Report of the internship in ONERA (Summer 2018)

STUDENT: Oriol CHANDRE VILA
TUTORS: Joseph MORLIER (ISAE-SUPAERO) and Sylvain DUBREUIL (ONERA)

August 24, 2018

Abstract

This document aims to explain the results of the internship in ONERA this summer. It has been the continuation of the Research Project performed until June 2018.

1 Introduction

1.1 The institution

ONERA [1], the French Aerospace Laboratory, is the main French research center in Aerospace and Defense, a public institution with industrial and commercial sense. Placed under the supervision of the Ministry of the Armies, ONERA was created in 1946 and nowadays approximately 2000 researchers, engineers and technicians are employed by the institution.
Its six key missions are:

- To direct and to conduct aeronautical research.
- To support the commercialization of this research by the national and the European industry.
- To construct and to operate the associated experimental facilities.
- To supply industry with high-level technical analyses and other services.
- To perform technical analyses for the government.
- To train researchers and engineers.

The current research of ONERA is designed to meet some of the challenges facing the society of today:

- To develop industrial competitiveness.
- To protect the environment.
- To enhance safety and security.

In order to achieve the goal, ONERA has been divided into several departments, distributed along the French territory:

1. DAAA - Aerodynamics, Aeroelasticity and Acoustics Dept.
2. DEMR - Electromagnetism and Radar Dept.
3. DMAS - Materials and structures Dept.
4. DMPE - Multi-physics for the energy Dept.
5. DOTA - Optics and related technique Dept.
6. DPHY - Physics, Instrumentation, Environment and Space Dept.
7. DTIS - System and Information Treatment Dept.

1.2 Internship

Enrolled within the DTIS department in the center of Toulouse, the title of the internship was: *Separated variables representation for the exploration of the Parametrized solution. Applied to a joined-wing aircraft.*

During the internship, a bibliographic study and the implementation of a case test has been performed in an approach of model reduction for real-time aeroelasticity parametric calculations. To accomplish the goal, I have understood and applied the numerical mathematical methods to an industrial problem, by using a scientific programming and collaborating with the research topic.

The final results of the internship have been achieved for an Airbus wing (see Figure 1), modifying the first approach of a joined-wing after the agreement with the tutors.

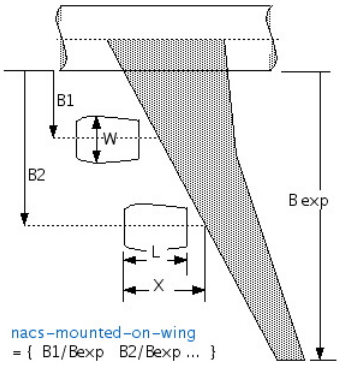

Figure 1: *Scheme of the wing used in the current parametrized study. [5]*

2 Methodology

2.1 Solver

This internship has been based on an aeroelastic code provided by ONERA. This code is divided in three phases.

The first phase is the Aerodynamic analysis, performed by using a Vortex Lattice Method (VLM). The VLM models the lifting surfaces of an aircraft as an infinitely thin sheet of discrete vortices to compute lift and induced drag. The influence of the thickness, viscosity is neglected.
Consequently, giving a geometry of a wing, the VLM solver computes the circulation, the Aerodynamics forces and the Pressure distribution.

In the second stage, a transfer matrix is computed in order to be able to compute the external structural forces from the Aerodynamic forces. This stage couples the Aerodynamics with the Structures.

Once the external loads are computed, the third stage is run. This stage uses a Finite Element Method for computing the structural behavior of the wing. The FEM approach used in this problem is the Discrete Kirchoff Triangles (DKT) of plates following an elastic linear homogeneous isotropic behavior. If the hypothesis of plane stress is applied ($\boldsymbol{\sigma}_{zz} = 0$), -in the local stiffness matrix- the membrane strain and bending strain are decoupled. The approach taken into account in this study uses the theory of the virtual work principle. For more information, see [2].
To finish with the current stage, the mesh is deformed with the results of the FEM and the Aerodynamic code is lunched again. Thus, a loop process is originated and is exited when a determined tolerance is achieved.

2.2 Parameters used

The code given by the ONERA that parametrizes the wing (see Figure 1) uses a lot of parameters. The parameters fixed (their value does not change) are the angle of attack, the Mach number, the Young modulus, the Poisson's ratio, the distance of the fuselage over the wing, the sweep angle, the dihedral angle, the number of ribs and the altitude (11000 meters).
The parameters that can change their values between a given bounds are the skin thickness, the ribs thickness, the leading edge and trailing edge spar thickness, the span and the wing surface. So, a point of interest (PoI) of our problem is defined by 6 values.

2.3 Reduced Order Model

In order to improve the computation time in a multiparametric analysis, a reduction of the problem has been considered. The technique chosen has been a Proper Orthogonal Decomposition (POD) built using a Greedy algorithm and interpolated using a kriging.

Firstly, a Greedy algorithm is used to create the reduced base POD (**V**). Here, the idea is to update the RB by adding the real solution of the worst point after solving the approach problem. In order to use a kriging to interpolate in the real time phase, no point can be added 2 times. Consequently, once a point is used in the enrichment process is automatically deleted from the list of points in which the RB will be evaluated in the following iteration. This consideration forces the number of points evaluated (candidates) to be higher than the number of iterations (samples).

Algorithm 1 POD training algorithm. [3]

```
Input: Parameters range. Here, $\boldsymbol{\mu}$ represents the parameters vector.
1. Select randomly a first sample: $\boldsymbol{\mu}^{(1)}$
2. Solve the HDM: $f(\mathbf{w}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(1)}) = 0$
3. Build a corresponding ROB: $\mathbf{V}$
4. For $i = 2, \cdots, s$
    (a) Solve: $\boldsymbol{\mu}^{(1)} = \underset{\boldsymbol{\mu} \in \{\mu_1, \cdots, \mu_C\}}{\arg\max} ||\mathbf{r}(\boldsymbol{\mu})||$
    (b) Solve the HDM: $f(\mathbf{w}(\boldsymbol{\mu}^{(i)}, \boldsymbol{\mu}^{(i)}) = 0$
    (c) Build a ROB $\mathbf{V}$ based on the samples: $\{\mathbf{w}(\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{w}(\boldsymbol{\mu}^{(i)})\}$
Return: V
```

Secondly, a kriging (or Gaussian process regression) is a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances. Under suitable assumptions on the priors, kriging gives the best linear unbiased prediction of the intermediate values. In Figure 2, squares indicate the location of the data. The kriging interpolation, shown in red, runs along the means of the normally distributed confidence intervals shown in gray. The dashed curve shows a spline that is smooth, but departs significantly from the expected intermediate values given by those means. As it can be seen in Figure 2, only one parameter is taking into account. In our problem two POD-RB has been built: one for the displacement and another one for the circulation. Consequently, two families of kriging have been created. Specifically, one 1D kriging for each sample: $\boldsymbol{\alpha}_i$ for displacement and $\boldsymbol{\beta}_j$ for circulation.
During the analysis it is necessary to note the difference between analyzing how accurate is our kriging (variance) from how accurate is our RB. In other words, our kriging can be spotless, but if it is created using a non-appropriated RB, the results cannot be accepted.

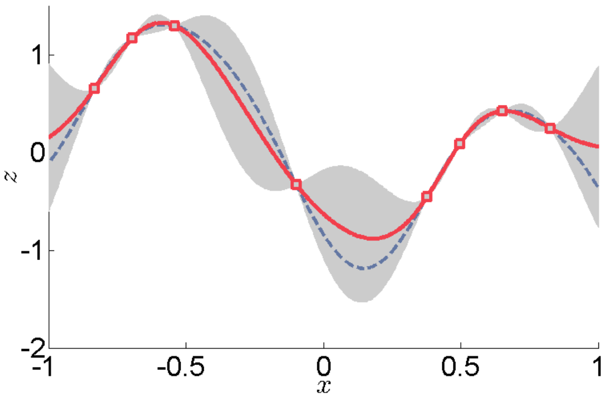

Figure 2: *Example of 1D data interpolation by kriging, with confidence intervals. [4]*

2.4 Offline phase

In the offline phase, the POD-RB is constructed (and saved) as explained before and, afterwards, the two kriging families are computed (and saved).

2.5 Online phase

In the online phase, a different script has been coded in order to be independent from the offline computation. The idea of this methodology is to perform a robust offline process in order to save the best possible results for the RB and the two kriging families (the computation time depends on the number of candidates and samples used but it can vary from several minutes to hours). After saving the offline data, the online code loads it and performs a kriging interpolation of a PoI ($\xi^{(0)}$). This process lasts in approximately 0.07 seconds, while the original calculus could last in 80 seconds.
The procedure to compute the approximative solution for the displacement and the circulation are as follows. Since it is the same procedure, only the nomenclature of the displacement calculation is showed.

$$\hat{\boldsymbol{u}}(\xi^{(0)}) = \sum_{i=1}^{nSamples} \hat{\boldsymbol{\alpha}}_i(\xi^{(0)}) \cdot \mathbf{U}_i^V \tag{1}$$

Since the $\hat{\boldsymbol{u}} \sim \mathcal{N}(\mu_{\hat{u}}, \sigma^2_{\hat{u}})$, both the mean and the variance have to be computed:

$$\begin{cases} \mu_{\hat{u}} = \sum_{i=1}^{nSamples} \mu_{\hat{\alpha}_i} \cdot \mathbf{U}_i^V \\ \sigma^2_{\hat{u}} = \sum_{i=1}^{nSamples} \sigma^2_{\hat{\alpha}_i} \cdot (\mathbf{U}_i^V)^2 \end{cases} \tag{2}$$

Here, $\mathbf{U}^V$ is the reduced base for the displacement.

2.6 Error identification

In this section, it is sought to identify possible sources of errors in the code proposed. On one hand, it exists the error due to the base truncation in the POD approach. On the other hand, one can identify the error of interpolation occurring in the kriging.

Consequently, one can drive these errors by modifying the number of samples: higher the number of samples used, more important influence of the kriging process into the error. But, lower the number of samples, more influence of the POD process.
Nevertheless, one could improve the error due to kriging by adding more points to the functions once the Offline process would be already calculated. This should last with a reduction on the error without deteriorating the computation time.

3 Results

In order to verify and to validate the POD procedure, different cases for the offline code (see Table 1) and different configurations for the online code (see Table 2) have been analyzed.

Table 1: *Cases analyzed for the offline procedure.*

ID	# Samples	# Candidates
1	5	10
2	5	50
3	5	100
4	15	30
5	15	50
6	15	100
7	25	30
8	25	50
9	25	100
0	65	70

Table 2: *Configurations analyzed for the online procedure.*

Configuration	*1*	*2*	*3*	*4*
Skin thickness (m)	0,028	0,029	0,030	0,031
Ribs thickness (m)	0,009	0,014	0,010	0,015
LE spar thickness (m)	0,017	0,019	0,022	0,025
TE spar thickness (m)	0,009	0,013	0,011	0,015
b (m)	64,75	66,83	73,20	75,65
S (m^2)	443,56	449,70	444,44	500,21

In order to do the analysis, for each configuration and case several aspects must be ensured. The idea is to verify that the real value is comprised between the two variance values (see Figures 3, and 4). Furthermore, it has been verified that the node where Von Mises stress is maximum is the same for all the cases.
In Figure 4, an artificial deformation ratio has been applied to give the deformed wing image.

From the these figures, some comments must be done. Firstly, while the real value of the circulation is comprised

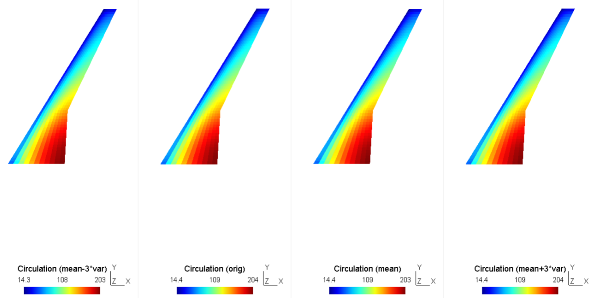

Figure 3: *Circulation comparison for Conf 1 and Case 10.*

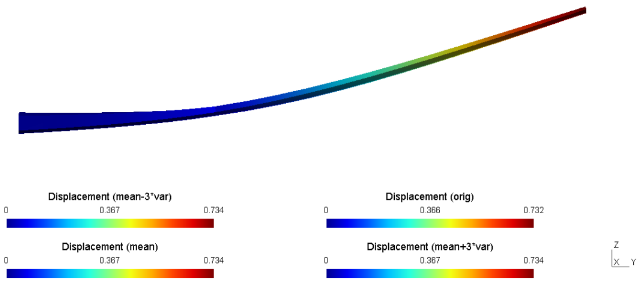

Figure 4: *Displacement comparison for Conf 1 and Case 10.*

between the value of the two variance, for the displacement this does not happen in any case evaluated. Secondly, if a comparison is performed between the POD solution and the *real* solution, the error computed is usually less than a 15%. The only value that the POD cannot predict in an acceptable accuracy is the minimum value of the equivalent Von Mises stress.

To conclude the study of the POD-RB, five random configurations have been tested for each case and compared with the original code results. This study drives to the same conclusions: ***(a)*** using only 5 iterations in the Greedy algorithm is not enough despite the fact that the results are not truly inaccurate; ***(b)*** the method is not able to well approximate the inferior limit in the Von Mises stress, but it is able to approximate highly reliable both maximum and average stress; ***(c)*** the results are a lot better than the ones achieved with the previous analysis (which were limit cases); and, ***(d)*** in the displacement calculation, the real value is normally out of the variance bounds, but the values are not distanced.

4 Conclusions

After the internship, several conclusions can be exposed.

First of all, it can easily be proved that a real time process saves a great amount of computation time and efforts once an offline process is performed before. However, it will be interesting to be applied in a problem if only a lot of cases must be simulated. If not, it does not worth to perform such an expensive offline calculus.

Secondly, the parameter that drives the accuracy of the RB is the number of samples used to create it (as it was expected). If more samples are computed, the approximation will use more modes of the problem, so the difference between the approximation and the real solution is smoother. Consequently, the number of candidates increases the computation time and adds more information to the kriging (more points to interpolate). Furthermore, because of the kriging process, they must be more candidates than samples ($nCandidates \geq nSamples$).

Finally, the kriging procedure is giving an acceptable variance values, but the real solution is not localized between the limits (in the case of the displacement). This can only be explained by a lag of accuracy of the POD-RB. A source of error could be the construction of the RB. Since it is a coupled problem, the real solution added to the RB could not reduced the error to zero. For sure, the error in that point is reduced, but the coupling of the problem can produce a non-zero error indicator. In the original Greedy algorithm, this problem does not exist because the same point can be added several times.

In future works, the idea of working on the different source of error to achieve more accurate results should be studied by improving the creation of the reduced base and by possibly adding points to the kriging functions.

For more information see the internship's GitHub or go to the next *url*: `http://github.com/mid2SUPAERO/Stage_CHANDRE_POD`.

References

[1] ONERA website (in English): *url*: `http://www.onera.fr/en`.

[2] De Soza, T, "Eléments de plaque: modélisations DKT, DST, DKTG et Q4g", Manual de référence Code Aster, May 2011.

[3] 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.

[4] Wahba, G., "Spline Models for Observational Data", Society for Industrial and Applied Mathematics (SIAM), 1990.

[5] Geometric Specifications (in English): *url*: `http://www.lissys.demon.co.uk/pug/c03.html`.