

Probabilistic analyses of a spatial vehicle equipment bay

Study in partnership with CNES



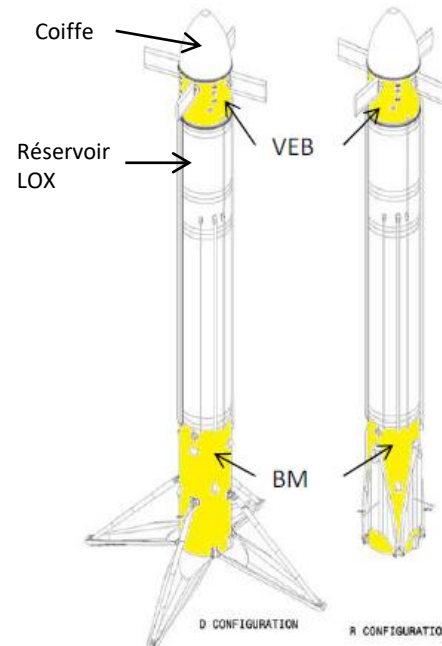
Antoine Dumas

5 June 2020

Contexte : projet CNES Callisto

Callisto – Dimensionnement d'un lanceur réutilisable

- Etude de la case à équipements



VEB : Vehicule Equipement Bay

Contexte : projet CNES Callisto

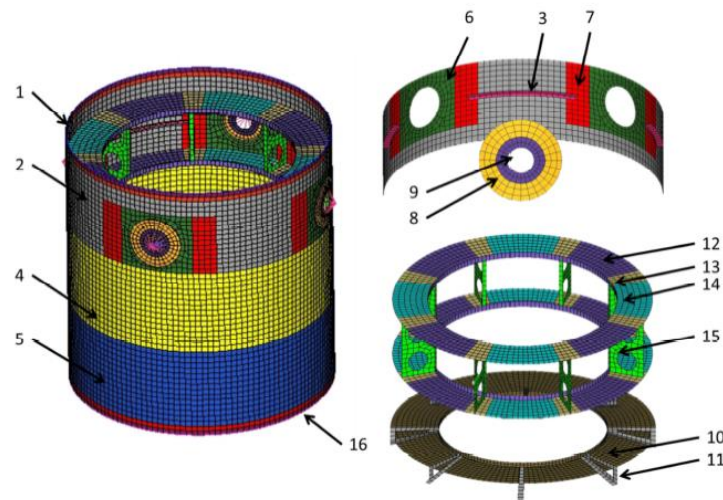
Etude de dimensionnement de la case à équipements

- Objectif : application de la méthodologie incertitude afin de minimiser la masse de la case à équipements
 - Analyse de sensibilité (Morris et Sobol)
 - Analyse de fiabilité
 - Analyse de sensibilité fiabiliste
 - Optimisation sous contraintes de fiabilité (pas encore réalisée)

Contexte : projet CNES Callisto

☐ Implémentation numérique

- Code EF Nastran fourni par le CNES
 - Analyse statique et en flambage
 - Temps de calcul unitaire ~ 5 min
 - Variables d'entrée aléatoires : 47 au total
 - 20 épaisseurs
 - Module d'Young
 - 26 valeurs de chargement
 - Variables de sorties : 5 x 10 cas de charge
 - 2 contraintes, Flux, Déplacement, Valeur propre (flambage)
- Couplage du modèle EF avec Python et utilisation des modules classiques
 - OpenTURNS, otmorris, otwrapy, Numpy, Pandas

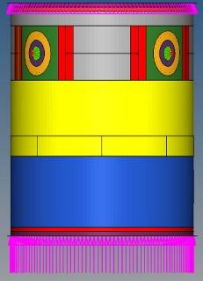
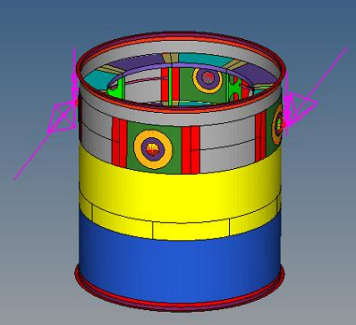
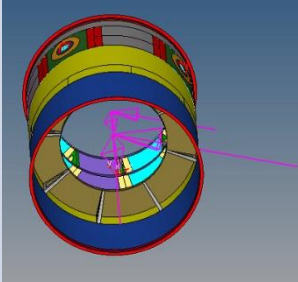


Callisto : chargement



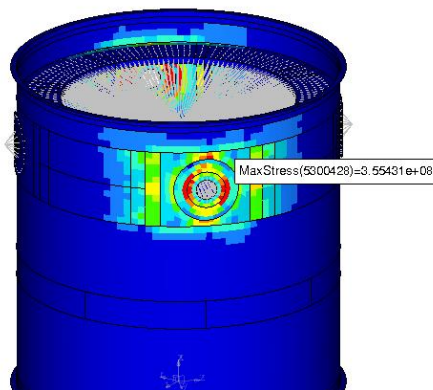
10 cas de charge

- Plusieurs types d'efforts

Efforts généraux	Efforts aérodynamiques (FINs)	QSL
<p>Upper Interface Clamped</p>  <p>Lower Interface Loaded in compression</p>	<p>Fins loaded in opposite direction</p> 	<p>QSL applied in the same direction At both nodes</p>  <p>Upper direction for the vertical QSLs 135° from fin 2 for the lateral QSLs</p>

Résultats : contraintes / flambage

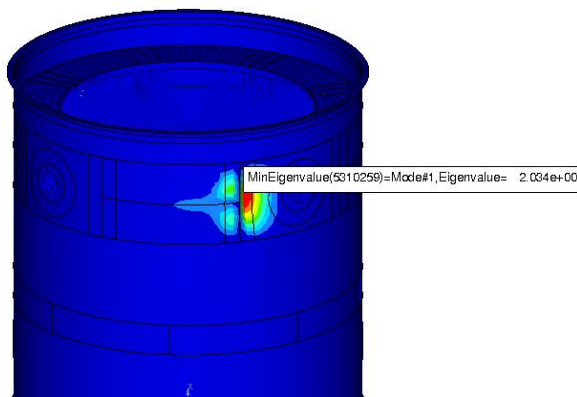
Contour Plot
Stress(vonMises, max)
Analysis system
3.554E+08
3.159E+08
2.764E+08
2.370E+08
1.975E+08
1.580E+08
1.185E+08
7.898E+07
3.949E+07
0.000E+00
No result
Max = 3.554E+08
Shell 5300428
Min = 0.000E+00
Shell 5230246



Contrainte maximale

Loadcase	Element	MAX_Stress VM_Z1	Element	MAX_Stress VM_Z2
150303	5300258	323.9043	5300428	355.431

Contour Plot
Displacement(Mag, min)
Analysis system
3.044E-02
2.706E-02
2.368E-02
2.030E-02
1.691E-02
1.353E-02
1.015E-02
6.765E-03
3.383E-03
0.000E+00
No result
Max = 3.044E-02
Node 5310259
Min = 0.000E+00
Node 5009000

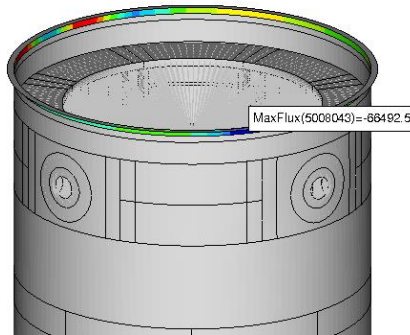


Valeur propre minimale

Loadcase	Node	MINEigenvalue
150303	5310259	2.034

Résultats : flux I/F supérieur et déplacement

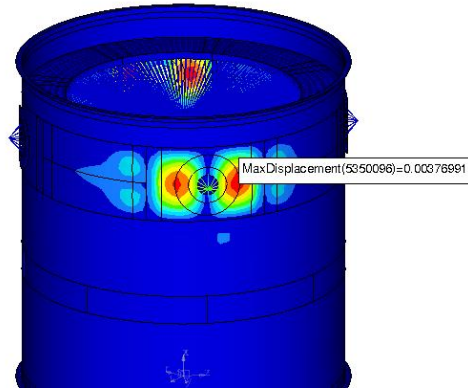
Contour Plot
2D Element Force (XX, min)
Analysis system
1.732E+04
8.009E+03
-1.304E+03
-1.062E+04
-1.993E+04
-2.924E+04
-3.855E+04
-4.787E+04
-5.718E+04
-6.649E+04
No result
Max = 1.732E+04
Shell 5008122
Min = -6.649E+04
Shell 5008043



Flux I/F Maximal

Loadcase	Element	MAX_IF-Flux	Element	MIN_IF-Flux
150303	5008122	17.32	5008043	-66.49

Contour Plot
Displacement (Mag, min)
Analysis system
3.770E-03
3.351E-03
2.932E-03
2.513E-03
2.094E-03
1.676E-03
1.257E-03
8.378E-04
4.189E-04
0.000E+00
No result
Max = 3.770E-03
Node 5350096
Min = 0.000E+00
Node 5009000



Déplacement maximal

Loadcase	Node	MAX_Displacement
150303	5350096	3.76

Couplage du code EF

Couplage avec Python et OpenURNS

- Le couplage s'effectue en 3 parties
- Création du fichier d'entrée à partir d'un fichier modèle : `ot.coupling_tools.replace`
 - 2 fichiers « modèles » créés à partir des vrais fichiers
 - Les variables d'entrée sont repérées à l'aide d'une balise (ici entre les #)

```
$HNAME PROP                    5702"PSHELL_5702" 4
$HWCOLOR PROP                  5702      54
PSHELL      5702      5000#Actuator_Outer_Doubler# 5000      5000      0.0
$-----1$-----2$-----3$-----4$-----5$-----6$-----7$-----8$-----9
$PSHELL      PID      MID1      T      MID2 12I/T**3 MID3      TS/T      NSM
$-----1$-----2$-----3$-----4$-----5$-----6$-----7$-----8$-----9
$
$HNAME PROP                    5712"PSHELL_5712" 4
$HWCOLOR PROP                  5712      20
PSHELL      5712      5000#Actuator_Inner_Doubler# 5000      5000      0.0
$-----1$-----2$-----3$-----4$-----5$-----6$-----7$-----8$-----9
$PSHELL      PID      MID1      T      MID2 12I/T**3 MID3      TS/T      NSM
$-----1$-----2$-----3$-----4$-----5$-----6$-----7$-----8$-----9
$
$HNAME PROP                    5312"PSHELL_5312" 4
$HWCOLOR PROP                  5312      30
PSHELL      5312      5000#Upper_Fins_Cyl_b# 5000      5000      0.0
```



Les cartes Nastran accepte
8 caractères par défauts

```
{:0.3f}
{:0.2f}
{:0.2f}
```


Couplage du code EF

Couplage avec Python et OpenTURNS

- Le couplage s'effectue en 3 parties
1. Exécution de Nastran : [ot.coupling_tools.execute](#)
 - `nastran GLA150_SOL_200_May_19_Min_Tech_1.dat ifpstar=no scr=yes memory=200MW buffsize=16385`
 3. Lecture des résultats dans le fichier de sortie `gla150_sol_200_may_19_min_tech_1.f06`
 - Utilisation d'un script python + [ot.coupling_tools.get](#)

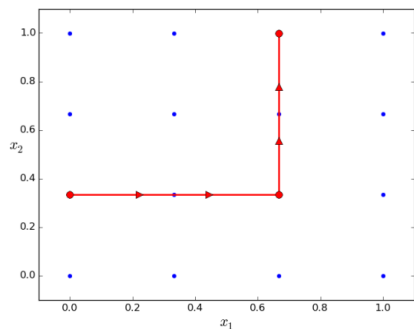
+ utilisation du module `otwrapy`

- Gestion d'un dossier de travail temporaire
- Parallélisation

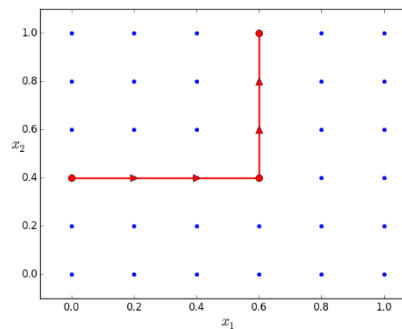
Analyse de Morris

☐ Analyse de Morris pour réduire la dimension

- Utilisation du module otmorris
- Permet d'identifier les variables influentes
 - Effet linéaire sans interactions
 - Effet non linéaire ou avec interactions
- Principe
 - Répétitions (trajectoire) de plans d'expériences *un-à-la-fois* sur une grille découpée en un nombre de niveaux prédéfinis
 - Calcul statistique (moyenne, écart-type) des effets élémentaires



$nb_{niveaux} = 4$



$nb_{niveaux} = 6$

Analyse de Morris

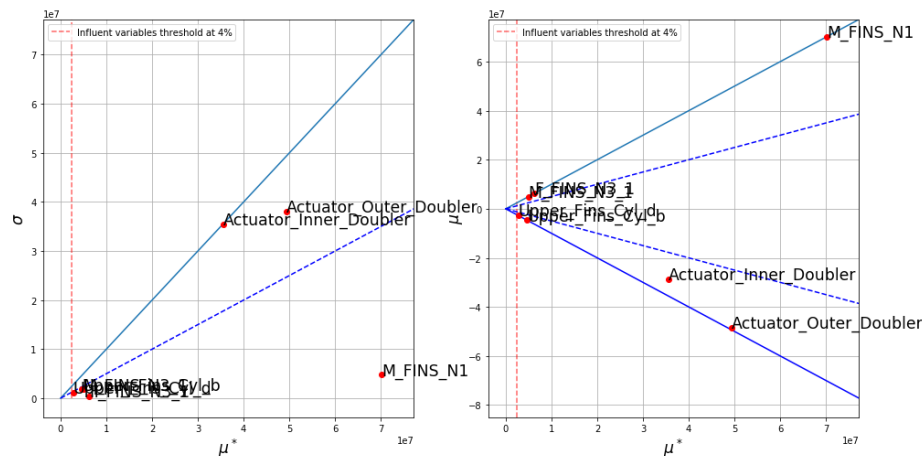


Application à Callisto

- 6 niveaux et 15 trajectoires : 720 évaluations
- Un maximum de 15 paramètres sont sélectionnés par sortie
- En cumulé : 28 variables sont influentes sur au moins 1 sortie
- Comportement des sorties :
 - monotone et linéaire
 - monotone et non linéaire

	MAX_Stress_VM_Z1_1	MAX_Stress_VM_Z2_1	MAX_IF_Flux_1
0	Actuator_Outer_Doubler	Actuator_Outer_Doubler	Actuator_Outer_Doubler
1	Actuator_Inner_Doubler	Upper_Fins_Cyl_b	Actuator_Inner_Doubler
2	Upper_Fins_Cyl_b	Upper_Fins_Cyl_d	Upper_Fins_Cyl_b
3	Upper_Fins_Cyl_d	F_FINS_N3_1	F_FINS_N3_1
4	F_FINS_N3_1	M_FINS_N1	M_FINS_N1
5	M_FINS_N1		M_FINS_N3_1
6	M_FINS_N3_1		

Morris indices for MAX_Stress_VM_Z1_3

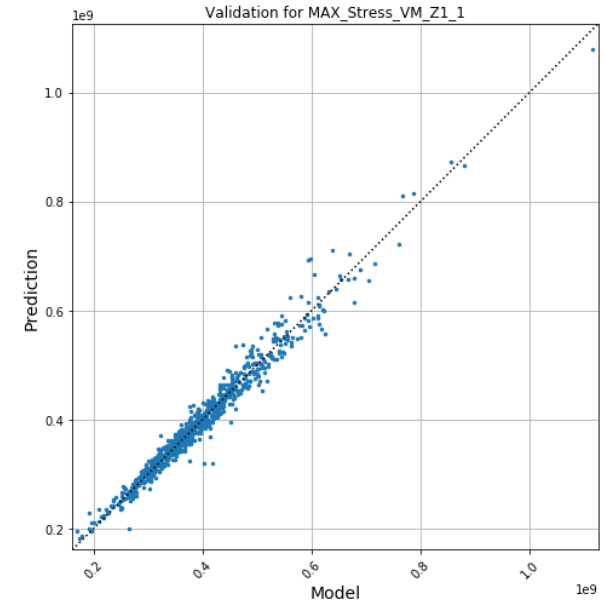


Construction d'un métamodèle

Utilisation du chaos polynomial

- Construction d'un **chaos polynomial** spécifique par sortie
 - Sélection des paramètres d'entrée influents
 - Construction d'un chaos creux avec **sélection automatique du meilleur chaos** parmi les degrés 1, 2 ou 3 via le critère du Q2 analytique
 - Agrégation de tous les chaos** pour manipuler une seule fonction
 - Élargissement du domaine de définition des variables aléatoires

R2 (test) et Q2 minimale de 0,8 sur les 50 sorties la majorité > 0,9



```
composed_model_collection[i] = ot.ComposedFunction(chaos_model_collection[i], transformation_model_collection[i])
self.full_metamodel = ot.AggregatedFunction(composed_model_collection)
self.full_metamodel.setOutputDescription(output_sample.getDescription())
```

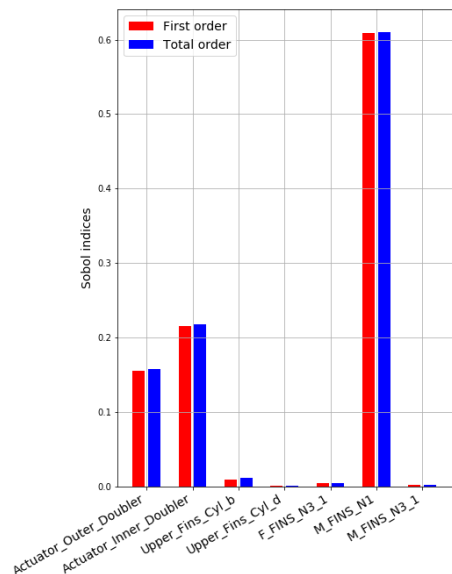
Analyse de sensibilité globale

Indices de Sobol issus du chaos polynomial

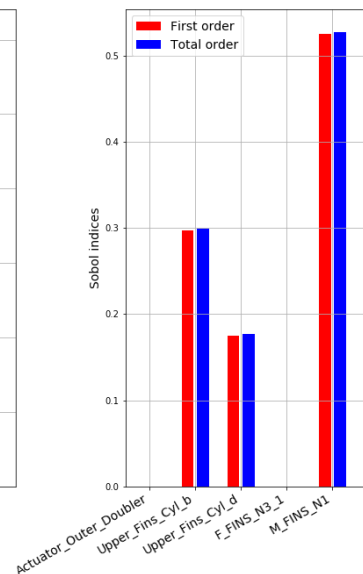
- Indices définis uniquement en fonction des entrées influentes selon l'analyse de Morris
- Permet une optimisation « manuelle »
 - Diminuer les épaisseurs non influentes

Input variable	MAX_Stress_VM_Z1_1		MAX_Stress_VM_Z2_1		MAX_IF_Flux_1		MAX_Displacement_1	
	First order	Total order	First order	Total order	First order	Total order	First order	Total order
Inertia_Box_Web							0	0
Inertia_Box_Tray_a								
Lower_Cylinder_5							0	0
Upper_Fins_Cyl_a								
Upper_Cyl_RCS_a								
Actuator_Outer_Doubler	0.155	0.158	0	0	0.246	0.246	0.01	0.01
Actuator_Inner_Doubler	0.215	0.218			0.015	0.015		
Upper_Fins_Cyl_b	0.009	0.011	0.297	0.299	0.004	0.004	0.178	0.18
Upper_Cyl_c							0.014	0.014
Inertia_Box_Tray_b								
Inertia_Box_Tray_c								
Upper_Cyl_RCS_b								
Upper_Fins_Cyl_d	0.001	0.001	0.175	0.177			0.021	0.022

Sobol indices for MAX_Stress_VM_Z1_1



Sobol indices for MAX_Stress_VM_Z2_1



Analyse de fiabilité système

Analyse pour le cas de charge 1

- Événement système (valeurs seuils fictives)
 - Max_Stress_VM_Z1 > 4.2e8 U
 - Max_Stress_VM_Z2 > 4.56e8 U
 - Max_IF_Flux > 8.3e5 U
 - Max_Displacement > 2.8e-3 U
 - Min_EigenValue < 1.55
- Possible directement avec OpenTURNS depuis 1.14

UnionEvent

```
class UnionEvent(*args) ¶
```

Event defined as the union of several events.

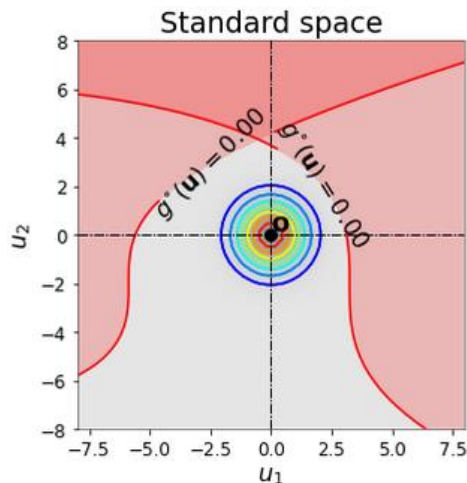
An occurrence of one single event E_i yields the occurrence of the system event (series system):

$$E_{sys} = \bigcup_{i=1}^N E_i$$

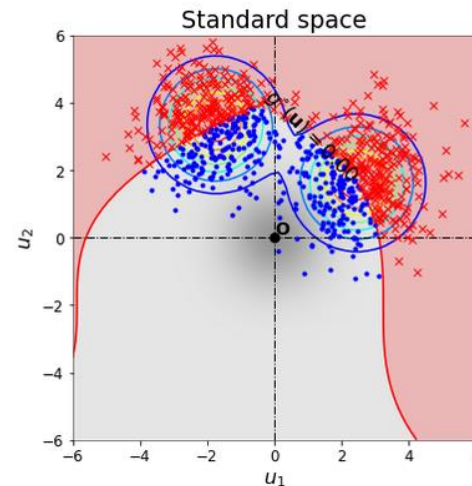
Analyse de fiabilité système

Utilisation du chaos polynomial « étendu »

- Estimation de la probabilité système
 - simulation de Monte Carlo
 - FORM + tirages d'importance centré sur tous les points P^*
- Estimation des probabilités unitaires
 - Permet de déterminer quel événement est le plus prépondérant



Exemple avec 2 événements



Tirages d'importance à partir d'une mixture

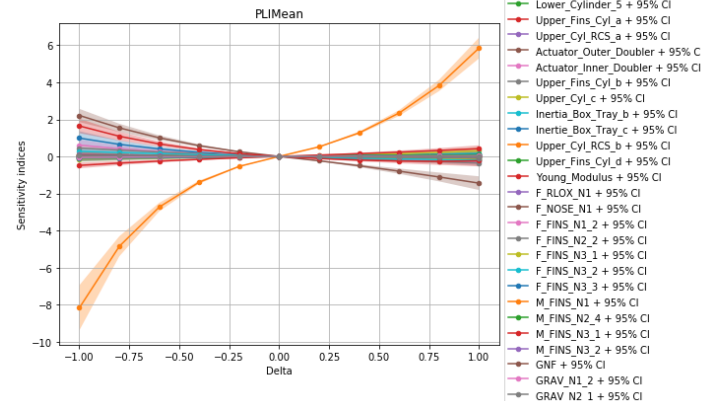
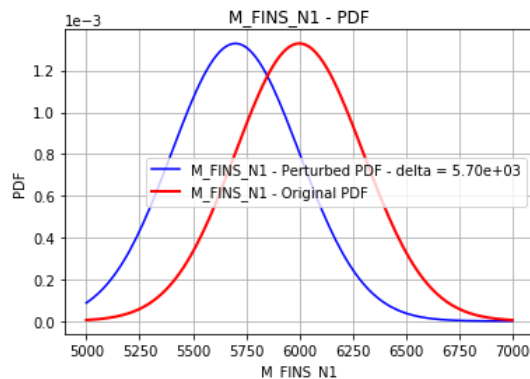
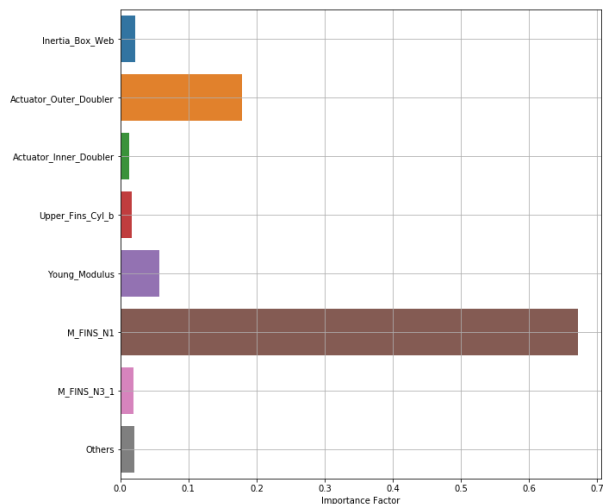
Analyse de sensibilité fiabiliste

Facteurs d'importance

- Obtenus à partir de la simulation de Monte Carlo

Indice PLI (Perturbation Law Indices)

- Lois initiales gaussiennes
 - Modification de la moyenne : $\pm \delta \times \sigma$
- Permet de voir l'évolution de la probabilité si on modifie la moyenne des lois initiales



Conclusion et suite

Application de la méthodologie incertitude

- Possible entièrement avec OpenTURNS

Suite

- Evaluation de la probabilité système pour tous les cas de charges
 - Méthode identique ?
 - Méthode adaptative telle que AK-Sys ?
- Minimisation du poids sous contraintes de fiabilité
 - Choisir les paramètres de conception à optimiser
 - Utilisation des librairies d'optimisation intégrées à OpenTURNS