 89% for pressure, both are much higher than the published accuracy in [1], which is 97.4% for velocity and 85.24% for pressure. The higher prediction error in the pressure field is due to the reason that in most of the flow field, pressure is near zero, so small prediction error will amount to significant relative error. We have tried some different ways, such as to increase the number of hidden nodes, increase the number of training epochs, use different loss criteria etc, none of these seem to further increase the accuracy. The sampling grid resolution of 128x128 is used to obtain the training and testing/validation dataset. The total training samples are 1339, and the test samples are around 149. To ensure that the model predictability is true, the testing/validation cases are not in the training loop.

Figure 9 and Figure 10 show the model-predicted pressure field and the 'target' (ground-truth) pressure field. As one can see, the predicted results are very close to the target ones.

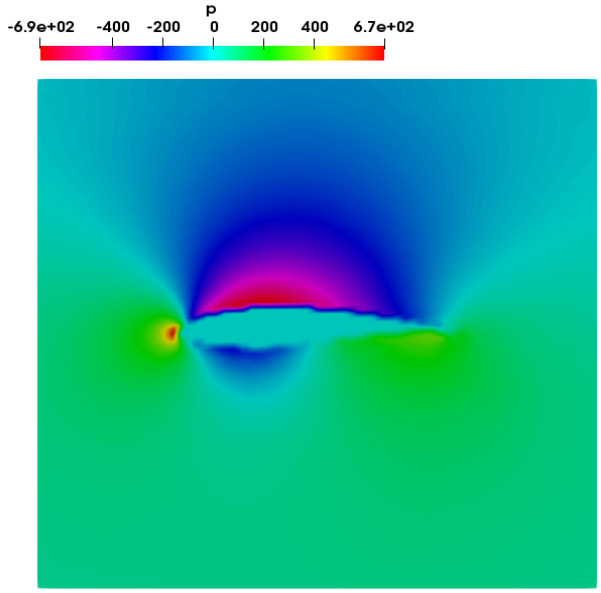

Figure 9. Pressure field predicted by Machine-learning model

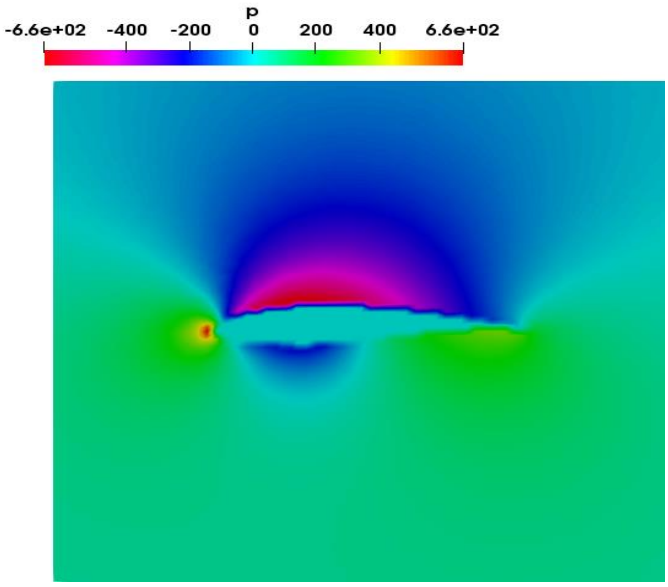

Figure 10. Sampled (target) pressure field

Figure 11 and Figure 12 compare the predicted and ground-truth results for velocity around the airfoil. Again, the two fields are very close to each other.

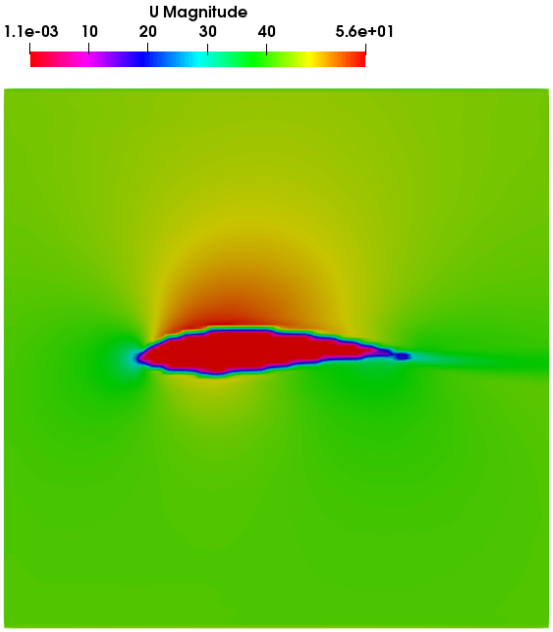

Figure 11. Velocity field predicted by the maachine-learning model

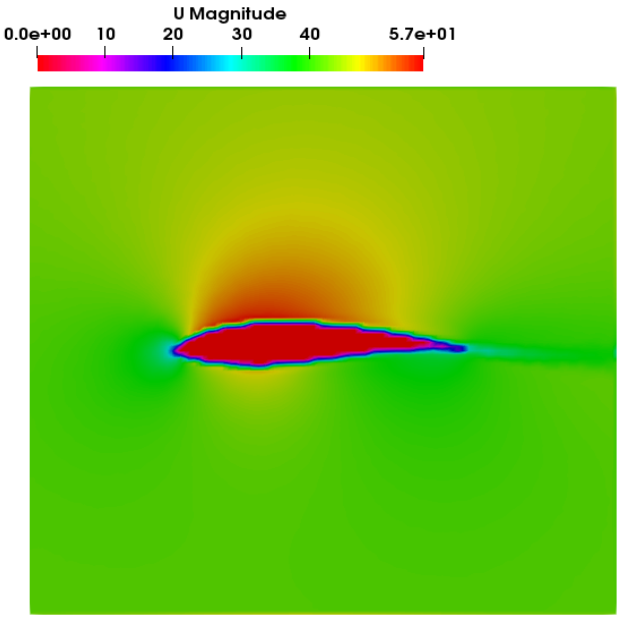

Figure 12. Sampled velocity field

Figure 13 shows the normalized difference between the predicted and ground-truth pressure fields. The difference is normalized by U_0*U_0, where U_0 is the free stream velocity. **Figure 14** shows the normalized difference between the predicted and ground-truth velocity field. The normalization factor is U_0. From these two figures, one can see that the largest difference in the pressure and velocity fields is located in a small region near the airfoil surface. In most of the computational field, the difference between prediction and target values are very small.

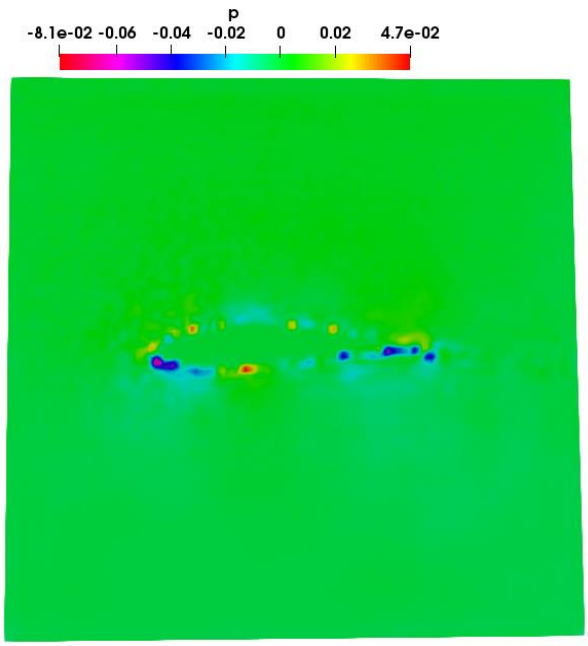

Figure 13. Difference between maching-learning model prediction and the ground-truth pressure field.

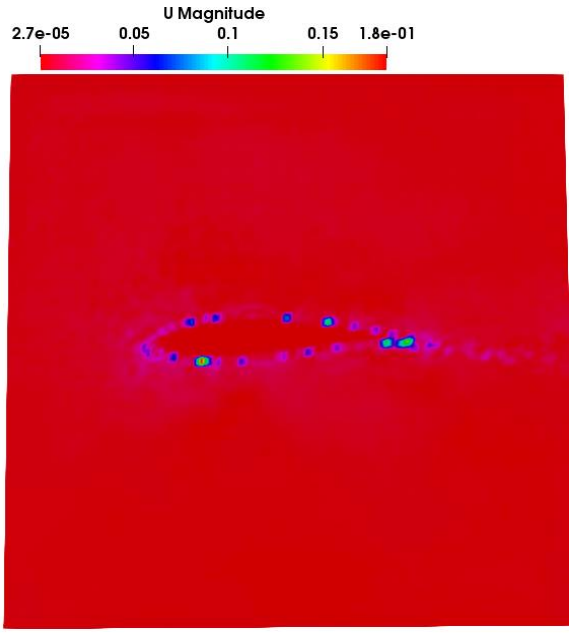

Figure 14. Difference between machine-learning model prediction and the ground-truth velocity field.

2.2. Output fields and storage requirements for automotive simulations

This subsection is divided into two parts. The first focusing on general machine learning methods and the second part on hybrid physics informed machine learning

2.2.1. Output and storage requirements for machine learning approach

One of the aims with Upscale is to predict aerodynamic quantities from a universal parametrization. Therefore, data from different simulations on different geometries should be saved in a common format. It is proposed here to resample data to a uniform grid. Mapping will be done to a few differently sized grids in order to allow for collecting the data for both high resolution analysis as well as simulations with quick turnaround time. Based on the results from the two dimensional aerofoil, the proposal is to sample a three dimensional case to a grid sized 512x256x128, 256x128x64 and 128x64x32 cells. The car should be scaled so that the front wheel center x coordinate is 140, the rear wheel x coordinate is 300. The following quantities should be saved on the uniform grid:

- Flow velocity: u, v, w (m/s)
- Pressure: p (Pa)
- Turbulence eddy viscosity: ν_t $(\mathrm{Pa} \cdot \mathrm{s})$
- Signed distance: d (m)

The grid of size 512x256x128 contains a total of $16 \cdot 1024^2$ cells. There are 6 field quantities. Given that the data is stored as 32 bit (4 byte) single precision floating point values, 384 MB of data needs to be stored from each simulation. This is a manageable size even if the data is kept in a database that grows to thousands number of simulations.
The following scalar quantities should be saved:

- Scale factor: f (m), which is the length of a uniform grid cell in meters.
- The location in physical space of the origin of the uniform grid, i.e. the corner at ground level in front of the car to the left. x_0, y_0, z_0 (m)
- Free stream velocity: v_∞ (m/s)
- Yaw angle: ψ (°)
- Mass flow through heat exchanger(s) ($\dot{m}$)
- Projected frontal area: A (m^2)
- Non-dimensional aerodynamic forces and moments: C_D, C_L, C_S, C_{PM}, C_{RM}, C_{YM} defined according to SAE J1594_201007.[2]

The aerodynamic forces acting on the body of a vehicle are and are shown in Figure 15. Side view of the Drivaer model inside the proposed domain is shown in Figure 16 and the side view in Figure 17.

[2] https://saemobilus.sae.org/content/J1594_201007/

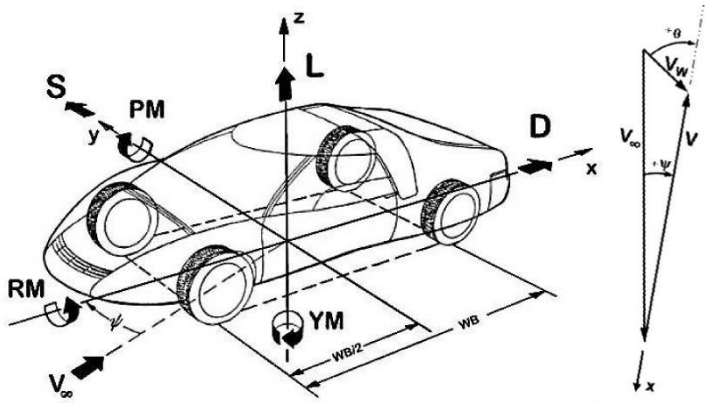

Figure 15. Coordinate system and definition of forces and moments and yaw angle according to SAE J1594_201007.

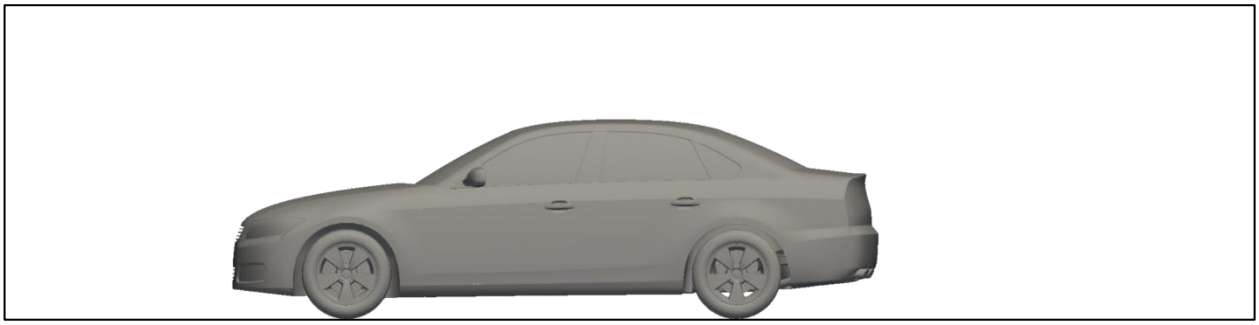

Figure 16. Side view of a Drivaer model in a 512x256x128 volume with front wheels at x=140 and rear wheels at x=300.

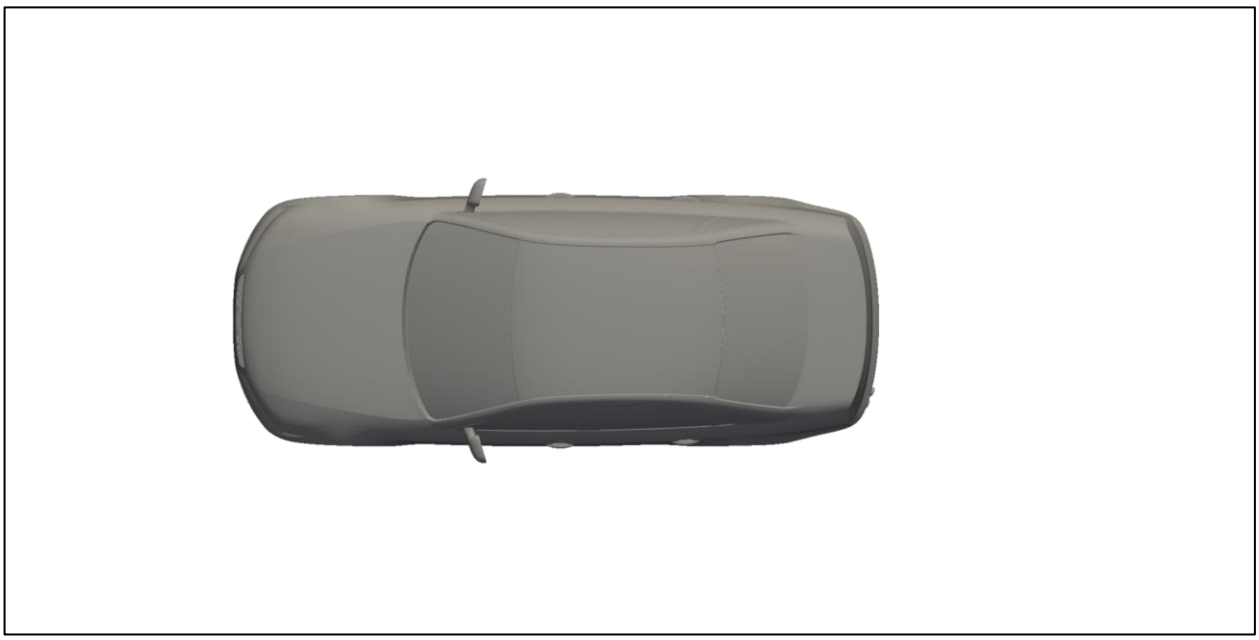

Figure 17. Top view of a Drivaer model in a 512x256x128 volume with front wheels at x=140 and rear wheels at x=300.

The data will be stored in the HDF5[3] format. An HDF5 file is structured similar to a file system. It is advised to use the shuffle feature and either the lzf (for fast saving) or gzip (for smaller files) compression for the floating point field data sets to reduce the file sizes and speed up

[3] https://support.hdfgroup.org/HDF5/

loading.[4] Metadata relating to the specific case can also be stored in the HDF5 files, such as a name or description of the simulation and values of variables used to generate the geometry.

2.2.2. Outputs and requirements for Hybrid PIML approach

The Physics Informed Machine Learning (PIML) approach requires additional processing of the flow field quantities mentioned in 2.2.1 and few others as described below. This approach differs in that it aims to improve the accuracy of conventional RANS simulations, therefore it will not resample data to a uniform grid. On the contrary, in this approach every cell in the RANS grid, can be a sample for the training. This practically means that we do not need hundreds of simulations to get enough training samples. A lot fewer simulations are needed, but each one will likely have a bigger grid than the uniform grid mentioned in the previous approach and at each grid point we will need to store more quantities.
The bigger grid is due to the fact that RANS (2D/3D) simulation grid of automotive CFD simulations is used directly in the PIML model training phase and also the Direct Numerical Solution (2D) / Large Eddy Simulation (3D) solutions are interpolated to RANS grid. The 2D conditions are to be used only to establish the Hybrid PIML workflow and hence the requirements are specified in the next sections mainly for 3D geometries. The detailed discussion related to the number of training samples and corresponding data storage requirements for the Hybrid PIML approach is done at the end of this sub-section (2.2.2). Data storage requirements will certainly fit within the resources available in a typical automotive industry.

It should be noted that as sub task 1.1.3 in work package 1 evolves, the parametric set could be reduced/increased. Hence, the flow field variables data like pressure, velocity etc. along with the inputs and responses parametric set necessary for PIML model need to be stored. [11][12]

Output Fields:

Input Parameter (q_β)	Description	Feature ($\hat{q}_\beta$)	Normalization factor($\tilde{q}_\beta$)
q_1	Ratio of excess rotation rate to strain rate	½ (||Ω||2 - ||S||2)	||S||2
q_2	Turbulence Intensity	K	$\frac{1}{2}U_iU_i$
q_3	Wall-distance based Reynolds number	$\min\left(\frac{\sqrt{k}d}{50\nu}, 2\right)$	-NA-
q_4	Pressure gradient along streamline	$U_k\frac{\partial P}{\partial x_k}$	$\sqrt{\frac{\partial P}{\partial xj}\frac{\partial P}{\partial x_j}U_iU_i}$
q_5	Ratio of turbulent time scale to mean strain time scale	$\frac{k}{\varepsilon}$	$\frac{1}{\|\|S\|\|}$
q_6	Ratio of pressure normal stresses to shear stresses	$\sqrt{\frac{\partial P}{\partial x_i}\frac{\partial P}{\partial x_i}}$	$\frac{1}{2}\rho\frac{\partial(U_k)^2}{\partial x_k}$

[4] http://docs.h5py.org/en/stable/high/dataset.html

q_7	Non-orthogonality between velocity and its gradient	$\left\lvert U_i U_i \frac{\partial U_i}{\partial x_j} \right\rvert$	$\sqrt{U_l U_l U_i \frac{\partial U_i}{\partial x_j} U_k \frac{\partial U_k}{\partial x_j}}$
q_8	Ratio of convection to production of turbulent kinetic energy	$U_i \frac{dk}{dx_i}$	$\lvert\overline{u_j^{'}\ u_k^{'}\ S_{jk}}$
q_9	Ratio of total to normal Reynolds Stresses	$\left\lVert \overline{u_i' u_j'} \right\rVert$	k
q_{10}	Streamline Curvature	$\frac{D\tau}{Ds}$ where $\tau = \frac{U}{\lvert U \rvert}$ $Ds = \lvert U \rvert Dt$	$\frac{1}{L_C}$

Where
U_i = Mean velocity
S = strain rate tensor
u_i' = Fluctuation velocity
ε = turbulence dissipation rate
Ω = rotation rate tensor
v = fluid viscosity
d = distance to wall
D = material derivative
L_c= Characteristic length scale

Finally, input parameter is defined as $q_\beta = \hat{q}_\beta \big/ (\hat{q}_\beta + \tilde{q}_\beta)$

β = 1, 2, …10 except β = 3
Repeated indices imply summation for indices i, j, k, and l but not for β

The input parameters are computed using the RANS simulation data sets whereas the Reynolds stress tensor invariant based components are post-processed from both RANS and DNS (or any considered true solution) simulations. As sub-task 1.1.3 evolves and the Hybrid PIML work flow is established, the true solution is to be substituted with industry computable LES simulation solutions. So, from one RANS simulation, a total of 10 input parameters and 2 responses are to be computed and stored whereas from the true solution, only 2 responses are required. However, the responses from the true solution are stored as deltas (calculated from the difference between the RANS solution and the true solution responses). This results in a dataset of 14 parameters including 10 inputs, 2 responses from RANS and 2 deltas corresponding to the RANS and true solution response differences. Data is to be stored as 32 bit/4 byte single precision floating point values.

The approach is to map inputs with the deltas for training cases using machine learning (ML) algorithms. Later for the new test case, the inputs and responses are to be provided from RANS simulation for computing corresponding deltas by the ML algorithm, which are then added to the test case responses of RANS solution, for correcting them.

The two responses correspond to the RANS models ill-predicted Reynolds Stress anisotropy tensor. The objective of machine learning model design in the PIML approach is to correct the responses from the RANS simulation using an anisotropic tensor computed in true solution,

i.e. DNS, (while establishing proof of concept)/LES (while dealing with full scale car simulations). The Reynolds stress tensor is projected to Galilean invariant quantities representing the shape of anisotropic tensor (ξ , η)

$$\tau = 2k\left(\frac{1}{3}I + A\right) = 2k\,(\frac{1}{3}I + V\Lambda V^T)$$

Where,
τ = Reynolds Stress Tensor
I =Identity matrix
A = anisotropic tensor
k = turbulent kinetic energy or isotropic component
V = [v1 v2 v3], corresponds to Eigenvectors of A
Λ = diag[λ1 λ2 λ3], Eigenvalues of A

With eigenvalues and eigenvectors computed, the invariants of anisotropic tensor A can be obtained as below.

First invariant of A	$I_1 = \lambda_1 + \lambda_2 + \lambda_3$
Second invariant	$I_2 = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1$
Third invariant	$I_3 = \lambda_1\lambda_2\,\lambda_3$

Then, the second and third invariants are used finally to compute the responses (ξ, η) necessary for the ML model.

$$\xi = \sqrt[3]{\frac{I_3}{2}}$$

$$\eta = \sqrt{-\frac{I_2}{3}}$$

Data Storage Requirements:
The normalized values of all the 10 input parameters are in the range [-1, 1], except for q3 which lies in [0, 2] and hence the parameters are of 32 bit (4 bytes) single precision floating point type.

The grid size requirements for this approach are categorized as per the standard CFD three dimensional (3D) simulation grids applicable to automotive industry geometries and flow Reynolds numbers. The final objective of PIML model is to be able to predict 3D turbulent flows, considering geometries relevant to electric automotive vehicles. The Ahmed body and DrivAer models (2) available in literature are to be used for training PIML model. The latter also complements the geometry and the simulations performed in the alternative ML approach that is briefly described in section 2.2.1.

The grid size for a conventional 3D full scale RANS simulation of the flow around a car is approximately 60 million cells (for Reynolds number of O (10^7). A total of 12 floating point variables (flow fields) are needed for calculating the PIML model's inputs and responses. This results in approximately 2750 MB of data. Then 12 floating point variables for inputs and responses require another 2750 MB, totalling 5500 MB of data. However, including some simplifications of under-hood appropriate for electric vehicles, along with other simplifications arising from the PIML model objectives being concentrated in external aero-thermal analysis

has led us to consider DrivAer model accounting for nearly 20 million cells(one third of full scale combustion based car statistics above). Hence the storage requirement for one single 3D RANS simulation is 1833 MB (one third of 5500 MB). For the true solution, LES simulation will be performed instead of DNS as discussed in Section 2.5.2. The Reynolds stress symmetric tensor (6 components) along with 2 responses is to be interpolated to the RANS grid and stored. For the case of 20 million cells, considered above, this requires 610 MB data storage. Then, for one 3D PIML training sample (including both RANS and LES), a total of 2440 MB is required.

For the 3D PIML model, the number of training samples is not linked to the number of simulations, since each cell can theoretically be used as a training sample. The total number of required simulations will therefore be greatly reduced compared to the ML ROM approaches mentioned above. Although, the exact number of total training simulations can't be objectively presented at this phase of project, it is to be well within the storage space availability of any automotive industry.

As an example, in 2D, we have an initial estimation of 100 training simulations used to train the PIML model for providing proof of concept and establishing workflow. This number will most likely be significantly reduced for the 3D PIML due to significantly higher sample size available in every single training case. Also, as the model development progresses in sub-task 1.1.3, one of the objectives is to adapt the PIML for less computational resource requirements, in its training and execution phases. This will be achieved by working on different strategies, like localizing in certain flow regions, reducing the parametric space dimension, and increasing/reducing the PIML model input parameters and responses. This will be discussed in the corresponding work package 1 reports, related to sub-task 1.1.3.

2.3. Computational cost of high fidelity CFD simulations

We will assume that an ordinary high fidelity CFD simulation on a detailed car takes 24 hours to perform on an industrial simulation cluster in the beginning of the UPSCALE project, and an adjoint simulation 72 hours. The purpose of adjoint simulations will be described in a later section. Simulation speed is assumed to at least double over the duration of the project, so these numbers can be halved. The goal of the project is to achieve a trained aero-thermal model with less than a month of simulation time. That allows us to run 60 high fidelity CFD simulations on a detailed car, or 20 adjoint simulations, or any mix of the two. The number of low fidelity CFD simulations that can be run depends on which simplifications are made. This will be assumed to be the simulation budget for training a model. The numbers may be revised upwards if WP1 delivers significant speed increases.

2.3.1. Cost of covering the parameter space

The framework will initially be developed with the Drivaer model (2)[5] parametrized into 15 continuous and 2 discrete (on/off) variables. The model is simpler than a real car, but the parametrization is realistic. Running a full factorial with these parameters would require a total of $2^{17} (= 131072)$ simulations, which would take 182 years to run. Another alternative is to solve for the coefficients in a model:

$$y = a + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{i-1} c_{ij} x_i x_j$$

Where

y is the quantity to be modeled, for example the aerodynamic drag area, $C_D A$, of a car,
x_i is independent variable (factor) i,
n is the number of variables, and
a, b_i, c_{ij} are the coefficients that need to be fit in the model.

This captures main effects and two factor interactions and requires at least $1 + n + \frac{n(n-1)}{2}$ simulations, which for 17 independent variables becomes 154. This is still many more than the goal, and yet no non-linear effects along any single independent variable is captured.

2.3.2. Reducing complexity

We can reduce the number of simulations in a number of ways:

- Screen variables, and use only the most important ones.
- Split the problem into smaller sub-problems. If the 17 variable problem is split into an 8 and a 9 variable problem, the number of samples needed to fit to the equation for y above is reduced from 154 to 83 (=37+46). Splitting to three problems can reduce the number to 60 and four problems to 49. If we have knowledge of which variables interact, this can be done more effectively.
- Far fewer function evaluations are needed for a good fit if the shape of the problem is known. E.g. if a function is known to be a $\log$ function, that information can be used instead of fitting to a general function such as a polynomial. For the scope of this work, it is assumed that we do not have any such information.

2.3.2.1. Shape sensitivity from adjoint CFD

By solving the adjoint method for CFD, the surface-normalized shape sensitivity $\frac{\partial J}{\partial \beta_j}$ of a cost function J can be obtained, where β_j is the normal node displacement of node j (3). The shape sensitivity to a parameter x_i can in turn be defined as a sum over all nodes affected by changing the parameter:

$$\frac{\partial J}{\partial x_i} = \sum_j \frac{\partial J}{\partial \beta_j} \frac{\partial \dot{\vec{u}}_j}{\partial x_i} \vec{n}_j$$

where $\vec{u}_j$ is the location in space of node j and $\vec{n}_j$ is the outward surface normal at node j.

2.4. Proposed screening workflow

We have a model with n independent continuous variables. We want to minimize a cost function J, for example the aerodynamic drag, $C_D A$, of the model. Scale the variables so that the range of interest is $[-1,1]$.

$$J = f(\vec{x}), J \in R, \vec{x} \in R^n, x_i \in [-1,1]$$

1. Run an adjoint simulation with all independent variables at 0, i.e. evaluate $J(\vec{x} = \mathbf{0})$. Map the adjoint simulation results to the independent variables to get $\frac{\partial J(\mathbf{0})}{\partial x_i}$.
2. For independent variables x_i where the adjoint simulation indicates that $\frac{\partial J(\mathbf{0})}{\partial x_i} < 0$, flip the mapping so that $\frac{\partial J(\mathbf{0})}{\partial x_i} \geq 0 \; \forall \; i \in [1, n]$.

3. Run one CFD adjoint simulation with $\vec{x} = -\mathbf{1}$ and one with $\vec{x} = \mathbf{1}$ and evaluate $\frac{\partial J(-\mathbf{1})}{\partial x_i}$ and $\frac{\partial J(\mathbf{1})}{\partial x_i}$. Again, the adjoint simulation results are mapped to the independent variables.
4. If, at this stage, there is any independent variable i that the aerodynamicist who is trying to optimize the car is free to choose, and $\frac{\partial J(\vec{x})}{\partial x_i} \geq 0 \forall \vec{x} \in \{-\mathbf{1}, \mathbf{0}, \mathbf{1}\}$, then that variable can be set to -1 in all future simulations and dropped from further investigation, regardless of its importance.
5. Now, the contribution to J from variable x_i can be estimated with the equation:
$$\Delta J_i(x_i) = a_i x_i + b_i x_i^2 + c_i x_i^3$$
with the assumption:
$$\frac{\mathrm{d}\Delta J_i(x_i = -1)}{\mathrm{d}x_i} = \frac{\partial J(-\mathbf{1})}{\partial x_i}, \frac{\mathrm{d}\Delta J_i(0)}{\mathrm{d}x_i} = \frac{\partial J(\mathbf{0})}{\partial x_i}, \frac{\mathrm{d}\Delta J_i(1)}{\mathrm{d}x_i} = \frac{\partial J(\mathbf{1})}{\partial x_i}$$
which yields:
$$a_i = \frac{\partial J(\mathbf{0})}{\partial x_i}$$
$$b_i = -\frac{1}{4}\left(\frac{\partial J(-\mathbf{1})}{\partial x_i} - \frac{\partial J(\mathbf{1})}{\partial x_i}\right)$$
$$c_i = \frac{1}{6}\left(\frac{\partial J(-\mathbf{1})}{\partial x_i} - 2\frac{\partial J(\mathbf{0})}{\partial x_i} + \frac{\partial J(\mathbf{1})}{\partial x_i}\right)$$
6. Find the minimum and maximum of $\Delta J_i(x_i)$ in the interval $x_i \in [-1,1]$. The x_i that yields the minimum and maximum ΔJ_i are denoted $x_{i,min}$ and $x_{i,max}$, respectively.
7. Run an ordinary CFD simulation with $\vec{x} = \vec{x}_{min} = [x_{1,min}, \dots, x_{n,min}]^T$ and one with $\vec{x} = \vec{x}_{max} = [x_{1,max}, \dots, x_{n,max}]^T$. The difference in J between the two simulations will be used to estimate the order of magnitude of change of J in the domain under investigation.
8. In the next step is optional but very useful when the number of variables is large. Sequential bifurcation (4) is used to exclude unimportant variables. This is done by splitting the variables in to two groups of as equal size as possible. The screening through sequential bifurcation is accelerated by putting the variables that are expected to be less important in the first group, and the rest in the second group each time the problem is split. The measure of how important variable i is expected to be is:
$$I_i = \max\left(\mathrm{abs}\left(\frac{\partial J(-\mathbf{1})}{\partial x_i}\right), \mathrm{abs}\left(\frac{\partial J(\mathbf{0})}{\partial x_i}\right), \mathrm{abs}\left(\frac{\partial J(\mathbf{1})}{\partial x_i}\right)\right)$$
To determine the importance of a group of variables contained in the set S, run a CFD simulation with $\vec{x} = \vec{x}_S$ such that:
$$x_i = \begin{cases} x_{max} & \text{if } x_i \in S \\ x_{min} & \text{otherwise} \end{cases}$$
The group is important if $J(\vec{x}_S) - J(\vec{x}_{min})$ is larger than a threshold set by the user. If the group is important, it is split again, otherwise it is discarded and the variables will not be

used anymore.

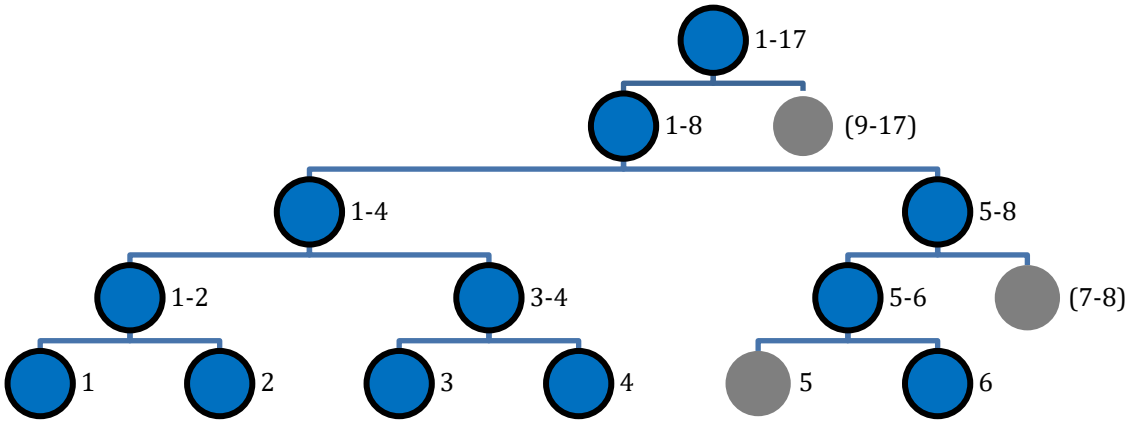

Figure 18. Sequential bifurcation example reducing 17 independent variables to 5 important ones in just 14 simulations. The blue dots with black circles are simulations that end up showing groups of variables are important, and the grey dots circled are simulations showing variables are unimportant.

9. If there are too many important variables remaining after screening, the problem should be split into sub problems. For example splitting variables affecting the front into one group and variables affecting the rear in a different group.

When only the most important variables remain, the remaining variables can be investigated further. The remaining budget for further investigation, assuming 3 adjoint simulations and 14 ordinary CFD simulations during the sequential bifurcation, is 37. An alternative is to skip the sequential bifurcation step and instead keep a desired number of most important variables, as determined by the importance function I_i. Then the remaining budget is 51 CFD simulations which is within the scope of the project when running high fidelity transient simulations.

2.5. Accuracy requirements for the reduced models

This subsection describes the accuracy requirements. Subsection 2.5.1 presents the accuracy requirements for the general machine learning models applied in this work package and subsection 2.5.2 focuses on the accuracy requirement for the hybrid physics informed machine learning.

2.5.1. Machine learning

In the development phase of a vehicle the accuracy requirements of the simulation methods used for external aerodynamics are highly dependent on the context. When evaluating the overall aerodynamic properties of the vehicle the accuracy of the absolute values of the drag and lift forces will be important. While when e.g. focusing on improving the aerodynamic drag on the vehicle, it can be acceptable to know in which direction to move the surface in order to decrease drag. The trends of the observed parameters might therefore give enough information to be able to improve the aerodynamic properties of the vehicle.

Currently available publications on 3 dimensional simulations on a detailed passenger vehicles include results where POD has been applied. The results present qualitative comparison between field variables (5) as well as accuracy comparison where the error when comparing to high fidelity simulations is in the range of around 5% for the velocity and pressure fields and lower than 1% for the drag coefficient. (6) (7) (8) (9). The results presented in section 2.1 and previous work where a convolutional neural network is applied on a 2 dimensional aerofoils (see e.g. (10)) give indication on the accuracy we can expect when applying it on a 3 dimensional geometry. The general assumption here is that the accuracy will be lower when applying the method on a 3 dimensional case. The requirement, to be able to apply it in an optimization process, is that the trends of the global variables, drag and lift, are correctly predicted.

2.5.2. Hybrid physics informed machine learning

There are different approaches to model turbulence in CFD simulations. This includes RANS, DES, LES and DNS in the increasing order of prediction accuracy in comparison with natural flow physics/experiments. Also, the grid requirement increases approximately by 1 - 2 orders of magnitude when model changes from RANS to DES, DES to LES and LES to DNS are involved. The consequence is much higher computing resource requirements.

Although DNS is more accurate, it has been performed only at low Reynolds numbers of O (10^4 - 10^5) and for highly simplified geometries in the literature, even using state-of-the-art supercomputing facilities. However, RANS, even though it is less accurate, is the work horse turbulence modelling approach for industry relevant CFD simulations. For industry level aerothermal simulations, DES and LES are chosen when there is a requirement for better time and space resolved turbulent flow along with other product design relevant parameters like drag coefficients (Cd), heat losses, efficiencies etc. As a part of product development, a typical approach for industrial applications is to choose a few possible final design relevant geometries, from the RANS simulations results, and then simulate using a more accurate DES/LES simulation for better aerothermal analysis.

The final objective of the hybrid PIML approach is to obtain the level of 3D LES models accuracy by correcting the results obtained from the RANS simulation. This is achieved by training the machine learning models using RANS steady state simulation results and considering LES modelled results on corresponding simulation geometries as true solutions. This will be done in an iterative manner, increasing the complexity of the simulated geometries gradually in order to test the limits of this approach and its feasibility.

The work flow involving input parameters and responses, as elaborated in section 2.2.2, is executed considering RANS (less accurate) and DNS (true solution) results for 2D/simplified 3D geometries to validate the PIML model with natural flow physics/experiments. In this process, until the RANS solution is closely similar to corresponding DNS solution, the inputs and responses are extensively iterated within the constraints imposed by physical and mathematical aspects of fluid mechanics theory. The modifications include reducing the number of total parameters used or modifying/replacing chosen parameters in 2.2.2 by including new turbulent flow quantities, based on the accuracy of corrections provided by the PIML model. With the standardization of inputs and responses for this 2D cases, the PIML model will be extended for 3D full scale automotive geometries by further iterating sensitivities of LES modelling into the inputs and responses.

So the objective of the trained Hybrid PIML model in terms of accuracy is to obtain LES level accuracy for an input of RANS simulation result, on slightly simplified electric cars. Typically, RANS simulation for a full scale automotive geometry requires 12-24 hours whereas a LES simulation for the same geometry, and using same number and type of CPU cores, requires around 70-120 days. Considering these averaged statistics, if the computational complexity of the PIML model, during the prediction phase, is less than the difference in computing time between RANS and LES, and it provides LES level accurate result, any reduction in number of days is an advantage for industrial applications. It should be noted that increasing cores or CPU performance will have a different set of statistics, with the ratio/trend being similar.

By targeting a maximum limit of 20% error in solution or PIML output reaching only 80% accuracy in comparison with LES, the increase in run-time for PIML prediction will be reasonable for industry requirements. Assuming a least favourable scenario, PIML model can take up to 5 days and its input from RANS simulation needing 1 day would result in the reduction run-time as compared to 70 - 120 days for computing a LES solution. In this situation, even a 80% accurate LES solution is a positive position for using PIML model in regular industry projects. The most favourable scenario, for using PIML in product development phase, is when the PIML model computes the result in less than a day. The PIML model predictions, when already trained, are computed for the grid sizes equivalent to typical RANS simulations. Hence, the aforementioned scenarios are estimated based on this data feed size (RANS meshes) for the trained PIML model.

To briefly provide the conditions for most and least favourable scenarios, it depends on the computational complexity of machine learning algorithm. The two approaches that are to be explored include decision trees based random forest algorithm whose prediction time is of O(p) and the neural networks approach with a prediction time complexity of order O($pn_{l1} + n_{l1}\,n_{l2} + n_{l2}\,n_{l3} + \cdots.$) depending on the number of neuron layers that results in an accurate solution. Where p is number of features being predicted and n_{li} being number of neurons in the i^{th} layer. So, the random-forests predicts solutions faster as compared to neural networks and the decision factor for choosing one of them for PIML model is the equilibrium point between speed and accuracy.

3. Simplified electric vehicle shape

The publicly available version of the DrivAer model consists of three different interchangeable rear ends of a simplified vehicle. The rear ends correspond to a notchback, fastback and a squareback. The model also includes a simplified internal combustion engine. For the purpose of applying machine learning on a simplified electrical vehicle, a modified version of the DrivAer model has been created. This model includes a simplified battery model located in the floor of the vehicle and the internal combustion engine has been replaced with an electrical engine. The modified version of the DrivAer model is presented in Figure 19, showing the exterior and the underbody.

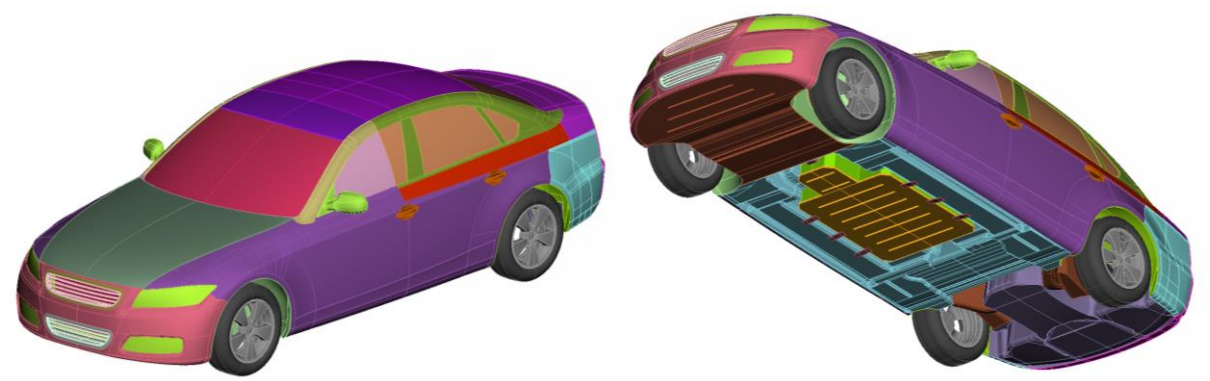

Figure 19. The exterior and the underbody of the electrical version of the DrivAer model. The battery is located in the underfloor.

The layout of the simplified electrical components, the engine and the battery is presented in Figure 20.

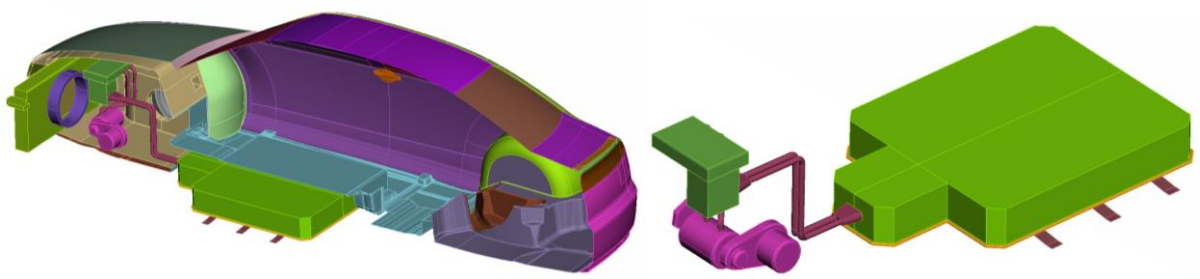

Figure 20. A cut through the center of the car along the x-axis showing the location of the simplified electrical components, including the electrical engine driving the front wheels of the vehicle as well as the battery.

For the purpose of optimizing the vehicle the geometry has been parameterized in the ANSA pre-processor. The parametrization consists of 14 continuous parameters and 3 discrete parameters. The continous parameters focus on the most important design features of the vehicle as well as the location of the battery and ride heigh. The discrete parameters are used for closing and opening the air intake in the front and to add or remove the optional spoiler. The parameters are presented in Table 1.

Table 1. List of parameters applied to the electrical version of the DrivAer model

1. Lower air intake closure (ON/OFF)
2. Upper air intake spoiler (ON/OFF)
3. Lower front grille height

4. Upper front grille height
5. Bonnet inclination
6. Front bumper ramp angle
7. Windscreen inclination
8. Battery pack z-position
9. Rear window inclination
10. Rear window length
11. Trunk lid angle
12. Trunk lid spoiler (ON/OFF)
13. Trunk lid spoiler inclination
14. Rear diffuser angle
15. Trunk length
16. Ride height – ground clearance
17. Ride height – pitch angle

ACKNOWLEDGEMENT

The author(s) would like to thank the partners in the project for their input, valuable comments on previous drafts and for performing the review.

Project partners:

PARTICIPANT Nº	PARTICIPANT ORGANISATION NAME	COUNTRY
1 (Coordinator)	IDIADA AUTOMOTIVE TECHNOLOGY SA (IDIADA),	Spain
2	VOLVO PERSONVAGNAR AB (Volvo Cars)	Sweden
3	VOLKSWAGEN AG (VW)	Germany
4	CENTRO RICERCHE FIAT SCPA (CRF)	Italy
5	ESI GROUP (ESI GROUP)	France
6	ENGYS LTD (ENGYS LTD)	United Kingdom
7	Kompetenzzentrum - Das Virtuelle Fahrzeug, Forschungsgesellschaft mbH (VIF)	Austria
8	VRIJE UNIVERSITEIT BRUSSEL (VUB)	Belgium
9	ECOLE NATIONALE SUPERIEURE D'ARTS ET METIERS (ENSAM PARISTECH)	France
10	ALGORITHMICA TECHNOLOGIES GMBH (ALGORITHMICA)	Germany
11	F INICIATIVAS I MAS D MAS I SL (F-INICIATIVAS)	Spain

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 824306

5. References

1. *Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows.* **N. Thuerey, K. Weißenow, L. Prantl, Xiangyu Hu.** Technical University of Munich. arXiv:1810.08217v2.
2. *Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations.* **A. Heft, T. Indinger, N. Adams.** Detroit, Michigan, USA:, April 23-26, 2012. SAE 2012 World Congress. pp. 2012-01-0168.
3. *Adjoint methods for car aerodynamics.* **Othmer, C.** 1, Journal of Mathematics in Industry, 2014, Vol. 4, p. 6. doi: 10.1186/2190-5983-4-6.
4. *Searching for important factors in simulation models with many factors: Sequential bifurcation.* **Kleijnen, J. P. C., & Bettonvil, B. W. M.** 1, European Journal of Operational Research, 1997, Vol. 96. 180-194.
5. *Application of POD techniques in CFD optimization of vehicle aerodymamics.* **L. Miretti, E. M. Ribaldone,.** 2016.
6. *On Active Subspaces in Car Aerodynamics.* **C. Othmer, T. Lukaczyk,P. G. Constantine, J. J. Alonso.** Washington D.C. AIAA Aviation forum, 2019. 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.
7. *Towards Real-time Vehicle Aerodynamic Design via Multi-fidelity Data-driven Reduced Order Modeling.* **A. Bertram, C. Othmer, R. Zimmermann.** Florida, USA: AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018.
8. *Reduced Order Modeling for Vehicle Aerodynamics via Proper Orthogonal Decomposition.* **M. Mrosek, C. Othmer, R. Radespiel.** SAE International Journal of Passenger Cars: Mechanical Systems - Accepted for publication, 2019-09-21.
9. *Free Form Deformation, mesh morphing and reduced order methods: enablers for efficient aerodynamic shape optimization.* **F. Salmoiraghi, A. Scardigli, H. Telib and G. Rozza.** : International Journal of Computational Fluid Dynamics, 2018-03-15. Vol. 32, pp. 233-247.
10. **Guo, X. and Li, W. and Iorio, F.** Convolutional Neural Networks for Steady Flow Approximation. *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA:, 2016, pp. 481-490.
11. **Wang, Jian-Xun, Jin-Long Wu, and Heng Xiao.** "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data." Physical Review Fluids 2.3 (2017): 034603.
12. **Ling, Julia, and J. Templeton.** "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty." Physics of Fluids 27.8 (2015): 085103.