eFlows4HPC

Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

D4.2 Design of the Pillar I use cases

Version 1.0

Documentation Information

Contract Number	9555558
Project Website	www.eFlows4HPC.eu
Contratual Deadline	31.08.2021
Dissemination Level	PU
Nature	R
Author	Stefan Boschert (SIEMENS)
Contributors	Arianna Bosco (SIEMENS), Sebastian Ares de Parga Regalado (CIMNE), Giovanni Stabile (SISSA), Joaquín A. Hernández (CIMNE), Raúl Bravo (CIMNE), Riccardo Rossi (CIMNE)
Reviewer	Nicolas Barral, Mario Ricchiuto, Heloise Beaugendre (INRIA)
Keywords	Simulation of large air cooled electrical engines, Reduced Order Model, Hyperreduction

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Change Log

Version	Description Change
V0.1	Initial version for review
V0.2	Including review comments
V1.0	Version for submission

Table of Contents

1. Executive Summary

Current deliverable aims at describing the test cases that will be tackled during the project. A number of test cases were selected with the aim of identifying high-impact cases which are at the same time feasible but challenging from the computational point of view. The selected cases were also organized hierarchically in an order of increasing complexity. The focus is on the simulation of large air cooled electrical engines, for which the thermal behaviour becomes crucial.

Although the efficiency of electrical engines has increased significantly over the years, intrinsic electrical losses imply that the thermal behaviour needs to be studied in detail to avoid potentially catastrophic overheating. In many cases, the engines are air cooled, with the rotor acting as a fan and pushing the air through multiple small-size gaps which go through the stator. Such cooling mechanism is optimal in stationary working conditions and minimizes construction difficulties. The downside is however that when the rotor is stopped, or moves slowly, the system's cooling becomes insufficient, thus resulting in a tendency to overheat. Current deliverable aims at the development of a Reduced Order Model able to tackle both steady state and transient conditions while at the same time having a sufficiently reduced online cost to be used in operation.

The deliverable begins with a detailed description of the application, continues with some details about the test cases to be considered and finishes with some details on the numerical techniques to be employed in solving the problem.

2. Problem description &application background

Electrical motors are the key component for all automation applications. They provide a good and efficient means for all kinds of motion generation. In our case we look at medium voltage motors which are extensively used for heavy-duty applications like compressors, mills, pumps, crushers shredders and many more. Due to these versatile application fields the motors are utilized all over the different industries like cement industry, mining, paper industry, metal and steel, chemical, oil and gas industry. The size of such motors is typically ~2.5m length, 1.8m height, with a total mass of ~7t. This gives a rated power of 2-8 MW (Figure 1)

Depending on the application, the motor can be connected directly to the power line, or a converter can be used to control the rotation rate and torque accordingly. In the case of a direct line connection during the start and stop of the motor a large amount of additional heat is generated, as the rotation rate is not in sync with the supply power frequency. This in turn can lead to an overheating of the motor- especially to the coils - and cause permanent damage up to a total failure of the asset. The operation of the motor using an additional converter attenuates this situation at additional costs for the converter. Using a converter the rotation rate of the motor can be adjusted to the needs of the operation. As the cooling is directly linked to the rotation rate, a lower rotation rate goes in sync with lower cooling air flow.

To get rid of the excess heat generated during the operation, a sophisticated cooling concept of the coils was developed. To realize a compact structural shape, the Siemens Simotics H-compact PLUS series motors are designed such that air is guided through slits in the rotor and stator such that it can take up the heat loss from the windings. In this configuration the motor itself works as a pump and drives the air flow due to centrifugal forces (Figure 2).

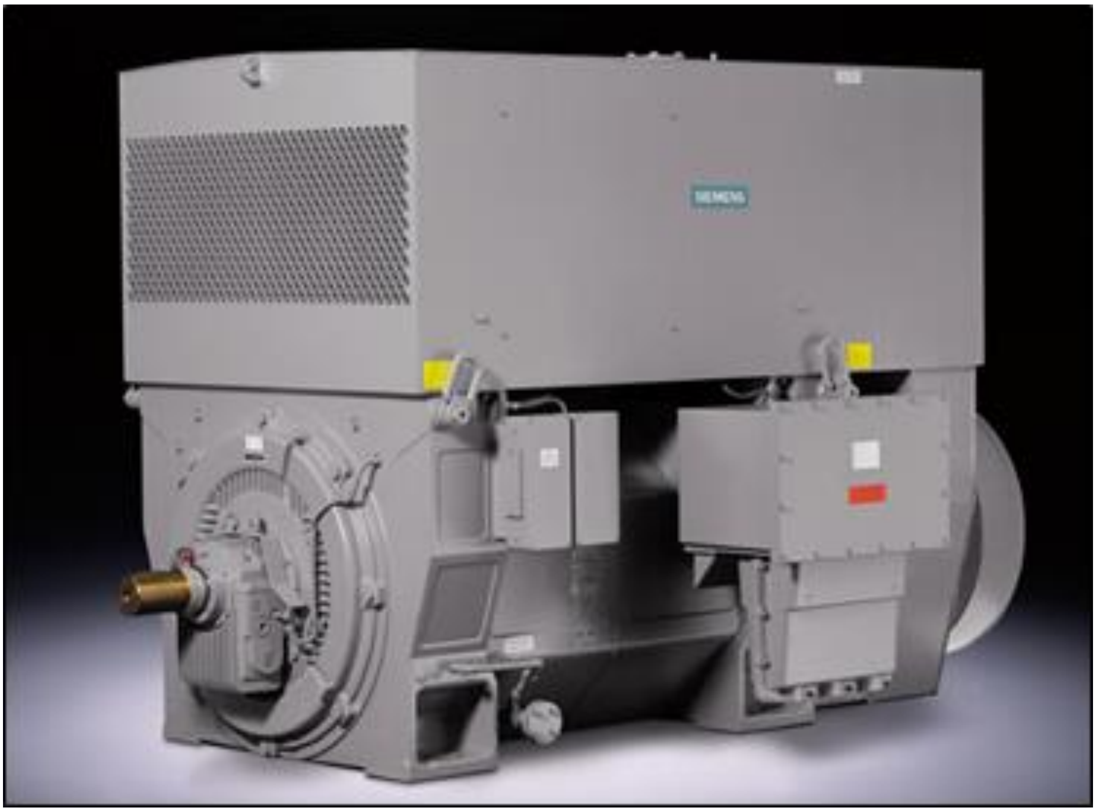


Figure 1. Simotics H compact PLUS motor with water cooler (heat exchanger) mounted on top.

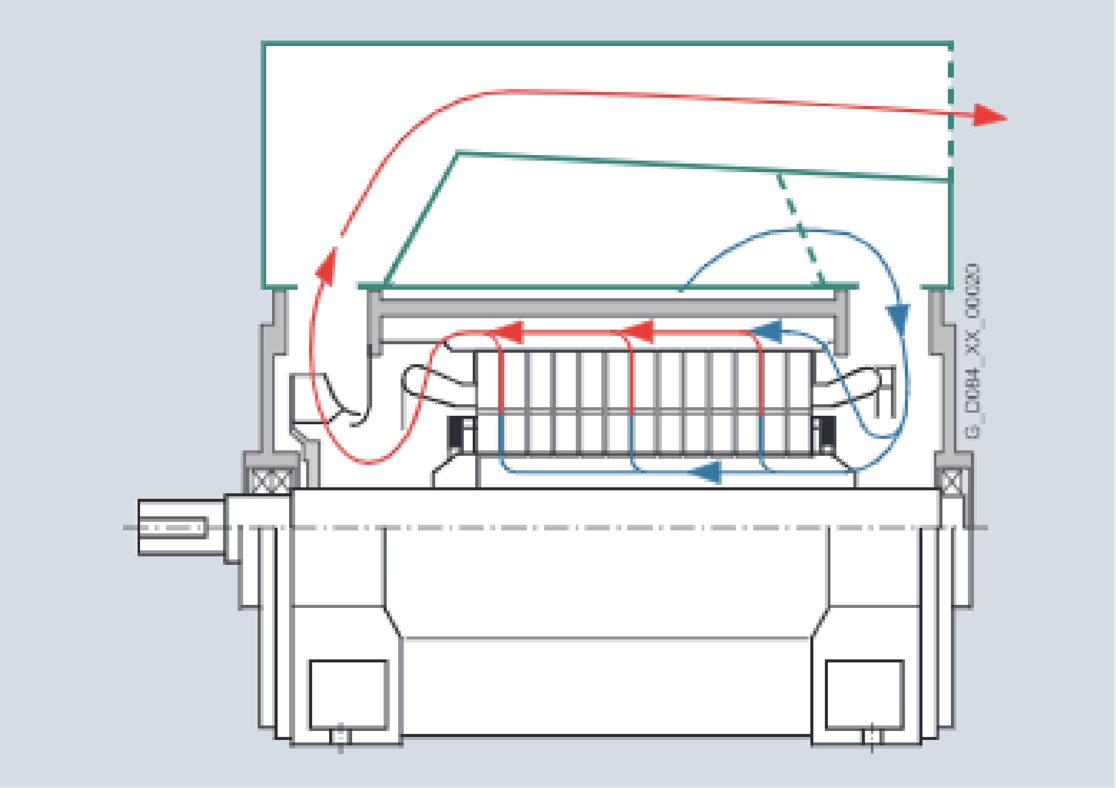


Figure 2. Schematic air flow through Simotics H-compact PLUS motor [Sie17]

For a safe operation of the motor it is necessary that the temperature in the windings does not exceed a critical temperature, as the electrical insulation is damaged due to thermal degeneration. This limits the maximum possible mechanical load of the motor. Even if the motor has an efficiency of ~97%, electrical losses lead to substantial heating as the rated power of the motor with~2000kW gives a total heat load of ~60-80 kW, which is even significantly higher during start up and brake. The electrical losses occur within the motor not only in the copper windings, but also in the iron-parts and rotor cage due to induction.

During operation and predominantly after an emergency shutdown, the motor needs to cool down before it can be started safely again. A fixed time interval is prescribed by the operation guideline, which guarantees safe operation. As during the standstill of the motor the air circulation is interrupted as well, the cool-down is a slow process, as it is mainly dominated by heat conduction. Since the motor can be operated under different conditions and for different amounts of time, the guidelines are conservative in order to cover the worst case scenario.

For a more economic operation (less standstill and therefore less production downtime) it would be desirable to have a real-time prognosis tool that describes the current thermal state of the motor at all critical locations and calculates the lowest possible time-interval to the next restart.

To achieve this goal the following steps are taken:

1.) Build a full scale model of a segment of rotor and stator including conjugate heat transfer and heat sources

2.) Extend the model of the sector to full rotor and stator including winding heads and external housing

3.) Use Model Order Reduction techniques to derive a real-time capable model of the motor in operation.

Figure 3. Rotor during balancing process

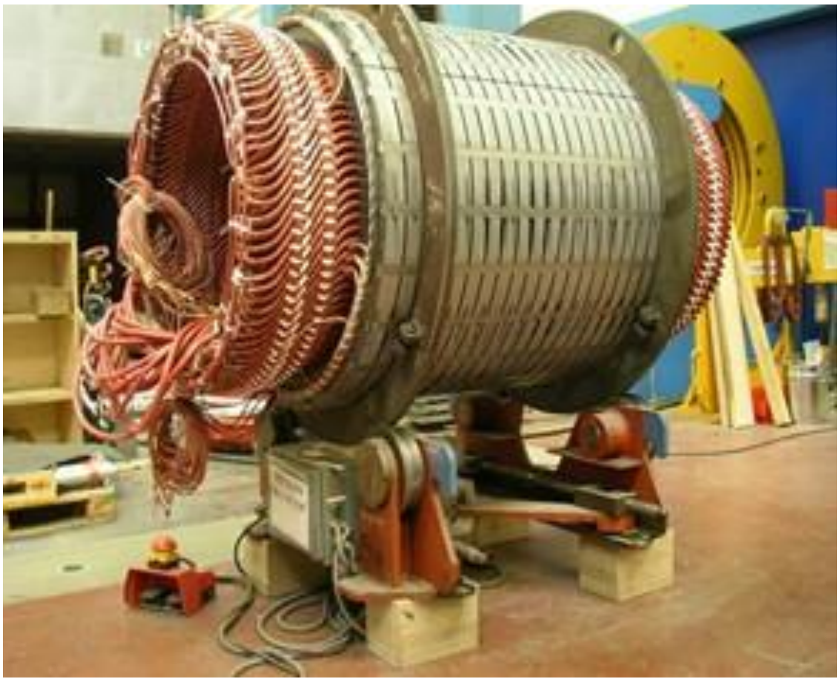


Figure 4. Fully assembled stator

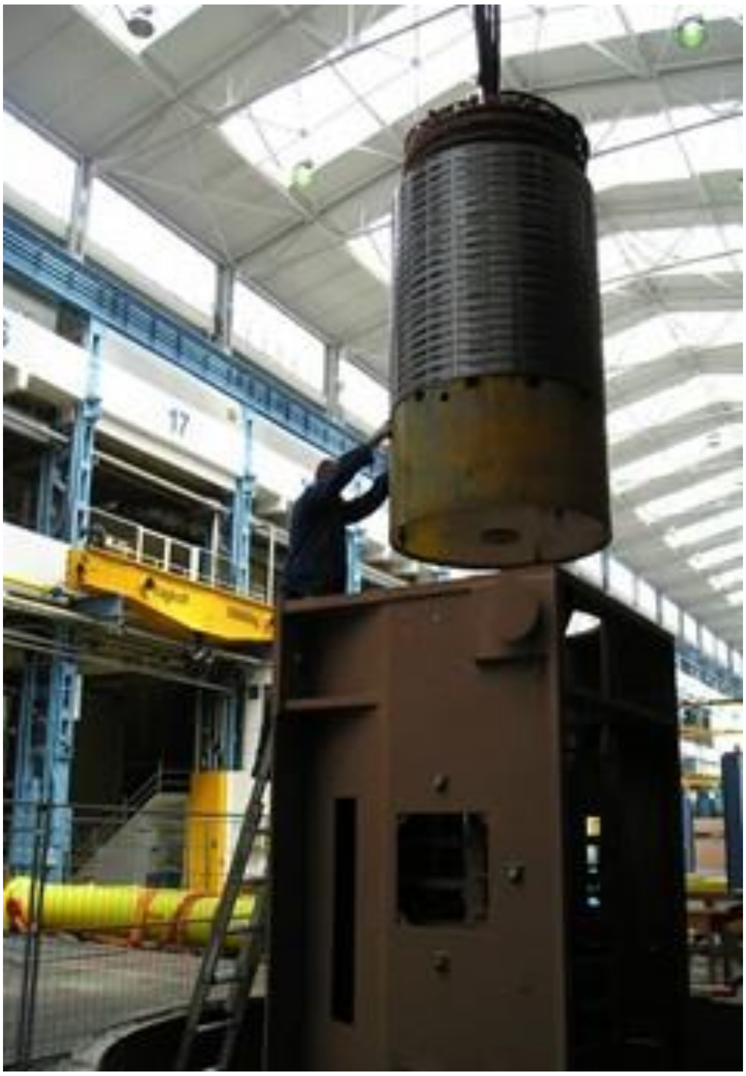
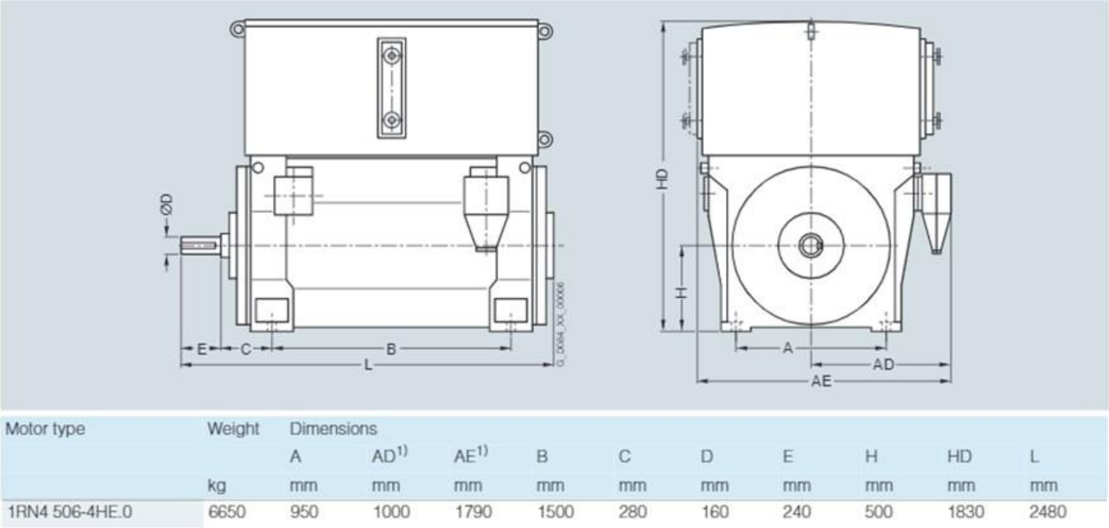


Figure 5. Mounting of stator in housing

2.1 Specification of motor model

For the project the Sinamics H-compact PLUS motor 1RN4 506 was chosen. To give a deeper impression of the motor some pictures from the motor manufacturing process are given in Figure 3, Figure 4, Figure 5 . The overall dimensions can be found in Figure 6.

To set-up a realistic motor model the material parameters given in Table 1 have been proved to be realistic. The nominal heat losses at full load of the motor are given in Table 2. Without loss of generality the load values can be scaled linearly for reduced load.



Motor type	Weight	Dimensions									
		A	AD1)	AE1)	B	C	D	E	H	HD	L
	kg	mm	mm	mm	mm	mm	mm	mm	mm	mm	mm
1RN4 506-4HE.0	6650	950	1000	1790	1500	280	160	240	500	1830	2480

Figure 6. Dimensioning of Simotion motor used in this study

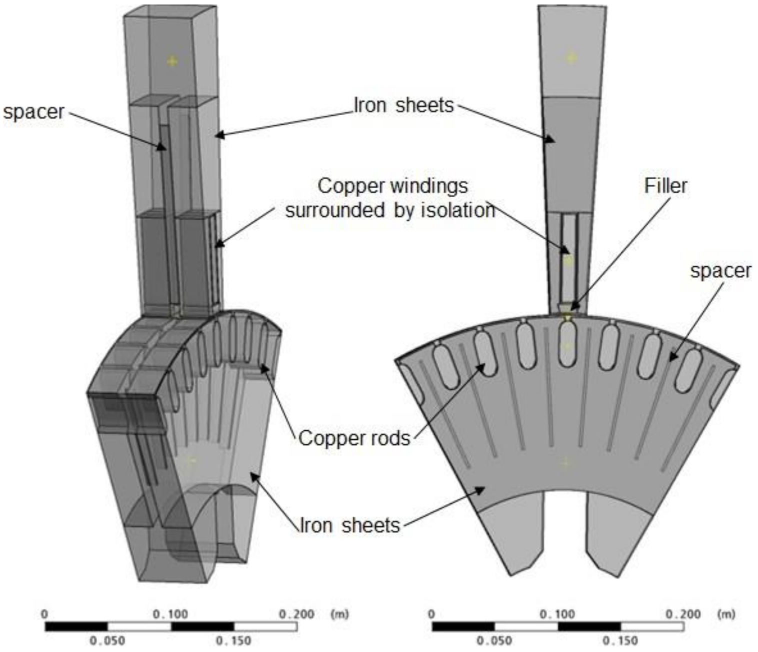


Figure 7. Segment through rotor and stator to explain the position of key components

Table 1. Material properties

Material	Property	Unit	Value
Copper	Density	kg/m^3	8933
	Specific Heat	J/kgK	385
	Conductivity	W/mK	401
Iron sheets	Density	kg/m^3	5000
	Specific Heat	J/kgK	500
	Conductivity axial	W/mK	2
	Conductivity radial	W/mK	27
Spacer	Density	kg/m^3	300
	Specific Heat	J/kgK	500
	Conductivity	W/mK	0.1
Filler	Density	kg/m^3	500
	Specific Heat	J/kgK	2000
	Conductivity	W/mK	20
Insulation copper rods	Heat transfer	W/m^2K	500
Interface filler/iron	Heat transfer	W/m^2K	250

Table 2. Heat sources (given for complete motor) at full load

Location	Heat loss	Remark
Rotor copper cage	25800 W	Volumetric in copper rods
Rotor pulsation & surface losses	3300 W	On surface between rotor & stator
Stator windings	21000W	Volumetric in windings
Stator iron yoke	8000W	Volumetric upper half of iron sheets
Stator iron teeth	22000W	Volumetric lower half of iron sheets(between windings)
Stator pulsation & surface losses	6300W	On surface between rotor & stator

3. Description of Test Cases

3.1 Simulation of steady state conditions using a partial domain + periodicity conditions

The first test case aims at tackling the modelling issues in the 3D simulation starting with the essential components: the rotor and the stator. For simplicity, a periodicity assumption of the configuration is made at this stage. This allows simulating only a sector of the 3D model as displayed in Figure 7. Parametric studies and sensitivity analysis are performed at this stage.

Temperature distribution in the motor and flow behaviour are the key quantities to be analysed. Thermal contact conditions will be employed to simulate thin layers.

3.2 Simulation of unsteady conditions (only one cycle)

At this stage the full motor is modelled, including winding heads and external housing. Due to the complexity of the geometry and the different number of stator and rotor segments, strategies to simulate a reduced domain will be investigated. One possibility would be to simplify the housing and simulate a 60° segment.

In order to provide reliable data for the construction of the ROM models, it is necessary to simulate different operating conditions. Here we want to represent the motor operating at its given rotational speed.

An important simplification to be taken into account is that even for transient conditions, the timescale at which the overheating happens is long when compared to the timescale at which fluid develops. In the practice this implies that in all conditions the fluid can be frozen to a slowly varying time averaged flow, something that can be effectively employed to reduce the computational cost.

3.3 Simulation of unsteady conditions (multiple start-stop cycles)

At this point realistic operating conditions can be simulated. These consist of multiple start and stop cycles. The results of these simulations are the basis for the construction of the unsteady ROM. The definition of the profile to be simulated is based on the requirements coming from the ROM building and the need to cover all possible operating conditions of the motor.

Since the timescale needed for the simulation of such phenomena may be of the order of hours, this would imply the simulation of many thousands of rotor rotations. This difficulty will be sidestepped by the use of “frozen fluid” assumptions.

3.4 Full model reduction (both steady and unsteady)

For the described geometry two types of reduced order models are generated: steady and unsteady. The use of the reduced steady state model is to be seen both in the design phase and at operation. During the design phase a parametric reduced order model is used to quickly assess the machine performance and behaviour under changing parameters like external air temperature, applied load and rotational speed. The generated data is used to define the optimal operating range of the machine. At operation, a fast model can give a measure of the performance losses under extreme conditions and support the decision process as to whether to allow the operation or stop it.

The unsteady reduced order model is intended to be used as a virtual sensor during the machine operation. The model must be able to run in parallel to the machine operation, take in measured data or estimated values as boundary conditions and predict the temperatures at critical positions. The value of the estimated temperature is used to estimate the moment at which the motor can safely be restarted after a shutdown to avoid overheating. In order to be used as a virtual sensor the reduced order model must be at least real-time capable, at best faster than real time to allow prediction into the future and planning of the operation. A maximal error in the estimation of the critical temperature with respect to the full order model of less than 10% is desirable.

Beyond the unsteady case, a ROM model for steady state operation is also necessary. This model should have air temperature, applied load (distribution) and rotational speed as variable input parameters. All models should be based on unstructured meshes on the given (slightly simplified) geometry of the motor.

The full geometry of the rotor and stator including the shaft is presented in Figure 8.

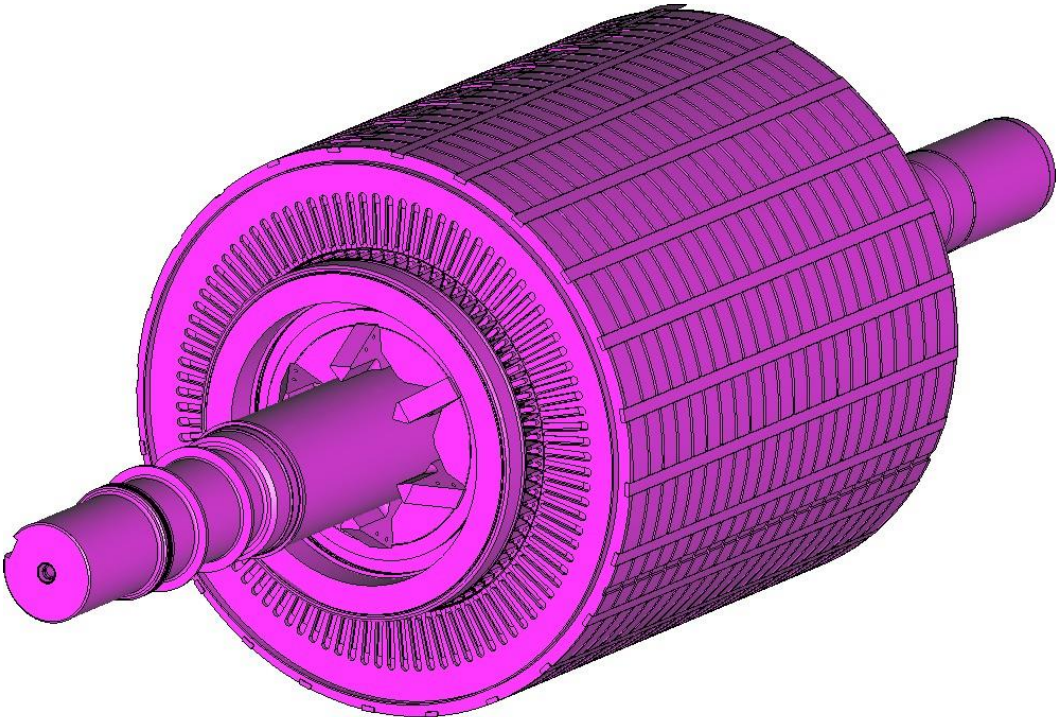


Figure 8. Rotor, motor, and shaft complete.

The number of nodes and elements needed to create a mesh for the whole model is very high, therefore, it is necessary to do some simplifications. The periodicity assumption allows us to simulate only a sector of the full rotor, stator, and shaft.

Mesh will be simplified as much as possible employing MMG (and/or ParMMg capabilities) as needed. Mesh adaptivity will also be applied whenever possible to take advantage of the coupling of Kratos and of the MMG toolbox.

To explain the geometrical properties of the rotor, Figure 9 and Figure 10 shows the different components, where:

- Green: Shaft made from steel.
- Grey: Stack of thin iron sheets (anisotropic conductivity).
- Red: Copper rods.
- Blue: Thin sheet of insulator (Copper rods are surrounded to electrically isolate them).
- Brown: Wood sticks and disk (Stacks of iron sheets are separated by wood sticks and there is a wooden disc on both ends of the rotor).

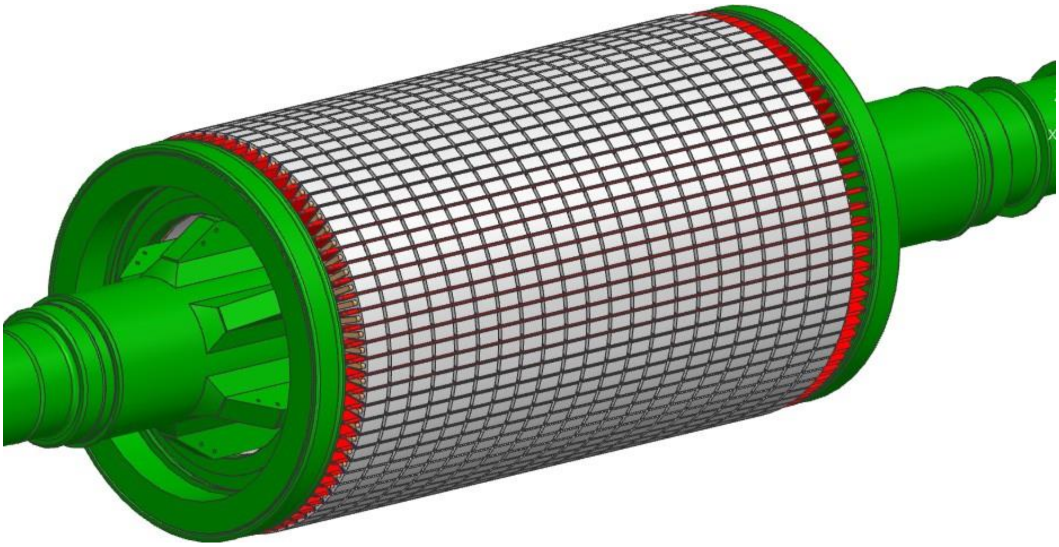


Figure 9. Rotor and shaft.

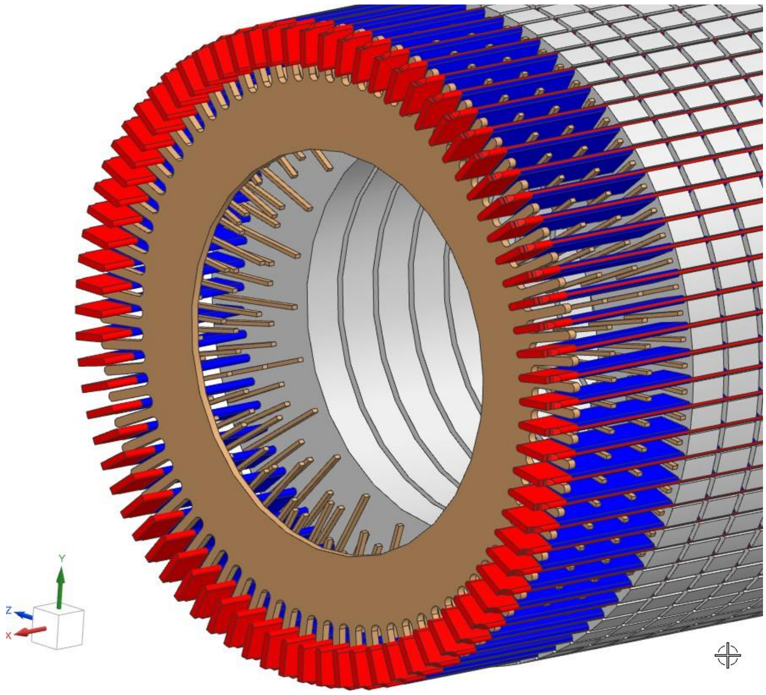


Figure 10. Rotor components.

Working with thin insulation layers (blue) leads to a very complex and fine mesh; this situation can be overcome by merging the insulation layers with the copper rods to become one body and applying an additional heat resistance between rod and iron.

To create the first mesh, a single disk containing half of the iron-sheet stacks is taken from the original geometry. This is shown in Figure 11.

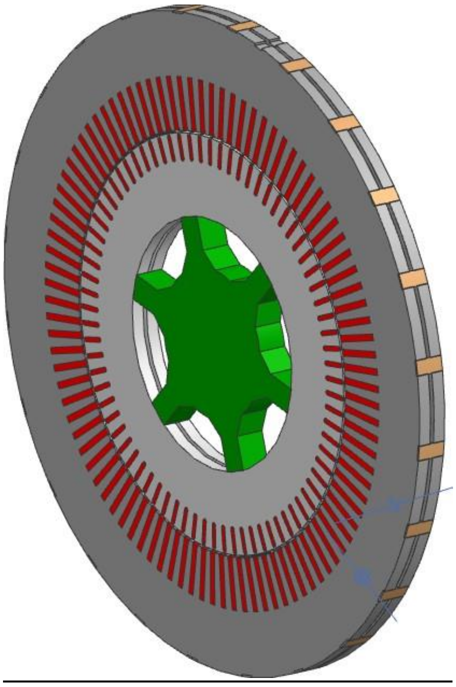


Figure 11. Rotor, stator, and shaft disk.

An extra simplification to this disk can be done by only taking a 30º slice (see Figure 12) which is expected to provide reliable data for the construction of the ROM model.

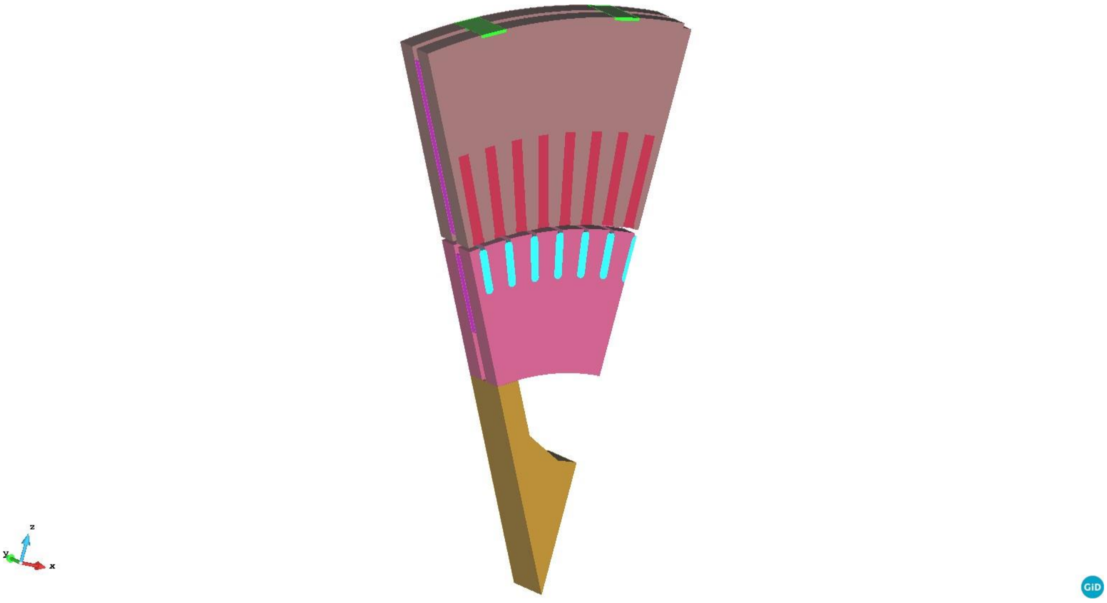


Figure 12. Solid volume rotor, stator, and shaft slice 30º.

The next step is to create the mesh given the previous simplifications. To obtain a well-conditioned mesh, the volumes of the original CAD model were redefined using the pre- and post-processor software GiD. Leading to an initial mesh (not final mesh) shown in Figure 13 with 362613 tetrahedral elements and 75805 nodes.

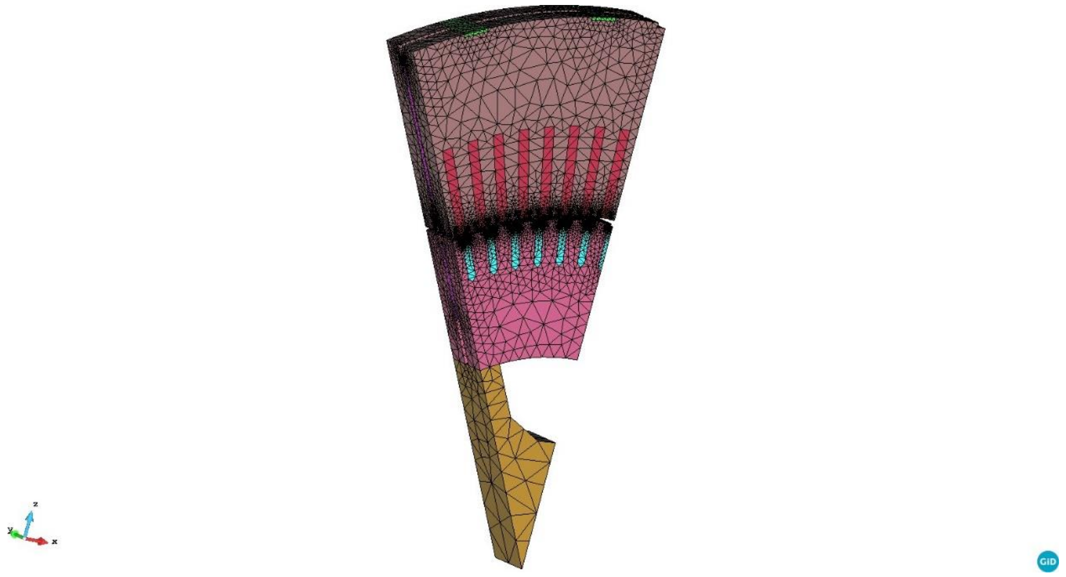


Figure 13. Solid volume rotor, stator, and shaft slice 30º mesh.

To recover the disk, the 30°slice can once be mirrored to give a 60° slice. Copying and rotating this 60° slice 6 times results in a full 360° disc. In addition, this disk can be copied several times to get the full motor.

Moreover, the mesh for the fluid comes from performing a negative mesh (Figure 14) of this simplified slice (Figure 12).

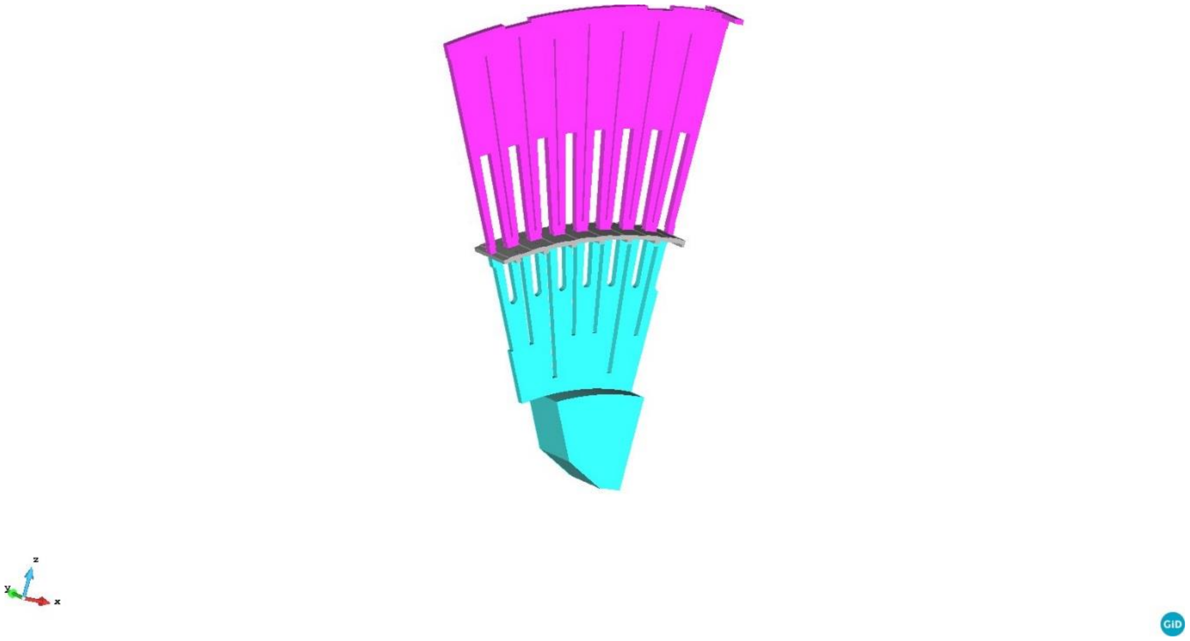


Figure 14. Fluid volume of rotor, stator, and shaft slice 30º (fluid volume).

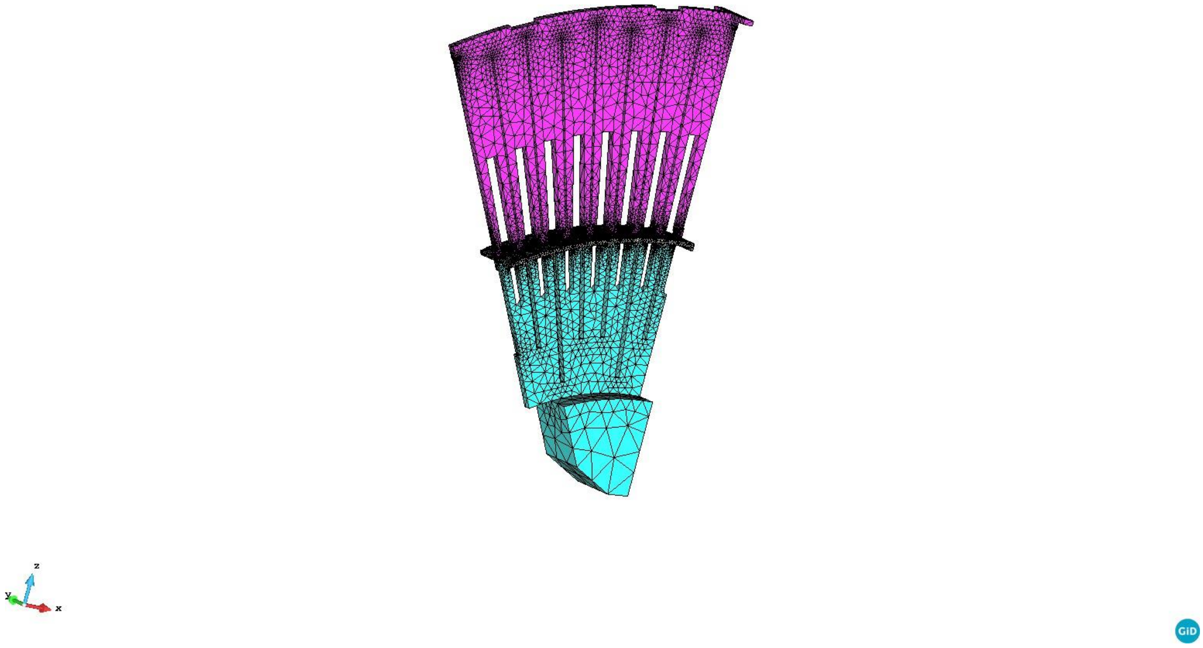


Figure 15. Fluid volume of rotor, stator, and shaft slice 30º mesh.

This Leads to an initial mesh (not final mesh) shown in Figure 15 with 132960 tetrahedral elements and 31659 nodes. Similar to the solid, the original model of the rotor, stator and shaft fluid volumes can be recovered by copying the slice to recover the disk and copying the disk to recover the wanted longitudinal size.

4. Intrusive approach (hyperreduction)

The intrusive approach for constructing the fast reduced-order model is based on the offline/online "hyperreduction" method advocated in [He17] and [He20]. We provide in what follows a succinct description of the steps followed for arriving at such a model.

4.1 Training stage

The training stage consists in solving the coupled finite element equations governing the problem under consideration for representative "training" scenarios, namely:

1. Conservation of mass and momentum equations (Navier-Stokes) for the fluid (air)

$$\boldsymbol{R}^f (\boldsymbol{v}, \boldsymbol{p}, \boldsymbol{T}^f ; \boldsymbol{\mu}) = \boldsymbol{0} \qquad (1)$$

2. Conservation of energy for the fluid

$$\boldsymbol{R}^t(\boldsymbol{v}, \boldsymbol{T}^f , \boldsymbol{T}^s, \boldsymbol{Q}; \boldsymbol{\mu}) = \boldsymbol{0} \qquad (2)$$

3. Heat conduction equation for the solid parts

$$\boldsymbol{R}^s(\boldsymbol{T}^s, \boldsymbol{Q}; \boldsymbol{\mu}) = \boldsymbol{0} \qquad (3)$$

where

- $\boldsymbol{v}, \boldsymbol{p}, \boldsymbol{T}^f$: Vector of nodal velocities, pressures and temperatures for the fluid (N^v, N^p and N^f entries each).
- $\boldsymbol{T}^s$: Vector of nodal temperatures for the solid parts (N^s entries).
- $\boldsymbol{Q}$: Vector of heat fluxes at the interface solid-air.
- $\boldsymbol{\mu}$: "Training" parameters, i.e., those input parameters of the problem that change from one training scenario to another. For simplicity of notation, the time t is included in this set.

To address the coupling of the above described equations, it will be assumed that the variations of temperature in the air do not significantly affect its motion during the heat transfer process. In doing so, the dependence on the temperature field in equation 1 can be dropped, leading to the standard *isothermal Navier-Stokes equations*. Accordingly, the strategy to address the coupled problem will be to first solve such Navier-Stokes equations (using a monoblock, mixed velocity-pressure formulation) under given boundary and initial conditions (encapsulated in the vector of parameters $\boldsymbol{\mu}$), and then attacking the *conjugate heat transfer problem* represented by Eqs. 2 and 3, using the nodal velocities $\boldsymbol{v}$ obtained in the Navier-Stokes analysis as input data.

As for the conjugate heat transfer problem, it will be treated in a staggered manner, using temperature and flux at the common boundaries as coupling variables. More specifically, the temperature of the solid at the common boundaries will be used as Dirichlet conditions for the balance energy equation for the fluid, whereas the reactive flux vector $\boldsymbol{Q}$ will play the role of Neumann condition in the heat conduction problem 3.

4.2 Dimensionality reduction

The outcome of the training phase described in the foregoing will be stored in 3 distinct matrices $\boldsymbol{S}^u$, $\boldsymbol{S}^f$ and $\boldsymbol{S}^s$, containing the velocity/pressure, fluid temperature and solid temperature solutions (or "snapshots") for each training scenario and time step, respectively, i.e.:

$$\boldsymbol{S}^u = [\boldsymbol{U}(\boldsymbol{\mu}1), \boldsymbol{U}(\boldsymbol{\mu}2), \cdots \boldsymbol{U}(\boldsymbol{\mu}P)] \qquad (4)$$

$$\boldsymbol{S}^f = [\boldsymbol{T}^f(\boldsymbol{\mu}1), \boldsymbol{T}^f(\boldsymbol{\mu}2), \cdots \boldsymbol{T}^f(\boldsymbol{\mu}P)] \qquad (5)$$

$$\boldsymbol{S}^s = [\boldsymbol{T}^s(\boldsymbol{\mu}1), \boldsymbol{T}^s(\boldsymbol{\mu}2), \cdots \boldsymbol{T}^s(\boldsymbol{\mu}P)] \qquad (6)$$

Here, P is the total number of snapshots (which is equal to the number of training scenarios times the number of time steps of each scenario), whilst $\boldsymbol{U} = [\boldsymbol{v}^T, \boldsymbol{p}^T]^T$.

The goal in the dimensionality reduction stage is to find a low-dimensional parameterization of the above three matrices, that is, we seek mappings of the form $\boldsymbol{U} = \boldsymbol{H}^u(\boldsymbol{q}^u)$, $\boldsymbol{T}^s = \boldsymbol{H}^s(\boldsymbol{q}^s)$ and $\boldsymbol{T}^f = \boldsymbol{H}^f(\boldsymbol{q}^f)$, where $\boldsymbol{q}^u$, $\boldsymbol{q}^f$ and $\boldsymbol{q}^s$ stand for the low-dimensional counterparts of the vectors of nodal fluid velocity/pressure, fluid temperature and solid temperature, respectively. For the reduced-order model to be considered effective, the dimensions of such vectors, denoted by n^u, n^f and n^s, must be much smaller than its nodal counterparts, i.e.:

$$n^u << N^p + N^v, \qquad n^f << N^f, \qquad n^s << N^s \qquad (7)$$

and, furthermore, the mappings should be able to represent any column of such matrices up to an accuracy threshold ε .

In the present project, we shall consider affine mappings of the form

$$\boldsymbol{U}(\boldsymbol{\mu}) = \boldsymbol{U}_n + \boldsymbol{\Phi}^u(\boldsymbol{\mu})\boldsymbol{q}^u(\boldsymbol{\mu}) \tag{8}$$

$$\boldsymbol{T}^f(\boldsymbol{\mu}) = \boldsymbol{T}^f{}_n + \boldsymbol{\Phi}^f(\boldsymbol{\mu})\boldsymbol{q}^f(\boldsymbol{\mu}) \tag{9}$$

$$\boldsymbol{T}^s(\boldsymbol{\mu}) = \boldsymbol{T}^s{}_n + \boldsymbol{\Phi}^s(\boldsymbol{\mu})\boldsymbol{q}^s(\boldsymbol{\mu}) \tag{10}$$

where the subindex *n* indicates the value of the corresponding variable at the previous time step.

If the basis matrices $\boldsymbol{\Phi}^u$, $\boldsymbol{\Phi}^f$ and $\boldsymbol{\Phi}^s$ are assumed independent of the training parameter $\boldsymbol{\mu}$, then such matrices are given as the left matrix of the truncated Singular Value Decomposition (SVD) of the corresponding snapshot matrix. For instance, for the velocity/pressure snapshot matrix, this decomposition reads

$$\boldsymbol{S}^u = \boldsymbol{\Phi}^u\boldsymbol{\Sigma}^u\boldsymbol{V}^{uT} + \boldsymbol{E}^u \tag{11}$$

where $\boldsymbol{\Sigma}^u$ and $\boldsymbol{V}^u$ designate the matrices of singular values and right singular values, while $\boldsymbol{E}^u$ stands for the truncation term. This term meets the condition $|E^u| < \epsilon|S^u|$; hence, the number of columns of $\boldsymbol{\Phi}^u$, and therefore, the number of reduced velocity/pressure coordinates, is controlled by the user-prescribed tolerance ε.

The SVD, in its standard form, is quite demanding in terms of computational requirements. To alleviate this computational burden, in this project we shall use a method based on the recursive application of the partitioned approach advocated in [He17], in combination with the randomization scheme proposed in [Pe15].

If the number of reduced coordinates for each variable proves to be overly high, we shall explore the derivation of parameter-dependent basis matrices. At its simplest, if the parameters do not change during each training scenario, we shall apply the SVD to the snapshot matrices corresponding to each value of the training parameters, and then determine the basis matrix for any input value $\boldsymbol{\mu}$ by interpolation. It should be noticed that, since the basis matrix are column-wise orthogonal, i.e., $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}$ (this is a property of the SVD), then, to preserve this property in the interpolated basis matrix, the interpolation should be carried out in the manifold of orthogonal matrices of size $N \times n$, as proposed in [Am08].

A more general strategy, valid also for the case in which the input parameters change during the simulation, is to apply some unsupervised clusterization technique such as the *k-means*, and determine basis matrices for each cluster. At each time step, the algorithm has to discriminate which is the most accurate basis matrix for describing the current state of the system by, for instance (see [Am12]), comparing the distances of the converged state and the centroids of each cluster. An even more advanced approach employs “autoencoders” to the same end.

4.3 Reduced-order model

Substitution of the approximations 8, 9 and 10 in the governing equations 1, 2 and 3 leads to an overdetermined system of nonlinear equations in the reduced coordinates $\boldsymbol{q}^u$, $\boldsymbol{q}^f$ and $\boldsymbol{q}^s$. To arrive at a determined system of equations, we shall approximate the weighting functions in the original finite element method by the basis matrices used for the solutions (Galerkin projection). It can be readily shown that this amounts to multiplying the balance equations by the transpose of the basis matrix, i.e.:

$$\boldsymbol{\Phi}^{uT}\boldsymbol{R}^f(\boldsymbol{q}^u;\boldsymbol{\mu}) = \boldsymbol{0} \tag{12}$$

$$\boldsymbol{\Phi}^{f\,T}\boldsymbol{R}^t(\boldsymbol{q}^u, \boldsymbol{q}^f, \boldsymbol{Q};\boldsymbol{\mu}) = \boldsymbol{0} \tag{13}$$

$$\boldsymbol{\Phi}^{sT}\,\boldsymbol{R}^{s}(\boldsymbol{q}^{s},\,\boldsymbol{q}^{f}\,,\boldsymbol{Q};\,\boldsymbol{\mu}) = \mathbf{0} \tag{14}$$

Galerkin projections are known to produce instabilities in advection-diffusion problems such as the energy equation 2 ([Do03]). Should this behaviour also be observed in its reduced-order counterpart 13, we would explore the possibility of applying a least-squares strategy (see [3]) rather than a Galerkin projection for this equation.

4.4 Hyperreduced-order model

The projected equations 12, 13, and 14 can be expanded in terms of its elemental contributions a follows

$$\Phi^{uT} R^{u} = \sum_{e=1}^{M^f} \Phi_e^{uT} R_e^{f} = 0 \tag{15}$$

$$\Phi^{fT} R^{t} = \sum_{e=1}^{M^f} \Phi_e^{fT} R_e^{t} = 0 \tag{16}$$

$$\Phi^{sT} R^{s} = \sum_{e=1}^{M^s} \Phi_e^{uT} R_e^{s} = 0 \tag{17}$$

Here, $\mathbf{R}^{\bullet}{}_{e}$ denotes the contribution to the residual nodal vector of the e-th element, $\boldsymbol{\Phi}^{\bullet}{}_{e}$ stands for the rows of the basis matrix $\Phi^{\bullet}$ corresponding to the nodes of the e-th element; finally, M^f and M^s are the total number of interior/boundary elements of the fluid/solid meshes. These equations evidence that, although the number of equations to be solved at each time step and iteration has diminished from $N^p + N^v + N^f + N^s$ to $n^u + n^f + n^s$, the complexity of the problem still depends on the total number of elements of both meshes. To culminate the complexity reduction process, it is necessary to evaluate the projected residual of each of these 3 balance equations in a more efficient manner —that does not scale with the complexity of the underlying meshes.

In the present project, this additional reduction step —known as hyperreduction— will be carried out using a procedure based on the *Empirical Cubature Method* (ECM), proposed by [He10, He20]. The first step in this method is to solve the reduced-order equations for the same training parameters used for determining the basis matrices. At each step, the projected residual of each equation is stored in 3 matrices $\mathbf{A}^{u}$, $\mathbf{A}^{f}$, $\mathbf{A}^{s}$ (with as many columns as elements in the corresponding mesh) such that

$$\boldsymbol{\Phi}^{\bullet T}\,\boldsymbol{R}^{\bullet} = \boldsymbol{A}^{\bullet}\mathbf{1} = \mathbf{0} \tag{18}$$

where **1** is a column matrix of ones.

Once these matrices have been computed, the ECM seeks a reduced set of elements for the fluid and solid meshes (denoted by E^f and E^s, respectively) such that

$$\Phi^{u^T} R^u \approx \sum_{e \in E^f} \omega_e^u \Phi_e^{u^T} R_e^f \qquad (19)$$

$$\Phi^{f^T} R^t = \sum_{e \in E^f} \omega_e^f \Phi_e^{f^T} R_e^t \qquad (20)$$

$$\Phi^{s^T} R^s = \sum_{e \in E^s} \omega_e^s \Phi_e^{s^T} R_e^s \qquad (21)$$

Notice that the above equations are formally identical to 15 to 17, the only differences being that the index *e* does not run over the whole set of elements (only over the reduced sets E^f and E^s), and that the element contributions in each equation are multiplied by positive weights $\omega_e{}^u$, $\omega_e{}^f$ and $\omega_e{}^s$, respectively. To determine such weights, as well as the reduced sets of weights, we shall proceed as follows:

1. Apply the truncated SVD (using the previously described randomized- partitioning strategy) to $\boldsymbol{A}^u$ (unassembled matrix of the Navier-Stokes residual).
2. Construct a matrix $\boldsymbol{G}^u$ formed by the matrix of right-singular vectors arising from this SVD, and augmented with a row of ones (this is done to eliminate the inherent ill-posedness of the problem, for it admits the trivial solution $\boldsymbol{\omega}^u = \mathbf{0}$).
3. By using the ECM, find a sparse positive vector $\boldsymbol{\omega}^u$ such that

   $$\boldsymbol{G}^u\mathbf{1} = \boldsymbol{G}^u\boldsymbol{\omega}^u \qquad (22)$$

   It is shown in [He17] that the number of nonzero entries calculated by the ECM is equal to the number of rows of $\boldsymbol{G}^u$. The indexes corresponding to such nonzero entries is the desired reduced set of fluid mesh elements E^f
4. Repeat the procedure for $\boldsymbol{A}^f$ (the matrix corresponding to the energy balance residual), that is, construct a matrix $\boldsymbol{G}^f$ from the right-singular vectors of Af , and find a sparse vector ωf such that

   $$\mathbf{Gf\ 1 = Gf\ \omega f} \qquad (23)$$

   In the search of the reduced set, use as initial guess the already computed set Ef —so that the selected set Cf shares as many elements as possible with Ef .
5. Lastly, repeat the procedure for $\boldsymbol{A}^s$ (the matrix corresponding to theresidual of the solid heat conduction problem), and solve the problem

   $$\boldsymbol{G}^s\mathbf{1} = \boldsymbol{G}^s\boldsymbol{\omega}^s \qquad (24)$$

   (the indexes corresponding to the nonzero entries here are denoted by E^s). To account for the coupling between solid and fluid, the initial set in the ECM algorithm should contain the elements of E^f and C^f contained in the common fluid/solid boundary.

The final set of elements to be tracked for the fluid mesh will be formed by the union of E^f and C^f, as well as the boundary elements common to the fluid in E^s. Likewise, the set of elements to be tracked for the solid mesh will be formed by E^s along with the boundary elements of the reduced fluid mesh common to the solid boundary

5. Non Intrusive Approaches

The construction of a reduced order model is based on two different stages. The first stage consists of the approximation of the solution manifold with a low dimensional subspace or manifold. The second stage retrieves the evolution of the system dynamics into this low-dimensional manifold for any value of the input parameters. As described before, intrusive approaches recover the latent variables (reduced basis coefficients) by means of a Galerkin projection of the underlying full order discretization onto the low dimensional manifold.

On the other hand, non-intrusive methods rely only on input-output quantities without the need to have access to the underlying full order model discretization and approximate the input-output function using different paradigms [Bru20]. Using the data collected in the snapshots matrices during the offline stage, non intrusive methods seek a low dimensional approximation of the function:

$$U = F_U(\mu)$$

where we report an example for the velocity field. The different methods differ in the way it is constructed the approximation of the solution manifold and the approach used to retrieve the evolution of the latent variables. We will test and develop both linear and non-linear approaches. Among the linear ones we mention the proper orthogonal decomposition with interpolation [Am08], proper orthogonal decomposition with neural networks [He18] or Gaussian process regression [Te20].

In these approaches the solution manifold is approximated by a linear subspace computed by POD. Neglecting the contribution of the mean part, the approximated solution is then expressed as in intrusive methods with:

$$U(\mu) = \Phi^u q^u(\mu),$$

$$T^f(\mu) = \Phi^f q^f(\mu),$$

$$T^s(\mu) = \Phi^s q^s(\mu).$$

The data coming from the full order snapshots is then used to approximate the functions:

$$q^u(\mu) \approx f^u(\mu),$$

$$q^f(\mu) \approx f^f(\mu),$$

$$q^s(\mu) \approx f^s(\mu).$$

Depending on the methodology this function can be reconstructed by interpolation or regression strategies.

We will also test non-linear approximation techniques that use non-linear methods also to approximate the solution manifold using methods such as convolutional autoencoders. An autoencoder is a type of neural network which learns a nonlinear encoding and decoding of a high dimensional dataset in order to unveil a low dimensional representation (Figure 16).

The autoencoder is then used to define a latent space of reduced dimension which is used for the reduced order model:

$$U \approx g(z)$$

where z is the latent variable of reduced dimension. The mapping between the latent variables and the parameter coefficient values μ is reconstructed using similar approaches as in the linear case with interpolation, neural networks or gaussian progress regression techniques.

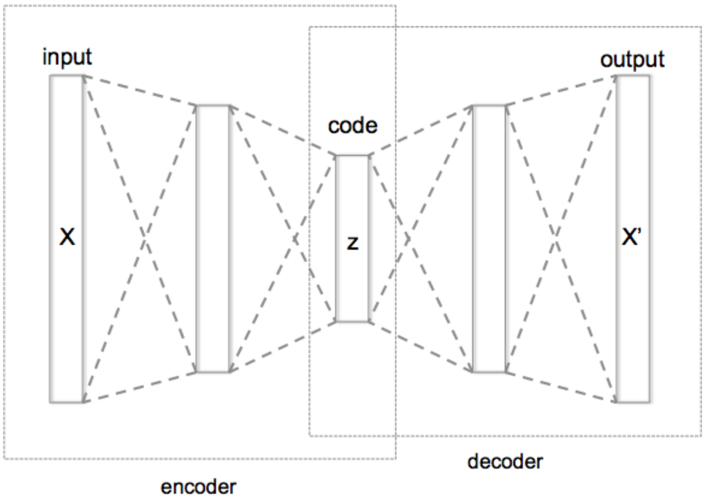


Figure 16. Schematic view of an autoencoder

Convolutional architectures are preferable in this case, since the dimensionality of the problem might be prohibitive and fully connected autoencoders will not be a viable approach. Therefore, we aim to investigate the coupling between linear reduction methods, such as the proper orthogonal decomposition, and non-linear compression techniques [Fre21]. In order to use convolutional autoencoders to compress the solution manifold, which consists of snapshots expressed on an unstructured mesh, a preprocessing step will be required. We aim to test the techniques developed in [Hea21] to the current use case. Both stationary and non-stationary cases will be considered, in case of non-stationary problems we will investigate the applicability of the dynamic mode decomposition [Sch10] and long short term memory neural networks to approximate the evolution of the system dynamics in time [Ha20]. The resulting reduced order models will be compared with intrusive approaches and full order models in terms of accuracy and computational cost.

6. Acronyms and Abbreviations

- HPC – High Performance Computing
- KPI – Key Performance Indicator
- WP – Work Package
- ML – Machine Learning
- DA – Data Analytics
- SVD – Singular Value Decomposition
- ROM – Reduced Order Model
- FOM – Full Order Model

7. References

[Sie17] Three-phase Induction Motors H-compact / H-compact PLUS Catalog D 84.1 ; Siemens AG;2017

[Fre21] S. Fresca and A. Manzoni, "POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition." 2021.

[He18] J. S. Hesthaven and S. Ubbiali, "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *Journal of Computational Physics*, vol. 363, pp. 55–78, Jun. 2018, doi: 10.1016/j.jcp.2018.02.037.

[Te20] 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*, vol. 7, no. 1, Art. no. 1, 2020, doi: 10.1186/s40323-020-00177-y.

[Hea21] C. E. Heaney, Y. Li, O. K. Matar, and C. C. Pain, "Applying Convolutional Neural Networks to Data on Unstructured Meshes with Space-Filling Curves." 2021.

[Bru20] S. L. Brunton and N. Kutz, "Data-driven methods for reduced-order modeling," in *Snapshot-Based Methods and Algorithms*, De Gruyter, 2020, pp. 307–344. doi: 10.1515/9783110671490-007.

[Am08] D. Amsallem and C. Farhat, "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity," *AIAA Journal*, vol. 46, no. 7, Art. no. 7, Jul. 2008, doi: 10.2514/1.35374.

[Ha20] K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata, "Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes," Theoretical and Computational Fluid Dynamics, vol. 34, no. 4, Art. no. 4, May 2020, doi: 10.1007/s00162-020-00528-w.

[Sch10] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, Jul. 2010, doi: 10.1017/s0022112010001217.

[Am12] D. Amsallem, M.J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 2012.

[Ca11] K.Carlberg and C. Farhat. A low-cost, goal-oriented 'compact proper orthogonal decomposition basis for model reduction of static systems. *Interna- tional Journal for Numerical Methods in Engineering*, 86(3):381–402, 2011.

[Do03] J. Donea and A. Huerta. Finite Element Methods for Flow Problems. 2003.

[He17] J.A. Hernández, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687–722, 2017.

[He20] J.A. Hernández. A multiscale method for periodic structures using domain decomposition and ecm-hyperreduction. *Computer Methods in Applied Me- chanics and Engineering*, 368:113192, 2020.

[Pe15] Per-Gunnar Martinsson and Sergey Voronin. A randomized blocked al- gorithm for efficiently computing rank-revealing factorizations of matrices. *arXiv preprint arXiv:1503.07157*, 2015.