

Learning physics simulations for improving design processes in the industry

Fabien Casenave

Abstract. This article is the written version of the Felix Klein prize lecture, given on 16 July 2024, at the 9th European Congress of Mathematics in Seville. In the same fashion as the lecture, general ideas and results are given, and references are provided for the reader interested in details and rigorous developments.

In an industrial group like Safran, numerical simulation of physics phenomena is used in the majority of the design processes of its manufactured products. At Safran's corporate research center, we develop technologies to improve these processes by deriving fast and reliable surrogate models for various domains of physics.

We address two technologies developed in recent years. First, a physical reduced-order modeling method for non-linear structural mechanics and thermal analysis. This method is applied to the lifetime computation of high-pressure turbine blades and the fast heat analysis of high-pressure compressors. Second, the learning of physics simulations under non-parameterized geometrical variability using classical machine learning tools, thanks to non-linear deterministic dimensionality reduction (involving morphing and principal component analysis) and Gaussian process regression, with predictive uncertainties. We also present some of our contributions to the open-source and open-data community.

1. Introduction: Surrogate modeling in the industry

Consider an operator F modeling a large and complex numerical problem or an expensive experiment. Design processes in the industry often require evaluating such expensive operators a large number of times, in so-called many-query tasks. Examples of such tasks are parametric exploration, optimization, uncertainty quantification or calibration. Without any modification or simplification, they are intractable and cannot be applied to real-life industrial configurations.

To give an example of a many-query task, consider the optimization of the shape of an aircraft wing, in order to maximize its finesse under a minimum thickness constraint; see Figure 1. When using an iterative algorithm to compute an approximated

Mathematics Subject Classification 2020: 41A46 (primary); 65N99 (secondary).

Keywords: reduced-order modeling, machine learning, industrial applications.

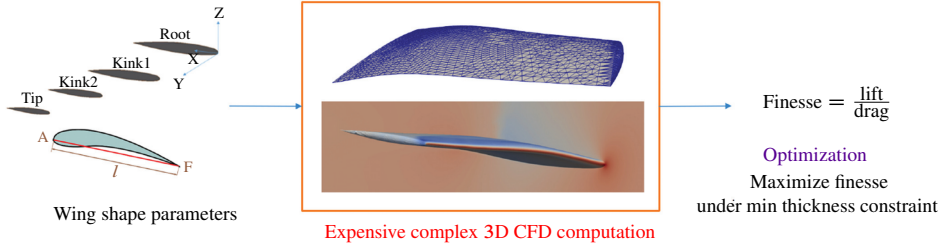


Figure 1. Optimization of the shape of an aircraft wing.

solution of this optimization problem, we need to run the complex fluid solver a large number of times.

A classical way of recovering tractable tasks is by replacing the costly operator F by a surrogate f . The behavior of F is learned by extracting a trend from data. A training database is built by choosing training inputs from a design of experiments (DoE) algorithm and computing the corresponding outputs by evaluating the operator F . The surrogate is constructed in the form of a regressor trained on the obtained input/output couples in the training stage. This surrogate should be accurate enough to replace, in the exploitation stage, the operator F in the considered many-query task and fast enough to make it tractable on an industrial configuration.

A favorable situation concerns tabular problems, where input and output are vectors in fixed low-dimensional vector spaces: $\mathbb{R}^d \ni x \mapsto y = F(x) \in \mathbb{R}^p$, with $d, p \in \mathbb{N}^*$. Many classical methods are available for low-dimensional input-output regression problems, such as linear or polynomial regression, nearest neighbors, random forests, support vector regression or Gaussian process regression.

In this work, we are interested in learning physics problems, which differ from tabular problems in the sense that physical settings involve complex heterogeneous objects of possible variable high dimension, such as meshes, boundary conditions, fields or material models. It should be noted that a physics setting differs from the inputs of a learning problem. The inputs constitute the subset of the physical setting that varies in the considered many-query task. Likewise, the outputs to be predicted by the surrogate model are the subset of the solution that will be used to make an informed decision after the many-query task has ended.

To give an example, consider the AirfRANS physics learning problem [3]; two samples in the training dataset are illustrated in Figure 2. In this learning problem, the physics setting is constituted of the mesh of the fluid domain, the boundary conditions, the Reynolds-averaged Navier–Stokes equations and the $k - \omega$ turbulence model; see [3] for more details. The inputs of the learning problem are the mesh, the angle of attack and the inlet velocity, while the outputs are the velocity, pressure and turbulent velocity fields. The chosen turbulence model is part of the physics setting,

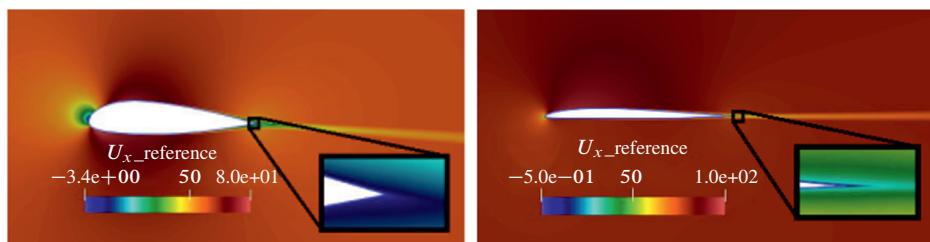


Figure 2. Component- x of the velocity fields of two samples in the AirfRANS dataset.

but being fixed by the samples of the training dataset, it is not part of the inputs of the learning problem.

The advantages of considering a physics learning problem instead of a tabular problem with only scalar inputs and outputs are the possibility to change the shape outside the training parameterization, for instance, in exemption studies, and the possibility to change the post-treatments of the predicted solution fields, after the learning has ended.

In what follows, we start by presenting some physical reduced-order modeling technologies developed at Safran in Section 2. Then, we introduce a machine learning method for dealing with non-parameterized geometrical variability in Section 3. Finally, we present some of our initiatives in the open-source and open-data communities in Section 4.

2. Physical reduced-order modeling

Physical reduced-order modeling (ROM), also called projection-based or *a posteriori* reduced-order modeling, is a class of surrogate modeling methods that relies on the resolution of the underlying physics equations in the exploitation stage.

2.1. Linear data compression

In the training stage, we are given inputs generated using a DoE algorithm and corresponding outputs computed by evaluating the operator F . A linear dimensionality reduction algorithm is then applied to the outputs to construct a reduced-order basis. The exploitation stage consists of applying the same algorithms as the ones used when solving the high-dimensional problems involved in evaluating F , but restricting the solution to the subspace spanned by the reduced-order basis. To give an example, assume that evaluating the operator F consists of solving a costly high-dimensional initial boundary value problem. A numerical solution can be calculated by deriving a variational formulation, choosing a time discretization scheme (for instance, the implicit Euler), a space discretization scheme (for instance, a finite element basis)

and computing at each time step the solution to the obtained system of non-linear equations (using, for instance, Newton's algorithm). In this case, a physical ROM of this problem can be derived by replacing the finite element basis by our computed reduced-order basis. We notice that to keep using the Galerkin method, we need to express the reduced solution as a linear combination of the elements of the reduced-order basis, in the same fashion as the reference solution is expressed as a linear combination of the finite element basis. Many variants have been proposed to take into account other types of problems and numerical simulation methods involved in the operator F evaluations. Other variants in the training stage have also been proposed; for instance, the reduced basis method [22] does not rely on a design of experiments, but rather it starts with a single input/output couple and enriches the data by selecting, in a greedy manner, the configuration where the current reduced-order model makes the largest error. Such enrichment strategies rely on fast and accurate error bounds [7] and can serve in the exploitation stage to provide accuracy assessments and trigger enrichment steps.

In physical ROM methods, the speedup comes from the fact that the cardinality of the reduced-order basis is much smaller than that of the finite element basis, leading to much smaller linear systems to solve in Newton's algorithm of the exploitation stage. However, constructing these small linear systems can be very costly, in particular when numerical integration over the geometrical domain (after discretization: the mesh) is required, which is generally the case with non-linear problems. Being able to assemble the reduced operators involved in Newton's algorithm as fast as possible is critical to obtain practical speedup. Precomputations and constructions of approximations (called hyperreduction) are carried out in the training stage in an operator compression step; see [6, Section 3.2] for more details and [8, Section 1] for references on hyperreduction methods.

We focus on non-linear structural mechanics and non-linear transient heat problems; see [6] and [8], respectively, for details on the considered formulations. In both cases, we rely on the snapshot proper orthogonal decomposition (POD) for the dimensionality reduction step, since it can be easily parallelized when domain decomposition is used. For the operator compression step, we use a localized version of the empirical cubature method (ECM) by deriving an approximate quadrature formula independently for each subdomain. The ECM aims to replace the expensive high-dimensional quadrature schemes (that cannot be precomputed) used to numerically integrate various quantities over the mesh when evaluating the operator F by reduced schemes requiring much fewer quadrature points. The optimization problem to solve consists of minimizing the number of non-zero elements of the vector of quadrature weights in a set of admissible solutions. The latter is the set of vectors with positive components that approximate all the integrals computed when evaluating F on the training inputs, up to a given relative accuracy. This problem being NP-hard, it

is replaced by a tractable greedy approach called non-negative orthogonal matching pursuit algorithm. The positivity of the weights ensures that the reduced operators inherit the spectral properties of F , namely in our context, that the reduced tangent operator in Newton's algorithm in the exploitation stage is guaranteed to be symmetric positive definite. Details on our two physics of interest are given in [6, Section 4.3] and [8, Section 3.2]. In the former, we illustrate the ability of our physical ROM to accelerate the computation of the lifetime prediction of 3D elastoviscoplastic industrial high-pressure turbine blades from a few weeks to a few hours (see Figure 3) and in the latter, to accelerate the computation of transient heat analysis of high-pressure compressors with non-linear radiation boundary conditions and non-linear heat capacity and conductivity, with error compatible with industrial exploitation.

In structural mechanics applications, the quantities of interest are usually dual quantities, such as the plastic strain or the von Mises stress. Reconstruction of dual quantities over the complete mesh can be carried out using the Gappy-POD. Moreover, enforcing the reduced quadrature scheme to include well-chosen quadrature points can ensure the Gappy-POD reconstruction is well posed; see [6, Proposition 1]. In [5], we noticed in our numerical experiments that the prediction error on dual quantities is well correlated to the ROM-Gappy-POD residual, defined as the error on the reduced quadrature points between the online computation of the corresponding quantity by the constitutive law solver and the Gappy-POD reconstruction. Hence, we introduce an error indicator as a regressor trained to predict the error from the

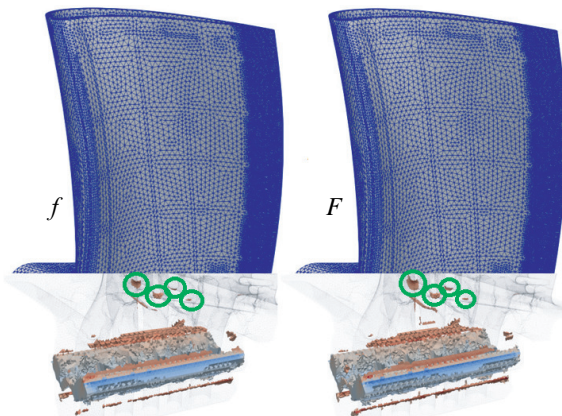


Figure 3. Comparison between the physical ROM f and the reference operator F : largest stress tensor eigenvalue in the areas where the von Mises criterion is large, at max rotation of the 100th cycle. Here, the ROM is trained using the outputs from 1 cyclic loading, and the following 99 cycles are used in the exploitation stage to compute a fast prediction. The four areas circled in green correspond to the locations where the internal cooling channels make a U-turn in the blade: the physical ROM was able to identify them correctly.

ROM-Gappy-POD residual. We illustrate in [5] that this error indicator succeeds in maintaining the error below a chosen threshold by triggering enrichment steps.

Linear dimensionality reduction, or data compression, enables the use of the Galerkin method in the exploitation stage, but this restriction may lead to situations where the cardinality of the reduced-order basis is too large to obtain meaningful speedup factors, even when hyperreduction is used. These situations of poorly reducible problems are the subject of the following section.

2.2. Piecewise linear data compression

The reducibility of the regression problem of approximating F by f using linear dimensionality reduction is evaluated by inspecting the rate of decay of the Kolmogorov N -width of the set of output solutions with respect to the dimension N of an optimal approximation vector space. The Kolmogorov N -width is the worst approximation error of the considered set of outputs, made by the best N -dimensional subspace. When the rate of decay is very fast, the approaches of the previous section, involving linear data compression, can be successful. Otherwise, the reduced-order basis has too many elements, and the reduced problems become slow to solve: the problem is poorly reducible. This is the context of the Ph.D. work by Thomas Daniel [11].

Various techniques have been proposed in the literature to address this situation; see the references in [11]. Here, we focus on the construction of a dictionary of local physical ROMs, using piecewise linear data compression on the outputs; see Figure 4: the outputs are grouped in clusters. Consider again the problem of lifetime prediction of 3D elastoviscoplastic industrial high-pressure turbine blades, and now suppose that the temperature loading field is an input. To be able to use a dictionary of local ROMs, we must know in advance in which cluster of the outputs the solution is located in

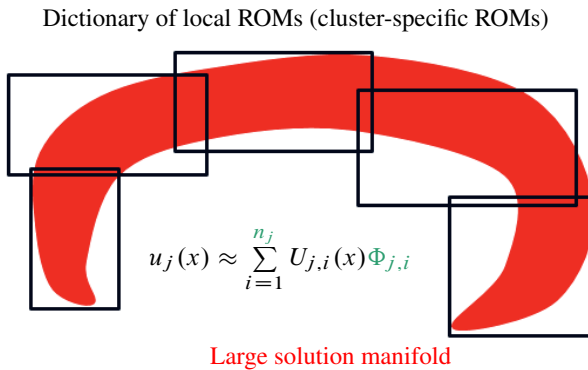


Figure 4. Piecewise linear data compression.

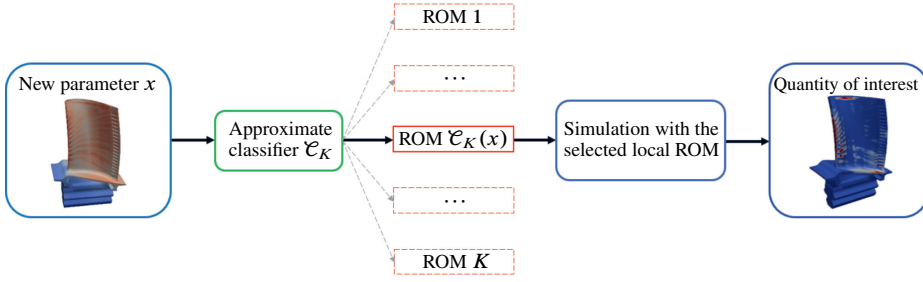


Figure 5. Local ROM recommendation.

order to use the adequate local ROM. To gain this insight, we train a classifier to recommend the right local ROM from the 3D input temperature field, in the same fashion as classifiers in computer graphics that can recognize an object type on a picture; see Figure 5. Such workflows have been carried out in [13]. Contrary to image classifiers, we are in a situation where we have very few inputs, which are of a large dimension. To improve the performance of the training of our classifier, we developed dedicated feature selection and data augmentation techniques in [14].

Partitioning of the outputs has been investigated in the literature using the L^2 norm in the ambient solution space. We notice that the L^2 norm does not necessarily give low-dimensional approximation subspaces; see Figure 6. It appears preferable to group the outputs with respect to their relative orientation instead of their magnitudes. In [12, Property 4.13], we show that the partitions of the outputs minimizing the k -medoid cost function with the sine dissimilarity are exactly the minimizers of the sum of a local (to the partition element) variant of the Kolmogorov 1-width weighted by the volume of the partition element. This result does involve the relative angles between outputs and has an important practical consequence: it provides an algorithm to partition the outputs in an optimal way for our task at hand.

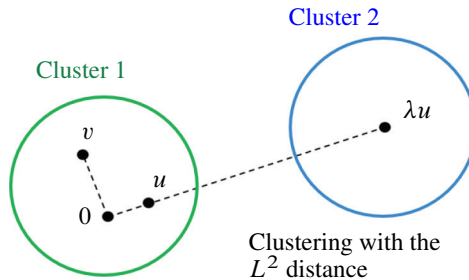


Figure 6. For two clusters, using the L^2 norm leads to cluster 1 having two independent directions, as many as a global ROM. However, considering a cluster with v and another one with u and λu leads to two local reduced-order bases of cardinal 1.



Figure 7. Some output fields for the 2D advection problem [12].

Consider a 2D advection problem, where Gaussians of two amplitudes (small: 0.1 and large: 1) and various vertical positions ξ_2^0 are translated from the left to the right; see Figure 7. For each value ξ_2^0 , both small and large magnitudes are considered as inputs.

Figure 8 illustrates multidimensional scaling representations of the clustering of the 2D advection problem outputs with 5 clusters using the L^2 norm and the sine dissimilarity. We notice that with the L^2 norm, all the outputs corresponding to the small amplitude 0.1 are tightly packed at the center of the figure, meaning that they are likely to be affected by the same cluster (cluster 5 in the figure). By construction, the set of all small magnitude outputs contains all the independent directions of the outputs; hence the Kolmogorov N -width of this cluster is not smaller than that of the set of all outputs. On the contrary, for the sine dissimilarity on the right-hand side of Figure 8, three connected components appear, corresponding to the three values

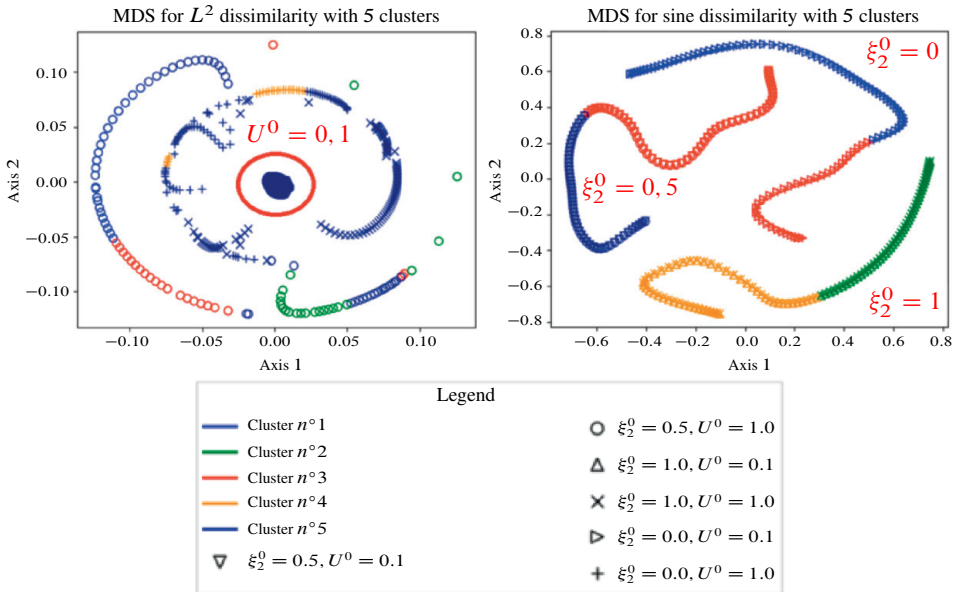


Figure 8. Multidimensional scaling representations of the clustering of the 2D advection problem outputs with 5 clusters using the L^2 norm (left) and the sine dissimilarity (right) [12].

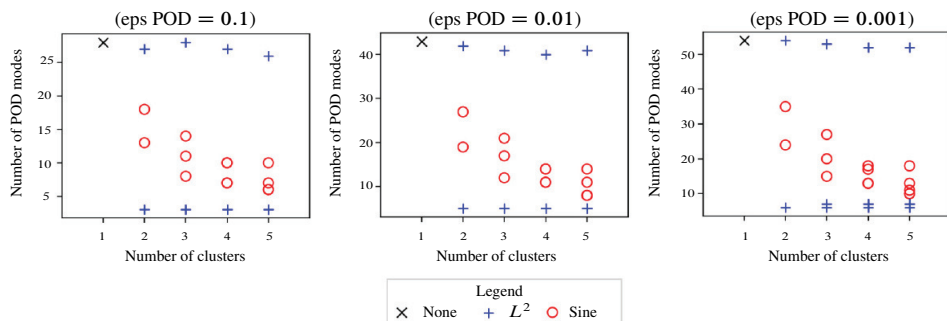


Figure 9. Number of POD modes for each local basis with respect to the number of clusters for various accuracy criteria of the POD truncation applied to the 2D advection problem [12].

of ξ_2^0 . At each position are located two outputs, corresponding to the same values of ξ_2^0 and ξ_1 (horizontal component), for both small and large magnitudes: it means that two outputs varying only by their magnitude have zero sine dissimilarity.

In Figure 9, we investigate the ability of the clusterings using the L^2 norm and the sine dissimilarity to construct local ROMs of small dimensions. Using the L^2 norm, we notice that one of the clusters has the same dimension as a global ROM: it is the one containing all the small magnitude outputs. This means that regardless of the chosen number of clusters, one of them is as poorly reducible as the set of all the outputs. On the contrary, using the sine dissimilarity, the dimension of all the local ROMs decreases as the number of clusters increases: it becomes profitable to cluster the outputs since we succeed in reducing the dimension of the ROMs. This illustrates that the choice of the dissimilarity used in the clustering of the outputs is very important and that the sine one is effective in this situation.

In [15], a dictionary of local ROMs has been constructed using the techniques developed in this section to quantify the uncertainty of the plastic strain and the von Mises stress in areas of interest with respect to the input temperature loading of a high-pressure turbine blade. First, we choose some parameterization of the variability of the input temperature field and construct two DoEs, one MaxProj of size 80 and one Sobol' of size 120. Then, we evaluate the operator F for these 200 configurations (7 days and 9 hours on 48 computer cores). We cluster the outputs on the MaxProj DoE using the sine dissimilarity with the PAM k -medoid clustering algorithm to obtain a partition of size 2 of the outputs (5 minutes). We construct the 2 corresponding local ROMs using the POD-ECM technique presented in Section 2.1 (5 hours on 24 computer cores). We label the 120 outputs of the Sobol' DoE by computing the sine dissimilarity between each output and each of the two previously identified medoids (5 minutes). We apply our feature selection algorithm (16 minutes on 280 computer cores) and train our classifier: a logistic regression with elastic net

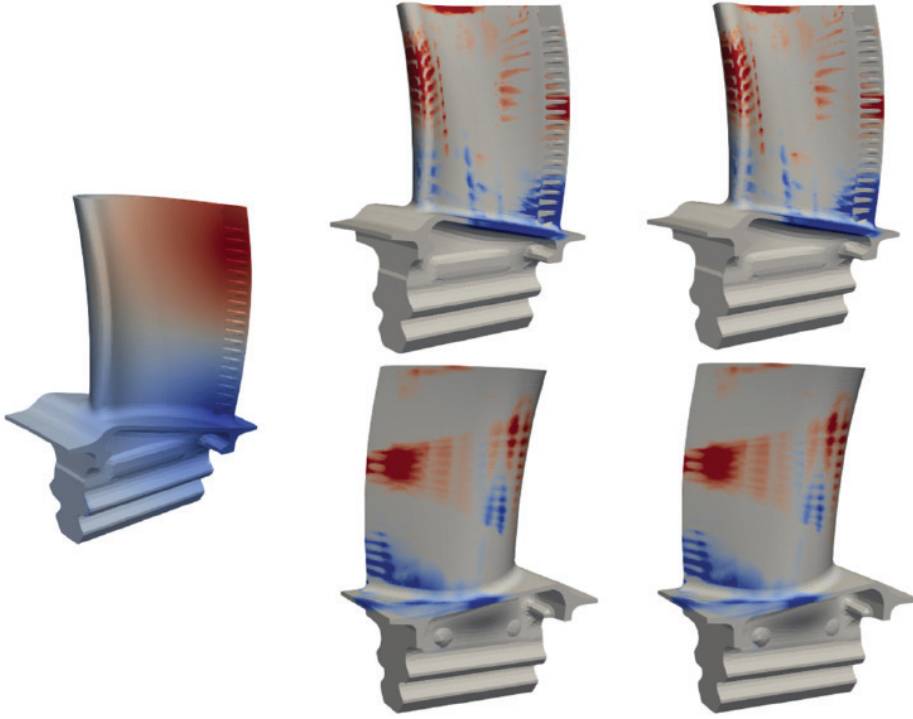


Figure 10. Comparison between high-fidelity predictions (middle column) and the dictionary of ROMs' predictions (right-hand column). The field on the left represents the difference between the current temperature field (belonging to cluster 1) and the reference one. The other fields correspond to the increments of the plastic strain with respect to its reference state obtained with the reference temperature field [15].

regularization (1 minute). At this stage, we have constructed a ROM dictionary of 2 local ROMs and trained the classifier to be able to recommend the local ROM to use in the exploitation stage, from the input temperature loading (see Figure 5); our surrogate f is available. Finally, probability density estimates of the plastic strain and the von Mises stress in areas of interest of the high-pressure turbine blade are computed from 1008 Monte Carlo evaluations of f , with relative errors of 1–2% and a speedup of 600; see Figure 10.

2.3. Limitations of physical reduced-order modeling

Although physical ROMs have been successfully derived for many equations and configurations, two important limitations remain. First, non-parameterized geometrical variations: when two outputs are supported on different meshes, comparing them (to estimate correlations) is not direct. Second, global non-linear data compression,

which would enable the derivation of a single model for poorly reducible problems, prevents us from using the Galerkin method in the exploitation stage.

Various authors have succeeded in tackling each of these limitations, while still constructing a surrogate f that relies on solving the physics equations. However, this comes with the price of complex algorithms, in particular for maintaining practical speedups: to cite a few, geometrical variations [2, 19] and non-linear data compression [1, 20].

In the following section, we propose a surrogate f that relies on non-linear data compression and deals with non-parameterized geometrical variations, but no longer relies on solving the physics equations in the exploitation stage.

3. Machine learning for learning physical problems under non-parameterized geometrical variability

3.1. Mesh morphing Gaussian process

In our recent work presented in [10], we introduce a technique named mesh morphing Gaussian process (MMGP). MMGP integrates four key components: (i) mesh morphing, (ii) finite element interpolation, (iii) dimensionality reduction and (iv) Gaussian process regression. This combination is used to learn solutions to PDEs that involve geometric variations not parameterized explicitly.

Figure 11 demonstrates the inference process for predicting a field of interest using this method. The figure should be interpreted from left to right: on the left side, various geometries corresponding to different samples are depicted, while on the right side, the target fields to be predicted on these geometries are shown.

When dealing with non-parameterized meshes, it is necessary to perform a shape embedding process, converting the meshes into learnable objects. In the MMGP

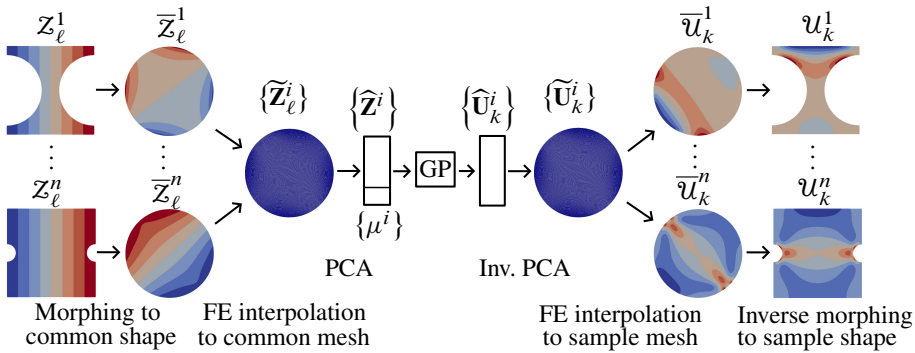


Figure 11. Illustration of the MMGP inference workflow for the prediction of an output field of interest [10].

approach, we utilize the coordinates of the mesh vertices as continuous fields. The left column of Figure 11 illustrates the continuous field of the x-component of the coordinates, naturally showing vertical iso-values. Next, we employ a deterministic morphing process to transform the input meshes into meshes of a chosen common shape, the unit disk in the case of Figure 11. At this stage, depicted in the second column of the figure, each sample is converted into a distinct mesh of the unit disk. This transformation uniquely modifies the coordinate fields for each sample based on the input mesh shapes. We then select a mesh of the common shape and project the coordinate fields from each morphed mesh onto this common mesh using finite element interpolation. Consequently, the coordinate fields of all samples are now represented on the same mesh, thus having a uniform size. This allows us to apply classical dimensionality reduction techniques, such as principal component analysis (PCA), to generate low-dimensional vectors that represent our meshes. If the learning problem includes scalar inputs, we append them at the end of these vectors. It is important to note that our shape embedding relies on the morphing of the coordinate fields, which are transformed according to the input shapes.

Similarly, for the output fields, a field embedding process is required to convert the variable-sized output fields into learnable objects. We apply the same transformations used for the coordinate fields: morphing onto a common shape, finite element interpolation onto a common mesh and PCA compression. This results in low-dimensional vectors that represent our fields of interest.

By applying these deterministic pre-treatments, we reduce the physics learning problem to a low-dimensional input-output tabular regression problem. The complete pre-treatments, including the morphing, should be regarded as the dimensionality reduction step, which is then possibly highly non-linear. The deterministic nature of these pre-treatments is crucial as it simplifies the machine learning stage, removing the complexity of embedding large and variable-dimensional objects. We rely on Gaussian process regression due to its accuracy and its capability to estimate predictive uncertainties effectively.

We refer to [10] for numerical experiments, where MMGP has been found to provide more accurate solutions than recent deep learning technologies (MeshGraphNet and U-net graph convolutional neural network), while being very cheap to train.

3.2. ML4PhySim competition

We have taken part in a competition [17] organized by IRT SystemX and other partners, using our MMGP technology as our entry. This competition consists of learning to predict the pressure, velocity and turbulent viscosity fields on the AirFRANS dataset [3], presented in Section 1.

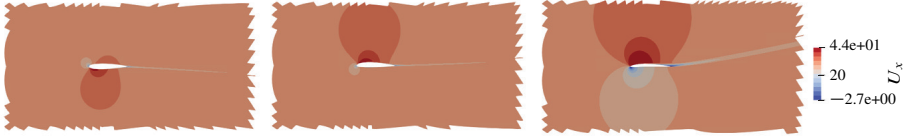


Figure 12. Illustration of the horizontal velocity for three samples of the AirfRANS dataset: the external boundaries are very irregular.

MMGP, being in its early development stages, still needs some dedicated treatment, especially in the morphing step. To derive an effective morphing, we first extend all the 2D meshes up to a common bounding box, since the external boundaries are very irregular; see Figure 12.

For the common shape, we select the geometry of the first pre-treated mesh from the training set. We employ a radial basis function (RBF) morphing algorithm, which involves transforming specific control points and using them to compute the transformation for all other points. The control points chosen for this process include the intrado and extrado of the airfoil, the points at the center of the wake, and the points on the external boundary. Figure 13 illustrates the morphing process applied to the second mesh in the training set. All fields, including input coordinates and output fields, are now projected onto the common mesh, which is the pre-treated and morphed mesh of the first sample of the training set. For dimensionality reduction, we utilize the snapshot proper orthogonal decomposition (POD), as detailed in [10, Annex C]. Objects reconstructed from a reduced representation obtained through snapshot POD are expressed as linear combinations of the compressed samples, preserving all zero linear relationships in the reduced objects. In the AirfRANS dataset, the velocity field satisfies a zero boundary condition at the airfoil surface. Since the competition scoring function involves scalar quantities derived from integrating output fields at the

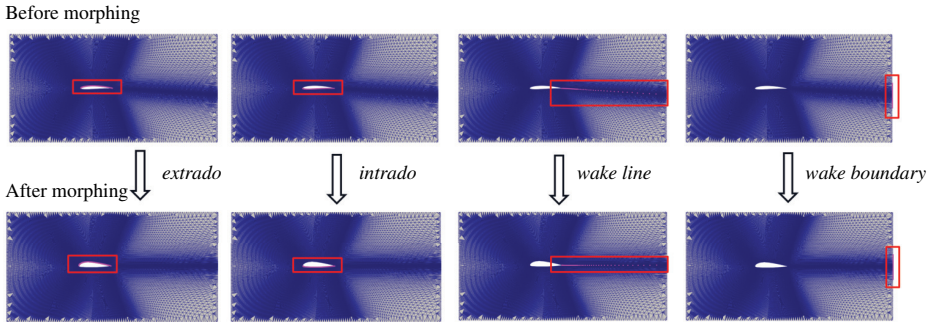


Figure 13. Morphing of the second mesh of the training set.

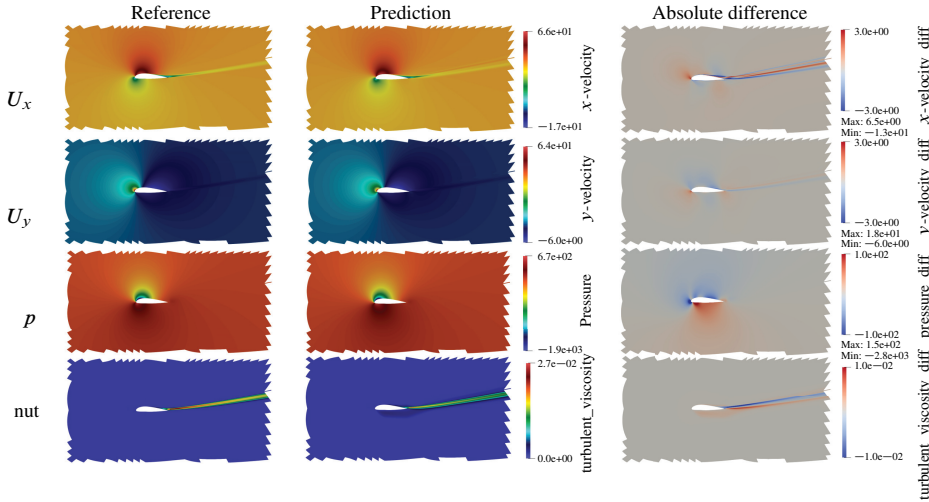


Figure 14. Illustration of the result on a test sample.

airfoil surface, maintaining this zero boundary condition is crucial. By using snapshot POD with MMGP, all predicted fields are expressed as linear combinations of velocity fields, which helps ensure that the zero boundary condition is respected.

We have completed the preprocessing of all meshes, including morphing them, projecting the fields onto a common mesh, and reducing the dimensionality of both the input coordinate fields and output fields. For the competition, we trained separate Gaussian processes for each generalized coordinate associated with the snapshot POD of each output field of interest. The Gaussian process kernel incorporates a constant term combined with an RBF kernel and a white noise term.

Some results are illustrated in Figure 14.

The score is a scalar between 0 and 100 that takes into account the speedup of the surrogate model and the accuracy of the output fields, as well as scalars obtained by integrating these fields (lift and drag coefficients, and their Spearman's rank correlation coefficients), on two different test sets, including one being out of distribution. In Figure 15, the final leaderboard of the competition is presented. Our method ranked first and won the competition, despite involving very simple and classical tools, while our competitors used various state-of-the-art deep learning methods.

3.3. Remarks and outlook

MMGP offers a method for approximating both field and scalar outputs of interest related to complex physics problems involving non-parameterized geometrical variability. The approach introduces two key innovations: firstly, the combination of mesh morphing with finite element interpolation, and secondly, a shape embedding process that uses dimensional reduction of coordinates, treated as continuous fields over the

		Criteria category																	Global score (%)	
		ML related (40%)					Physics (30%)					OOD generalization (30%)								
		Accuracy (75%)			Speedup (25%)	Physical criteria		OOD accuracy (42%)			OOD physics (33%)				Speedup (25%)					
Rank	Method	$\bar{u}_x$	$\bar{u}_y$	$\bar{p}$	$\bar{v}_t$	$\bar{p}_s$	C_D	C_L	ρ_D	ρ_L	$\bar{u}_x$	$\bar{u}_y$	$\bar{p}$	$\bar{v}_t$	$\bar{p}_s$	C_D	C_L	ρ_D	ρ_L	
1	MMGP																			28.08
2	GNN-FC																			572.3
3	MINR																			519.21
4	Bi-Trans																			556.46
5	NeurEco																			44.78

Figure 15. ML4PhySim competition final leaderboard [17].

geometric support. This approach eliminates the need for the machine learning task to manage variable-sized samples or to learn implicit shape embeddings. By reducing the dimensions of inputs and outputs, Gaussian process regression can be efficiently applied to smaller objects.

MMGP has several significant advantages. It is capable of handling very large meshes with ease, can be trained on a CPU, is interpretable, demonstrates high accuracy in our numerical experiments and the ML4PhySim competition, and provides readily available predictive uncertainties. We believe that in the context of industrial part design, the configurations typically have relatively low intrinsic dimensions. Therefore, small machine learning models, if properly reparameterized in a deterministic manner (using morphing in the case of MMGP), can be highly efficient. However, MMGP does have some limitations. The morphing process must be customized for each new use case; there are assumptions regarding fixed topology, and the speedup is limited due to the need for computing morphing and finite element interpolation during the inference stage.

As future work, we plan to work on the previously mentioned limitations. In [18], we recently proposed a generic method for computing the morphing of the shape embedding that relies on successive solutions of elasticity problems. Moreover, an efficient procedure allows us to quickly approximate the morphing in the exploitation stage. Next, we plan to derive an optimal notion of morphing, in the sense that it aims to minimize the number of PCA modes, leading to a regression problem in the smallest possible dimension.

4. Open-source and open-data: Playing a role in the community

Since the inauguration of the Safran corporate research center in 2015, we have introduced internal processes to publish our research codes as open-source and the non-confidential datasets that we produced as open-data.

We promote accessibility, quality and robustness of our libraries by resorting to DevOps tools and workflows: collaborative development is enforced using git and code reviews, automatic testing and internal deployment pipelines and extensive documentation. We choose permissive licenses for our codes (MIT or BSD 3-Clause when possible) and our datasets (CC-BY-SA), and our open-source codes are published on GitLab, documentation is deployed on Read the Docs and packages are distributed on conda-forge, as much as possible.

4.1. Open-source codes

Recently, we developed a datamodel called PLAID (Physics-Learning AI Data model) and published an implementation of this datamodel in the form of a library

of the same name. We aim to formalize physics learning problems, starting with the datasets, which involve complex and heterogeneous objects, as introduced in Section 1. If practitioners share the same standard for producing their datasets and training their models, it would favor productive sharing of data and results and streamlined workflows. The library PLAID heavily relies on the CGNS [16] standard and comes with many functions to handle the data.

The codes developed for the methods presented in the previous sections have been published as open-source; see Figure 16. First, we mention Muscat [4, 30], a library including a set of powerful tools to work on meshes in the context of finite element computations, in particular for pre- and post-treatments. Many readers and writers are available, enabling the handling of data computed by various commercial and open-source codes. We aim to promote PLAID [31] at the base of our open-source developments as our standard for physics learning problems. The physical ROM technologies detailed in Section 2 rely on two libraries: mordicus [21], which we co-developed within the MOR_DICUS consortium, and genericROM [26], fully developed by Safran. The implementation of genericROM is non-intrusive, parallel with distributed memory, and can deal with non-parameterized variability. We refer to [25] for details on these notions and a description of mordicus and genericROM. The code for the MMGP method, presented in Section 3, is also available as open-source [28]. Finally, we mentioned tabular learning problems in Section 1, for which powerful and classical methods exist. Our team at Safran has developed an open-source platform called Lagun [27], where workflows involving design of experiments, surrogate modeling, sensitivity analysis, uncertainty quantification, optimization and calibration can be carried out for these tabular problems.

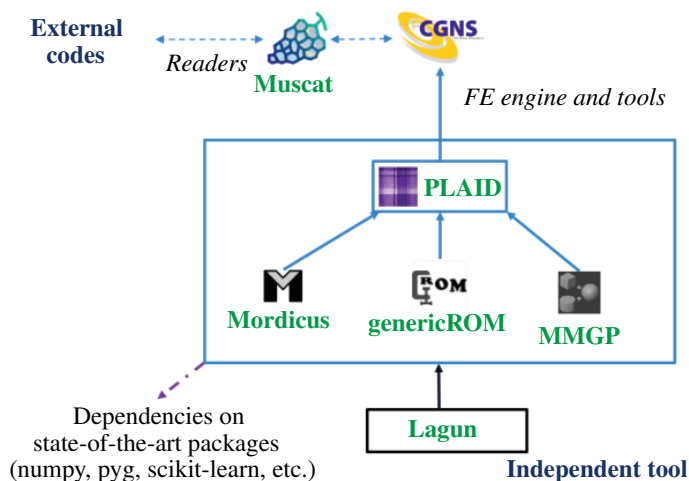


Figure 16. Open-source libraries involved in this article.

4.2. Open-data datasets

We are in the process of distributing the non-confidential datasets on which we develop our physics learning methods as open-data in PLAID format for the community to use. We use the sharing services of <https://zenodo.org> and <https://huggingface.co>: the three datasets used in [10] to compare MMGP with state-of-the-art deep learning methods are available on Zenodo [9, 23, 24] and on Hugging Face.

Hugging Face provides hosting services for machine learning applications called Spaces. We have distributed a demo [29] of our method MMGP applied to the physics learning problem in the ML4PhySim competition presented in Section 3.2; see Figure 17. In this demo, the user chooses a sample ID, which triggers the computation of the MMGP surrogate f , which involves the morphing, the finite element interpolation, the projection of the coordinate fields onto the snapshot POD modes, the inference of the trained Gaussian process regressor and the inverse snapshot POD and finite element interpolation. The user can then choose an output field to plot, which shows the prediction, reference and the error. We mention that this demo

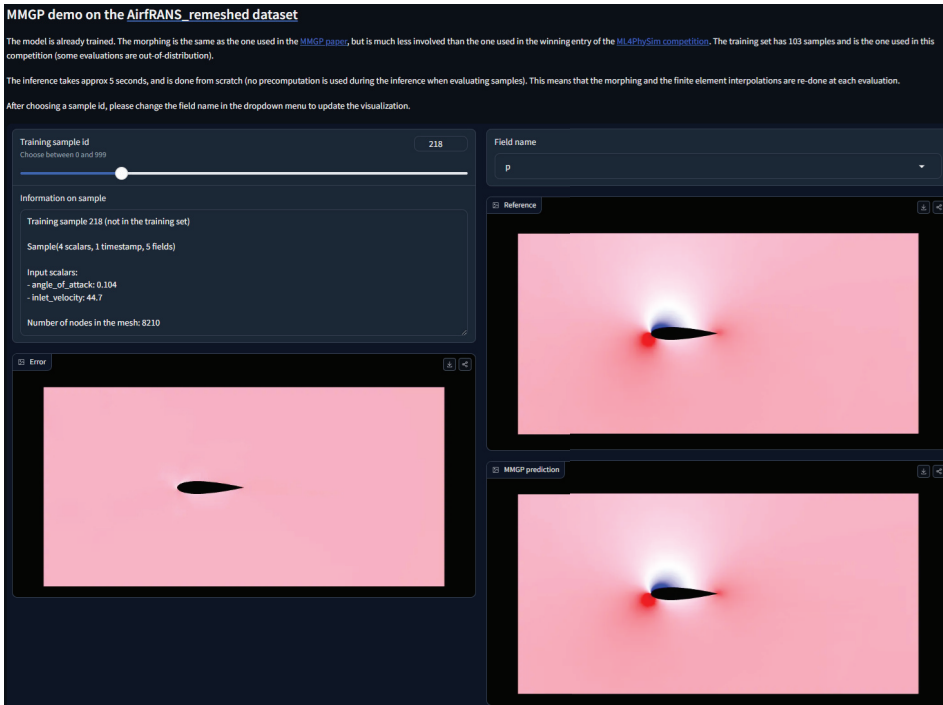


Figure 17. Hugging Face Space hosting a demo of our method MMGP applied to the physics learning problem in the ML4PhySim competition.

involves a remeshed dataset, where the meshes have been coarsened from approximately 180,000 nodes down to approximately 8000 nodes, for the demo to be in real time.

References

- [1] N. Akkari, F. Casenave, E. Hachem, and D. Ryckelynck, [A Bayesian nonlinear reduced order modeling using variational autoencoders](#). *Fluids* **7** (2022), no. 10, article no. 334, 24 pp.
- [2] N. Akkari, F. Casenave, D. Ryckelynck, and C. Rey, [An updated Gappy-POD to capture non-parameterized geometrical variation in fluid dynamics problems](#). *Adv. Model. Simul. Eng. Sci.* **9** (2022), no. 1, article no. 3, 34 pp.
- [3] F. Bonnet, J. A. Mazari, P. Cinnella, and P. Gallinari, [AirfRANS: high fidelity computational fluid dynamics dataset for approximating Reynolds-Averaged Navier–Stokes solutions](#). In *Proceedings of the 36th conference on Neural Information Processing Systems (NeurIPS 2022) track on datasets and benchmarks, November 28–December 9, 2022, New Orleans, LA, USA*, pp. 23463–23478, Adv. Neural Inform. Process. Syst. 35, Neural Information Processing Systems Foundation (NeurIPS), San Diego, CA, USA, 2022
- [4] F. Bordeu, F. Casenave, and J. Cortial, [BasicTools: a numerical simulation toolbox](#). *J. Open Source Softw.* **8** (2023), no. 86, article no. 5142, 4 pp.
- [5] F. Casenave and N. Akkari, [An error indicator-based adaptive reduced order model for nonlinear structural mechanics—application to high-pressure turbine blades](#). *Math. Comput. Appl.* **24** (2019), no. 2, article no. 41, 28 pp. MR 3977955
- [6] F. Casenave, N. Akkari, F. Bordeu, C. Rey, and D. Ryckelynck, [A nonintrusive distributed reduced-order modeling framework for nonlinear structural mechanics—application to elastoviscoplastic computations](#). *Internat. J. Numer. Methods Engrg.* **121** (2020), no. 1, 32–53 Zbl 1548.74772 MR 4047864
- [7] F. Casenave, A. Ern, and T. Lelièvre, [Accurate and online-efficient evaluation of the \$a\$ posteriori error bound in the reduced basis method](#). *ESAIM Math. Model. Numer. Anal.* **48** (2014), no. 1, 207–229 Zbl 1288.65157 MR 3177842
- [8] F. Casenave, A. Gariah, C. Rey, and F. Feyel, [A nonintrusive reduced order model for nonlinear transient thermal problems with nonparametrized variability](#). *Adv. Model. Simul. Eng. Sci.* **7** (2020), no. 1, article no. 22, 19 pp.
- [9] F. Casenave, X. Roynard, and B. Staber, [Tensile2d: 2D quasistatic non-linear structural mechanics solutions, under geometrical variations](#) [Data set]. Zenodo, 2023, <https://zenodo.org/records/14840177> visited on 8 May 2026

- [10] F. Casenave, B. Staber, and X. Roynard, [MMGP: a Mesh Morphing Gaussian Process-based machine learning method for regression of physical problems under non-parametrized geometrical variability](#). In *Proceedings of the 37th conference on Neural Information Processing Systems (NeurIPS 2023), December 10–16, 2023, New Orleans, LA, USA*, pp. 43972–43999, Adv. Neural Inform. Process. Syst. 36, Neural Information Processing Systems Foundation (NeurIPS), San Diego, CA, USA, 2023
- [11] T. Daniel, [Machine learning for nonlinear model order reduction](#). Ph.D. thesis, University of Paris Sciences and Letters, 2021
- [12] T. Daniel, F. Casenave, N. Akkari, A. Ketata, and D. Ryckelynck, [Physics-informed cluster analysis and a priori efficiency criterion for the construction of local reduced-order bases](#). *J. Comput. Phys.* **458** (2022), article no. 111120, 32 pp. Zbl 1560.65714 MR 4393883
- [13] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck, [Model order reduction assisted by deep neural networks \(ROM-net\)](#). *Adv. Model. Simul. Eng. Sci.* **7** (2020), no. 1, article no. 16, 27 pp.
- [14] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck, [Data augmentation and feature selection for automatic model recommendation in computational physics](#). *Math. Comput. Appl.* **26** (2021), no. 1, article no. 17, 25 pp.
- [15] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, and C. Rey, [Uncertainty quantification for industrial numerical simulation using dictionaries of reduced order models](#). *Mech. Ind.* **23** (2022), article no. 3, 20 pp.
- [16] CFD General Notation System (CGNS), <https://cgns.github.io> visited on 8 May 2026
- [17] IRT SystemX, Ansys, NVIDIA, Exaion, and RTE, ML4PhySim challenge. <https://www.codabench.org/competitions/1534> visited on 8 May 2026
- [18] A. Kabalan, F. Casenave, F. Bordeu, V. Ehrlacher, and A. Ern, [Elasticity-based morphing technique and application to reduced-order modeling](#). *Appl. Math. Model.* **141** (2025), article no. 115929, 25 pp. Zbl 1573.65052 MR 4851782
- [19] E. N. Karatzas, F. Ballarin, and G. Rozza, [Projection-based reduced order models for a cut finite element method in parametrized domains](#). *Comput. Math. Appl.* **79** (2020), no. 3, 833–851 Zbl 1443.65348 MR 4049381
- [20] K. Lee and K. T. Carlberg, [Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders](#). *J. Comput. Phys.* **404** (2020), article no. 108973, 32 pp. Zbl 1454.65184 MR 4043884
- [21] MOR_DICUS consortium, Mordicus. Code: https://gitlab.com/mor_dicus/mordicus; doc: <https://mordicus.readthedocs.io> visited on 8 May 2026
- [22] A. T. Patera and G. Rozza, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*. MIT Pappalardo Grad. Monogr. Mech. Eng., Massachusetts Institute of Technology, Cambridge, MA, 2007
- [23] X. Roynard, F. Casenave, and B. Staber, [Rotor37: a 3D CFD RANS dataset, under geometrical variations of a compressor blade](#) [Data set]. Zenodo, 2023, <https://zenodo.org/records/14840190> visited on 8 May 2026

- [24] X. Roynard, F. Casenave, and B. Staber, *AirfRANS_original* [Data set]. Zenodo, 2024, <https://zenodo.org/records/14840387> visited on 8 May 2026
- [25] D. Ryckelynck, F. Casenave, and N. Akkari, *Resources: software and tutorials*. In *Manifold learning. Model reduction in engineering*, pp. 53–69, Springer, Cham, 2024
- [26] Safran, genericROM. Code: <https://gitlab.com/drti/genericrom>; doc: <https://genericrom.readthedocs.io> visited on 8 May 2026
- [27] Safran, Lagun. Code: <https://gitlab.com/drti/lagun>; doc: <https://lagun.readthedocs.io> visited on 8 May 2026
- [28] Safran, MMGP. Code: <https://gitlab.com/drti/mmgp>; doc: <https://mmgp.readthedocs.io> visited on 8 May 2026
- [29] Safran, MMGP demo. https://huggingface.co/spaces/fabiencasenave/MMGP_demo visited on 8 May 2026
- [30] Safran, Muscat. Code: <https://gitlab.com/drti/muscat>; doc: <https://muscat.readthedocs.io> visited on 8 May 2026
- [31] Safran, PLAID. Code: <https://github.com/PLAID-lib/plaid>; doc: <https://plaid-lib.readthedocs.io> visited on 8 May 2026

Fabien Casenave

Digital Sciences and Technologies Department, Safran Tech, Rue des Jeunes Bois,
Châteaufort, 78114 Magny-Les-Hameaux, France; fabien.casenave@safrangroup.com