PRACTICAL SESSION - USING THE MYSTIC FRAMEWORK

Merlin Keller - Jérôme Stenger

June 4, 2018

ETICS 2018 - Roscoff

EDF - R&D

3 DIFFERENT OPTIMIZATION METHODS

- Auxiliary files
  - Model.py: Contains the hydraulic physical model and the cost function that calculates the p.o.f of the quantity of interest.
  - Solver_Simulated_Annealing.py: Contains our personal implementation of a simulated annealing solver. 4 schedules are available.
  - Cross_Visualisation.py: Visualisation of the results imported from the three script files.
- Script files
  - Script_Simulated_Annealing.py: Optimization of the cost function using the simulated annealing solver.
  - Script_Differential_Evolution.py: Optimization of the cost function using a differential evolution solver.
  - Script_Mystic_Framework.py: Optimization of the p.o.f using the mystic framework.

COST FUNCTION

- → 4 parameters Q, K_s, Z_v, and Z_m bounded and constrained only by their mean. $\mu_k = \omega_k \delta_{x_{1,k}} + (1-\omega_k)\delta_{x_{-1,k}}$,
- → Our cost function returns the CDF $P(f(Q, K_s, Z_v, Z_m) \leq h)$,
- → Parameterization with the position of the two Dirac masses constituting the support of the measure.

$$\begin{aligned} f(x_{\pm 1,1}, \ldots, x_{\pm 1,p}) &= \sum_{i_1 \in \{-1,1\}} \cdots \sum_{i_p \in \{-1,1\}} P(G(x_{i_1,1}, \ldots, x_{i_p,p}) \leq h)\,, \\ &= \sum_{i_1 \in \{-1,1\}} \cdots \sum_{i_p \in \{-1,1\}} \omega_{i_1,1} \ldots \omega_{i_p,p} 1_{\{G(x_{i_1,1},\ldots,x_{i_p,p}) \leq h\}}\,. \end{aligned}$$

Where $x_{-1,k}$ et $x_{1,k}$ belongs to $[l_k, E(\mu_k)]$ et $[E(\mu_k), u_k]$.

ROBUST QUANTIFICATION

The files provided treat the 1D case, the others parameters are set to their means:

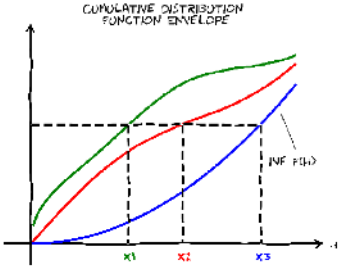
$$K_s = 30, \ Z_v = 05, \ Z_m = 54.5$$

The CDF is globally optimized:

$$\overline{q}(h) = \inf_{\substack{Q \in \mathcal{M}_1([160,3580]) \\ E(Q)=1870}} P(f(\mu) \leq h)$$

The plot shows the lowest CDF:

$$\overline{q} : h \mapsto \overline{q}(h)$$



STOCHASTIC OPTIMIZATION

The following scripts optimize the cost function either with a Simulated Annealing solver or a Differential Evolution solver.

→ Script_Simulated_Annealing.py

→ Script_Differential_Evolution.py

The last script Script_Mystic_Framework.py also optimizes with a Differential Evolution solver ...

THE MYSTIC FRAMEWORK

- → Mystic allows a really easy implementation of constraints of any order, the framework is build for OUQ.
- → We set the number of Dirac masses required for every measure.The cost function will depend on the position AND the weight.
- → Every new point generated is transformed to respect the constraint evaluation in the solver:
  - → Mean: it translates the measure. Variance-preserving.
  - → Variance: Scales the measure, then re-adjust the mean. Mean-preserving.
  - → And so on for higher moment orders.
- → Mystic automatically calculates the p.o.f of our model.