UTOPIAE LTW-1 EXERCISE

Robust aerodynamic design building blocks

The objective of the following two exercises is to provide an example of the basic elements of a robust aerodynamic design procedure. The first exercise concerns the aerodynamic part of the problem and illustrates an approach to the parametrization of a wing profile and the use of a simple fluid dynamic analysis program. The second one focuses on the calculation of risk functions and their use in the comparison of two different solutions.

Exercise 1: Flow solver and geometry handler usage

Use the bash script 'run_xfoil.sh' to handle the process of parametric airfoil shape generation and flow solver run. Please refer to the provided instruction manual to understand input and output of the procedure.

In particular, try to solve the following tasks:

1. Check the effects of different parameters and working conditions on the aerodynamic and geometric characteristics of the generated airfoil.
2. Test the effect of a single shape parameter, let say W(1) or W(4) on the airfoil shape (suggestion, if the 'plot.ps' illustration is not enough, you may use the -dbg flag to access the file containing the modified airfoil ('modified_airfoil.dat'). If you run the code with all W(i)=0, you should obtain the NACA2412 wing section.
3. Set the working conditions as follows: CL=0.5, MACH=0.1, Reynolds_number = 500000, XTRLO = 0.95, XTRUP = 1.0 and MAX_THICK = 0.12. Compute the NACA2412 performance without the effect of uncertainties.
4. Build a script (in a language of your choice, like bash, python, C or fortran) to perform a Monte Carlo sampling of the baseline airfoil CD considering U as the vector of uncertain parameters. Choose the uniform variation range of the elements of U so that the variation range of CD is roughly around 5% of the baseline CD value.
5. Repeat the same Monte Carlo sampling for different airfoils (using W) and save the results (use the variation rage computed in the previous step).

Exercise 2: Risk functions evaluation

Use the R script 'adg.risk_fnct.R' to evaluate VaR and CVaR after a Monte Carlo sampling. Refer to the provided instruction manual to understand input and output of the procedure.

In particular, solve the following tasks:

1. Apply the procedure to the provided 'DB1.des' database file to compute VaR and CVaR of the objective function 'OB_00000' at –alpha=0.9 without confidence intervals. Visualize the related empirical distribution function reported in the 'Rplots.pdf' file.
2. Compute the confidence intervals of VaR and CVaR for the quantity of interest 'OB_00000' in 'DB1.des' and –alpha=0.9, and test the effect on the results of different bootstrap iterations.
3. Consider subsamples of 'DB1.des' database of decreasing size and check the effect on VaR, CVaR and the related confidence intervals (Suggestion: launch several times the script 'adg.risk_fnct.R' with the –subsample flag set at, let say, 10 and produce a plot of the variation of VaR and CVaR).
4. Repeat operation 1 on the provided 'DB2.des' database and compare the results.
5. Repeat operation 2 on 'DB2.des' and check if the confidence intervals overlap.

6. Apply simultaneously to ‘DB1.des’ and ‘DB2.des’ the operation described in 3 with a subsample size of 15, register the results and repeat the procedure several times. At the end of the process, count the times that the comparison of VaR and CVaR obtained with the random subsampling procedure is equal to the comparison result given by the full-sized databases. For the sake of simplicity, you can avoid considering confidence intervals.
7. OPTIONAL task: When an evaluation of the objective function has the ERROR flag set to 1, a penalty must be applied. Indeed, the wrong result cannot be simply excluded from the Monte Carlo sampling without altering the MC distribution quality. On the other hand, assigning a very high penalty can alter the values of risk functions (especially CVaR) in an unbalanced way. Figure out a sound strategy to assign a penalty to the objective without introducing too a heavy bias (use file ‘DB2E.des’ instead of ‘DB2.des’).