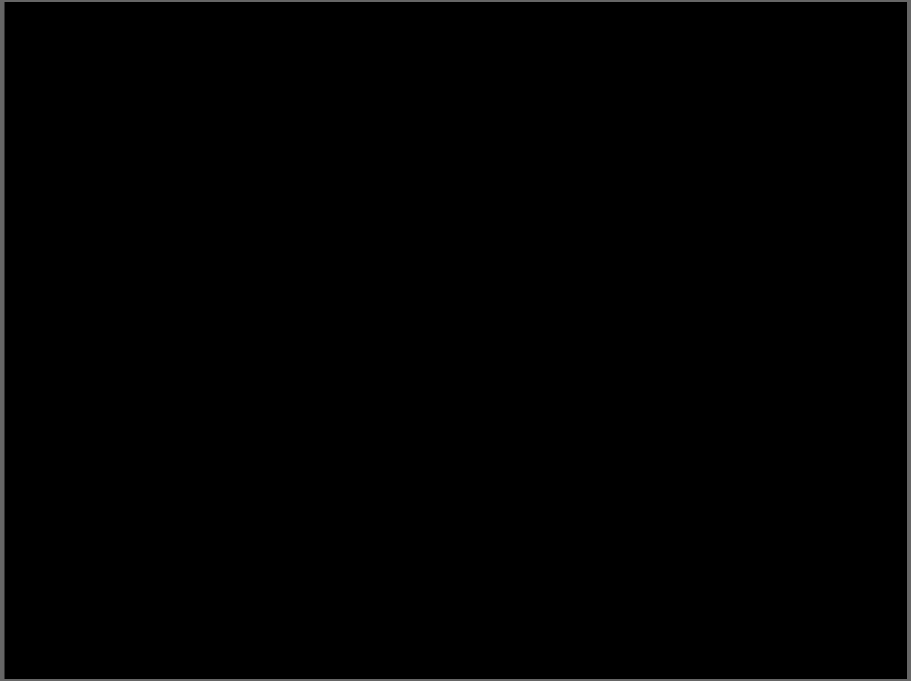
Model Reduction in Aeroelasticity for Preliminary Design



Oriol CHANDRE VILA

Supervisors:

Joseph MORLIER (ISAE SUPAERO)
Sylvain DUBREUIL (ONERA)

March 13th, 2019

Aim and Goals

AIM:
To study the use of Reduced Order Models (hereafter ROM) in order to **lighten** the computational requirement in fluid-structure problems simulations.

GOALS:
- To create a tool for exploring the design domain in real time for Preliminary Aircraft Design.
- To apply ROM strategies to the actual solvers.

Overview of the Project

February 2018 — June 2018 — September 2018 — March 2019

PIR		PIE

Reduced Order Modeling For Real Time Aeroelastic Simulation

➔ Study of differents methodologies.
➔ Selection of one methodology.

GitHub

Internship in ONERA

➔ Application of the selected approach to a fluid-structure interaction code.

GitHub

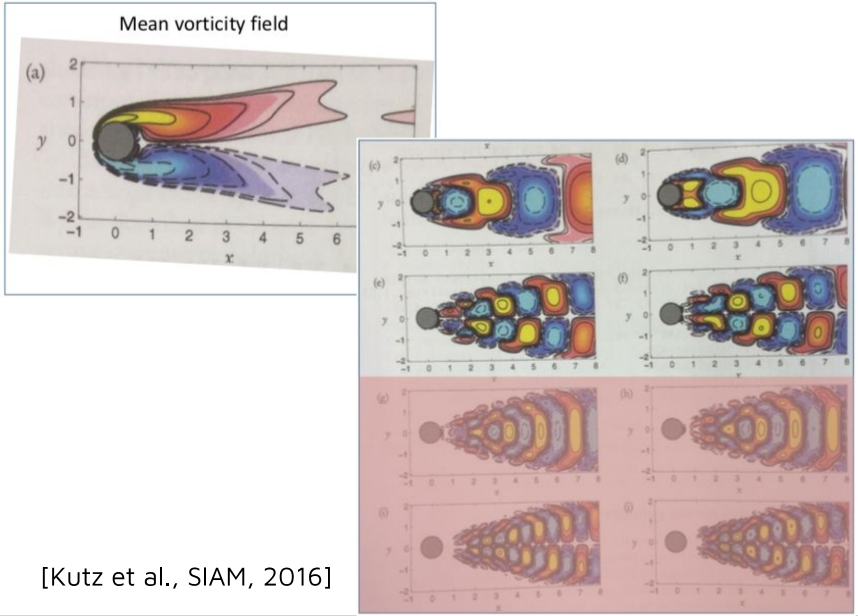
Model Reduction in Aeroelasticity for Preliminary Design

➔ Improvement on the ONERA internship’s code.
➔ Application to OpenAeroStruct (v2).

Table of Contents

Methodology: ROM

Reduced Order Methods (ROM) in a nutshell



[Kutz et al., SIAM, 2016]

[Rozza, SISSA, 2017]

$$\mathbf{M}(\xi) \cdot X(\xi) = f(\xi)$$

- **Offline**: very expensive preprocessing.
  $(\ )^{\mathcal{N}}$: “truth” high order method - *to be accelerated*
- **Online**: extremely fast real-time input-output evaluation thanks to an efficient assembly of problem operators.
  $(\ )_N$: ROM - *the accelerator*

$$X(\xi^{new}) = \sum_{i=1}^{nModes} \alpha_i(\xi^{new}) \cdot \Phi_i$$

Methodology: POD

Principal Component Analysis (PCA)

[Kutz, Chpt 15, Oxford University Press, 2013]

SVD in Algebra
POD in Mechanical Engineering

Observation → Orthogonal Transformation → Singular Values

- Linear parametric problem:
  $\mathbf{M}(\xi) \cdot X(\xi) = f(\xi)$
- Observation matrix:

$$\mathbf{O} = \begin{pmatrix} X(\xi^{(1)}) \; \ldots\ldots \; X(\xi^{(nObs)}) \end{pmatrix}$$

$$\mathbf{O} = \mathbf{L} \cdot \mathbf{\Sigma} \cdot \mathbf{R}^T$$

$$\mathbf{O_r} = \mathbf{L} \cdot \mathbf{\Sigma} \cdot \mathbf{R}^T \longrightarrow \begin{cases} \mathbf{L} = [\ldots,k] \\ \mathbf{\Sigma} = [k, k] \\ \mathbf{R} = [k,\ldots] \end{cases}$$

k: # Reduction modes

Methodology: Offline Phase Algo

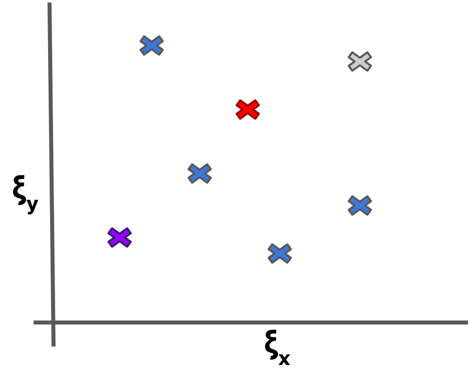
Construction of the L by Greedy Algorithm

[Amsallem, International Journal for Numerical Methods in Engineering, 2014]

1. Select randomly a first sample: $\boldsymbol{\xi}^{(1)}$
2. Solve the problem: $X(\boldsymbol{\xi}^{(1)})$
3. Build the Reduced Base (RB): **L**
4. for $i = 2, \dots, k$
   a. for $j = 1, \dots, nCandidates$
      i. Evaluate the candidate and $\boldsymbol{\xi}^{(i)} = \text{argmax} \, \| \mathbf{r}(\boldsymbol{\xi}^{(j)}) \|$
   b. Solve the problem: $X(\boldsymbol{\xi}^{(i)})$
   c. Rebuild **L** on the samples $\boxed{\{X(\boldsymbol{\xi}^{(1)}), X(\boldsymbol{\xi}^{(2)}), \dots, X(\boldsymbol{\xi}^{(i)})\}}$

O

Problem: $\mathbf{M}(\xi) \cdot X(\xi) = f(\xi)$



$$\mathbf{M}(\xi^{new}) \cdot \left[\sum_{i=1}^{nModes} \alpha_i(\xi^{new}) \cdot L_i \right] - f(\xi^{new}) = \mathbf{r}(\xi^{new})$$

Methodology: Online Phase Algo

[Kutz, Chpt 15, Oxford University Press, 2013]

$$\begin{cases} \Phi^{-1} \cdot \mathbf{M}(\xi^{new}) \cdot \Phi \cdot X(\xi^{new}) = \Phi^{-1} \cdot f(\xi^{new}) \\ X(\xi^{new}) = \sum_{i=1}^{nModes} \boxed{\alpha_i(\xi^{new})} \cdot \Phi_i \end{cases}$$

- ➢ Φ ? ≡ **L**
- ➢ How can we compute…
  - ○ Reduction Problem Resolution
  - ○ Interpolating the coefficient at the points $\{\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(i)}\}$

Methodology: Kriging

[Rasmussen et al., the MIT Press, 2006]

[Matheron, Masson, 1965]

Substitution Models

Linear Regression

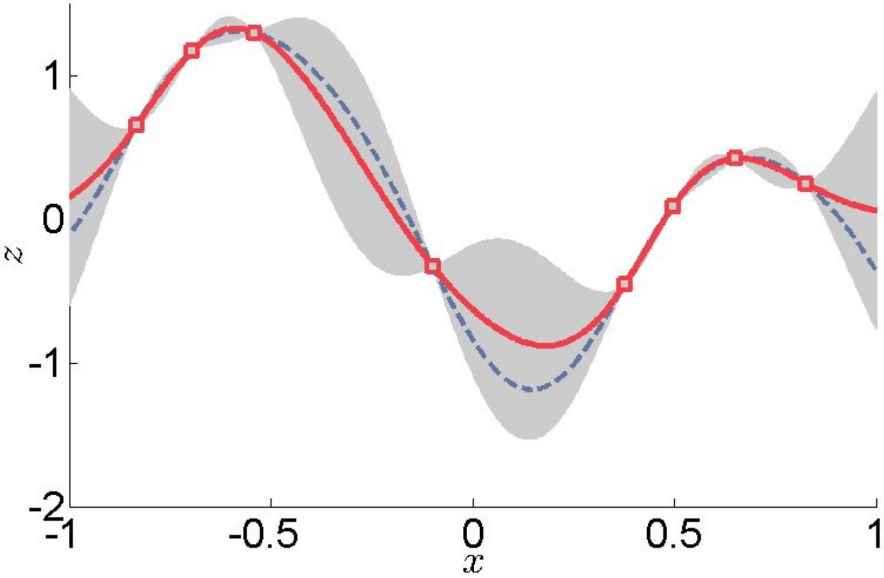
Interpolating Models

$$X(\xi^{new}) = \sum_{i=1}^{nModes} \hat{\alpha}_i(\xi^{new}) \cdot \mathbf{L}_i$$

Kriging Models

$$X(\xi^{new}) = \sum_{i=1}^{nModes} \hat{\alpha}_i(\xi^{new}, \omega) \cdot \mathbf{L}_i$$

Methodology: Kriging



[Wahba et al., SIAM, 1990]

$$\hat{X} \sim \mathcal{N}(\mu_{\hat{X}}, \sigma^2_{\hat{X}})$$

$$\begin{cases} \mu_{\hat{X}} = \sum_{i=1}^{nModes} \mu_{\hat{\alpha}_i} \cdot \mathbf{L}_i \\ \sigma^2_{\hat{X}} = \sum_{i=1}^{nModes} \sigma^2_{\hat{\alpha}_i} \cdot (\mathbf{L}_i)^2 \end{cases}$$

Application to Static Aeroelasticity

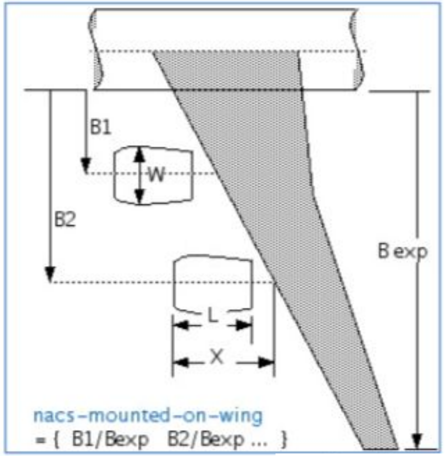
$$\begin{cases} \mathbf{A}(\xi)\cdot\Gamma(\xi) = b(\xi) \\ \mathbf{K}(\xi)\cdot U(\xi) = F_S(\xi,\Gamma) \end{cases} \longleftrightarrow \begin{cases} \Gamma(\xi) = \sum_{i=1}^{nModes} \hat{\alpha}_i(\xi,\omega)\cdot\mathbf{G}_i \\ U(\xi) = \sum_{i=1}^{nModes} \underbrace{\hat{\beta}_i(\xi,\omega)}\cdot\mathbf{D}_i \end{cases}$$

So it is possible to estimate the variance

- Fixed-point iteration within the Greedy ⇒ We converged it for the Reduced Problem.

- Real code → **Verified** and **Validated** $\begin{cases}\textbf{VLM}\\ \textbf{FEM}\end{cases}$

Application to Static Aeroelasticity: Problem



Source [8]

Aerodynamics → VLM Horseshoe

Matrix of Transfer & Mesh deformation

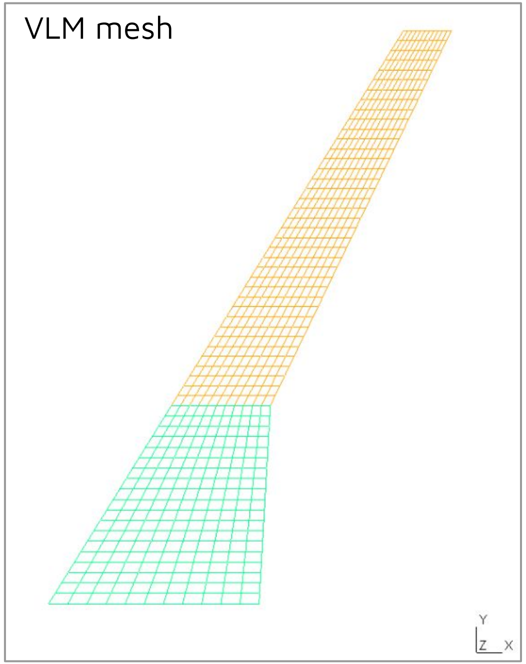
Structures → FEM (DKT for plates)

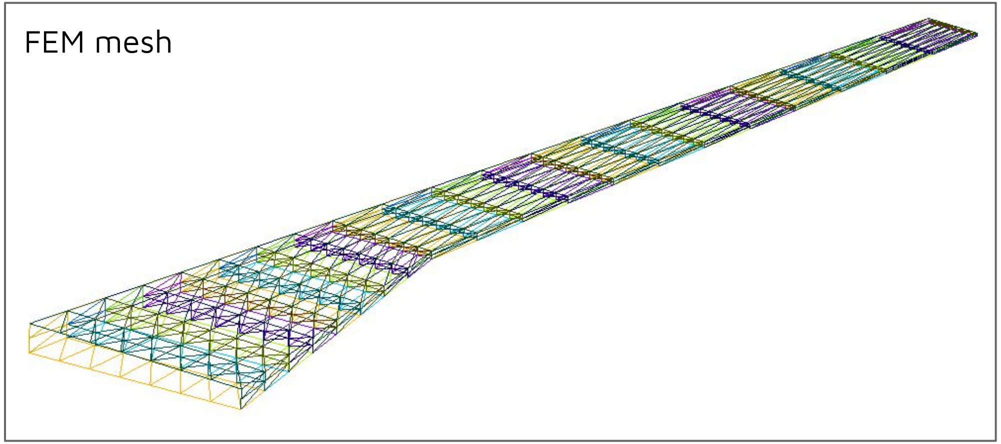
6 parameters of interest

- h_{skin}
- h_{ribs}
- h^{LE}_{sparse}
- h^{TE}_{sparse}
- Span
- Wing Area

ξ	Min	Max
h_{skin} [mm]	1.45	30
h_{ribs} [mm]	1.45	15
h^{LE}_{sparse} [mm]	4.35	90
h^{TE}_{sparse} [mm]	4.35	90
Span [m]	34	80
Wing Area [m^2]	122	845

Application to Static Aeroelasticity: Results





nSamples = 10 | *nCandidates* = 50 | *nKriging* = 150

Aeroelastic code solver ≈ 1min

Offline code ≈ 7h30

Online code ≈ 0.3 s

Conclusions

- Two nature of errors:
  - Reduced Problem.
  - Interpolation Error.
- Parametric study:
  - *nSamples* (base dimension) drives the accuracy of the RB. For improving the results, we could...
    - Use more modes (higher *nSamples*).
    - Reduce the domain of the problem.
    - Implement another method to converge the Reduced Problem inside the Greedy Algorithm.
  - *nCandidates* + *nKriging*: points of interpolation.
- **Saving** in computation time for the real time simulation.

Future Steps

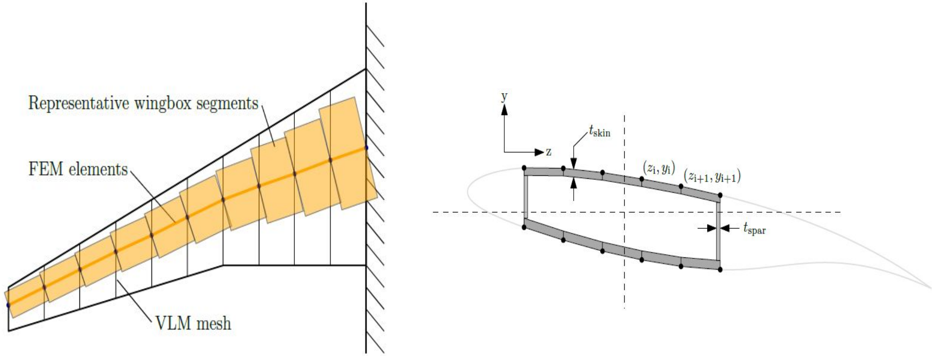
OpenAeroStruct_v2: **VLM** + **Wingbox FEM**

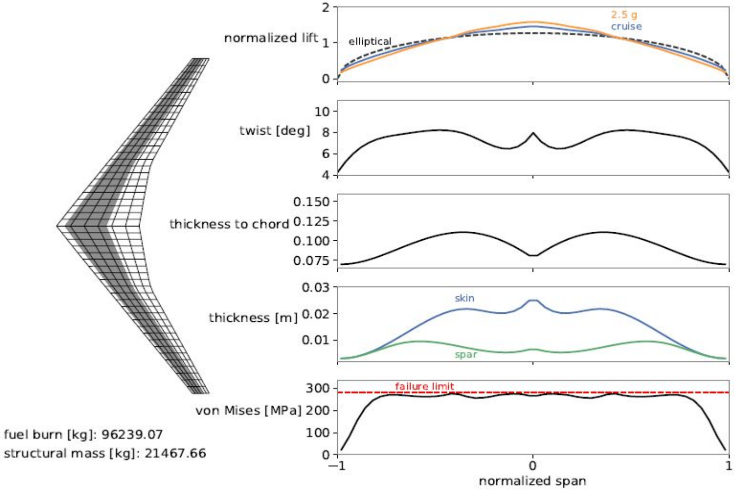
→ *Optimizer*: openMDAO

[Jasa et al., Structural and Multidisciplinary Optimization, 2018]

Reduced Base using POD+Kriging methodology for 17 parameters:

- ★ 4 for Aerodynamics
- ★ 4 for Structures
- ★ 3 for Fuel Data
- ★ 6 for Mission Requirements





[Chauhan et al., EngOpt, 2018]

References

[1] Kutz, J.N., Brunton, S.L., Brunton, B.W. and Proctor, J.L., "Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems", 1st ed., SIAM, Philadelphia, 2016.
[2] Rozza, G., Ballarin, F., Mada, Y., "Reduced order methods for parametric Fluid-Structure Interaction problems: applications to haemodynamics", SISSA International School for Advanced Studies, 2017.
[3] Kutz, J. N., "Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data", 1st ed., Oxford University Press, Oxford, 2013.
[4] Amsallem, D., "An Adaptive and Efficient Greedy Procedure for the Optimal Training of Parametric Reduced-Order Models", International Journal for Numerical Methods in Engineering, 2014.
[5] Rasmussen, C.E., Williams, C.K.I., "Gaussian Processes for Machine Learning", the MIT Press, 2006.
[6] Matheron, G., "Les variables régionalisées et leur estimation: Une application de la théorie des fonctions aléatoires aux sciences de la nature", Masson, 1965.
[7] Wahba, G., "Spline Models for Observational Data", Society for Industrial and Applied Mathematics (SIAM), 1990.
[8] http://www.dlr.de/ae/en/desktopdefault.aspx/tabid-1596/
[9] Jasa, J.P., Hwang, J.T., Martins J.R.R.A., "Open-Source Coupled Aerostructural Optimization using Python", Structural and Multidisciplinary Optimization, 2018.
[10] Chauhan, S.S., Martins J.R.R.A., "Low-Fidelity Aerostructural Optimization of Aircraft Wings with a Simplified Wingbox Model Using OpenAeroStruct", EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization, Springer, 2018.

THANK YOU FOR THE ATTENTION

Application to Static Aeroelasticity: Applied Solution

Algorithm of the OFFLINE Phase

1. Definition of the Domain.
2. Build the Reduced Base (RB) **G** and **D** with a Greedy Algorithm.
3. Improve the Kriging functions adding more points with the updated RB.
   for i = 1, ... , *(nKriging - nSamples)*
   a. while error > 5%
      i. $[\mathbf{G}^{-1}\cdot A\cdot \mathbf{G}]\cdot \boldsymbol{\Gamma} = [\mathbf{G}^{-1}\cdot b] \rightarrow F_a = f(\boldsymbol{\Gamma}) \rightarrow F_s = \mathbf{H}^{-1}\cdot F_a$
      ii. $[\mathbf{D}^{-1}\cdot K\cdot \mathbf{D}]\cdot U = [\mathbf{D}^{-1}\cdot F_s]$
      iii. error $=\max\{(\|\Gamma_o - \Gamma\|/\|\Gamma\|),(\|u_o - u\|/\|u\|)\}$
      iv. new_nodes = old_nodes + U
      v. Recompute A, b → Redo the process...
   b. Save **Γ**, U and the **i-point coordinates**.
4. Create the Kriging functions: **α** for **Γ**, **β** for U.
5. Save the Kriging functions.