

A guide for reproducibility of results

The methodology presented in the article entitled “A visual sensitivity analysis for parameter-augmented ensembles of curves” (authored by A. Ribés, J. Pouderoux, and B. Iooss; published in the *Journal of Validation Verification and Uncertainty Quantification*) has been implemented in ParaView, an open source software that can be downloaded here:

<https://www.paraview.org/download/>

In order to facilitate the reproduction of the results, we have written this short document: “A guide for reproducibility of results”. In the article three datasets illustrate our claims. Two synthetic examples and one industrial use-case, we named them:

1. Oscillating tangents
2. Campbell 1D functions
3. A hydraulics study-case

We include raw data, python scripts and detailed steps on how to reproduce the images of the article for these 3 cases. One demonstration video, in mp4 format, is available per dataset. The scripts were tested in ParaView version 5.4.1, which contains all diagrams and functionalities necessary to the reproduction of the figures of the article. Newer versions of ParaView should also support these functionalities.

We provide a python script for each dataset that can be executed in ParaView by use of its python shell: “Tools → Python Shell” makes a window appear, then click on button “Run Script” and select the python script.

Oscillating tangents

This synthetic example consists of an ensemble of time-oscillating analytical functions coming from an equation presented in the article. Four files are provided for this use case:

- oscillating_tangents_input_parameters.csv
A CSV file containing 400 sets of input parameters
- oscillating_tangents_output_curves.csv
A CSV file containing 400 output curves
- oscillating_tangents.py
A python script that can be read by ParaView and loads the data in the appropriate diagrams
- oscillating_tangents.R
An R script is provided that can be run in order create different CSV files. They can be produced with a number of curves, and sets of inputs parameters, chosen by the user (400 by default)

Steps:

1. Run "oscillating_tangents.py"
(the path to the CSV files should probably be changed to match the current location of the CSV files containing the curves and input parameters)
2. In the BagChartView perform selections (using for instance "Rectangle Selection", a button in the border of the view) in order to see the correspondence between clusters in the PCA plane and curves in the "FunctionalBagChartView".

Campbell 1D functions

This synthetic example is inspired by the articles [Campbell et al., 2006], [Marrel et al., 2011] and consists of an ensemble of analytical functions that evolve in time. This dataset is also generated by use of an equation presented in the article. Four files are provided for this use case:

- campbell_1d_input_parameters.csv
A CSV file containing 400 sets of input parameters
- campbell_1d_output_curves.csv
A CSV file containing 400 output curves
- campbell_1d.py
A python script that can be read by ParaView and loads the data in the appropriate diagrams
- campbell_1d.R
An R script is provided that can be run in order create different CSV files. They can be produced with a number of curves, and sets of inputs parameters, chosen by the user (400 by default)

Steps:

1. Run "campbell_1d.py"
(the path to the CSV files should probably be changed to match the current location of the CSV files containing the curves and input parameters)
2. First we will show all 400 curves in the "FunctionalBagChartView":
 - a. Click on the view and to "ExtractBagPlots" filter in the Pipeline Browser
 - b. Go to Properties tab and scroll down to see all variables
 - c. Select all curves by clicking on top left of "Series Parameters"
 - d. All curves should be now visible in your "FunctionalBagChartView".
3. By use of a selection tool (for instance "Rectangle Selection", a button in the border of the view) chose the curves ending with the highest values. They should be highlighted.
4. Select "CSVReader1", the CSV reader containing the values of input parameters, on the Pipeline Browser.
5. Now it is possible to use "Tools -> Link with Selection" in order to associate the current selected curves with the input parameters.
6. A "Selection Link Mode" window should pop up, click ok.
7. The Parallel Coordinates View should now highlight the parameters corresponding to the selected curves.
8. Once a link selection is performed the user can continue interacting with the data and the selections will be propagated. The action "Link with Selection" is only needed once.

A hydraulics study-case

This study-case concerns a maritime model of Alderney Ray (or Raz Blanchard in French), which is a strait that runs between Alderney (UK) and Cap de la Hague (France), a cape at the northwestern tip of the Cotentin peninsula in Normandy.

This case is described in the article, three files are provided:

- hydraulics-study-case-input-parameters.csv
A CSV file containing 1500 sets of input parameters
- hydraulics-study-case-output-curves.csv
A CSV file containing 1500 output curves
- hydraulics-study-case.py
A python script that can be read by ParaView and loads the data in the appropriate diagrams

Steps:

1. Run “hydraulics-study-case.py”
(the path to the CSV files should probably be changed to match the current location of the CSV files containing the curves and input parameters)
2. By use of a selection tool (for instance “Rectangle Selection”, a button in the border of the view) chose the highest values of the SEALEVEL parameter in the Parallel Coordinated View. They should be highlighted.
3. Select “ExtractBagPlots1”, the filter which output contains the statistical analysis of the curves, on the Pipeline Browser.
4. Now it is possible to use “Tools -> Link with Selection” in order to associate the current selected input parameters with the output curves.
5. A “Selection Link Mode” window should pop up, click ok.
6. The “FunctionalBagChartView” should now highlight the output curves corresponding to the selected values of SEALEVEL.
7. Once a link selection is performed the user can continue interacting with the data and the selections will be propagated. The action “Link with Selection” is only needed once.
8. Other values (low, middle) of SEALEVEL can be selected to verify the strong influence of this parameters on the outputs. Other parameters can be tested such as CF1 or CF2 in order to check their lack of influence. Other behaviors described on the article can be verified.