EADS Innovation Works

# Example of use of Polynomial Chaos with OpenTURNS 0.13.2

*Interests & Limitations*

Jayant SEN GUPTA - jayant.sengupta@eads.net

June, 7th, 2011

EADS

# Outline

# Place in the lifecycle of an A/C

## Different phases for the design of an A/C

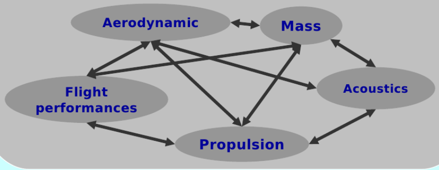
**EARLY DESIGN of an AIRCRAFT**

**Input parameters (geometry, engine performances, ...)**

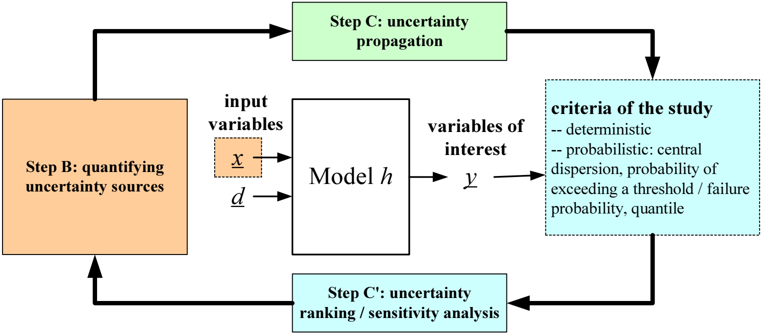
- Fuselage for the passengers: section, length cabin, number of passenger...
- Wings, tailplane, engines for the performances

**Simulation**

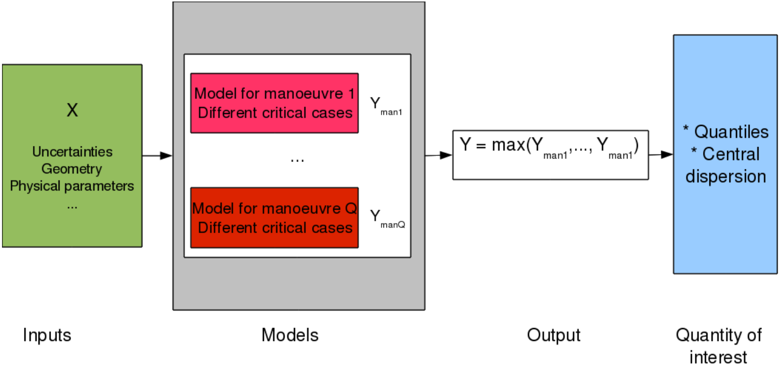
Take off, mission, landing, special operations, operational costs, ...



# Methodology for uncertainty propagation



# Step A: description of the study




## Model

- Complex calculation
- Different scenarios (corresponding to different calculations)
- Distant server, impossible to install OpenTURNS on it

# Step A: description of the study

## Inputs

- Geometrical parameters
- Physical parameters (Hooke tensor, damping, ...)
- Around 10 input parameters

## Outputs and quantity of interest

- Quantiles on different loads

$$Load^{(m)}(x) = \max_{c \in \mathcal{C}} \left( Load_c^{(m)}(x) \right) \tag{1}$$

$$Load(x) = \max_{m \in \mathcal{M}} \left( Load^{(m)}(x) \right) \tag{2}$$

# Step B: quantification of the uncertainty

### Involvement of Airbus experts

- Workshop organized
- Parametric uncertainty

### Quantification

- Mostly normal distribution (eventually truncated)
- Independence of each inputs

# Step C: uncertainty propagation

## Sampling strategy

- OpenTURNS not installed on the calculation server
- Create a sample on a computer with OpenTURNS
- Send the sample on the server
- Automate the computation
- Send back the output database for analysis

## Quantiles

- Empirical quantiles
- Wilks formula

## Sensitivity analysis

- Very important for design

# Outline

# Surrogate model building

## PCE on the model of each manoeuvre

- Build a surrogate model of $Load(x)$

$$Load(x) = \max_{m \in \mathcal{M}} \left( Load^{(m)}(x) \right) \tag{3}$$

- Build a surrogate model for each $Load^{(m)}(x)$

$$Load^{(m)}(x) = \max_{c \in \mathcal{C}} \left( Load_c^{(m)}(x) \right) \tag{4}$$

- Because of reduction dimension, second option is chosen

# How to build a Polynomial Chaos Expansion?

Script

```
# Create Polynomial Chaos Expansion of outVar = g(Z)
################################
## FIRST STEP: Polynomial Basis
################################
polyColl = PolynomialFamilyCollection(inputNbr)
hermiteFamily = HermiteFactory()
for i in range(inputNbr):
  polyColl[i] = OrthogonalUniVariatePolynomialFamily(hermiteFamily)
mycollZ = DistributionCollection()
for k in range(inputNbr):
  mycollZ.add(Distribution(Normal()))
myDistZ = ComposedDistribution(mycollZ)
multivariateBasis = OrthogonalProductPolynomialFactory
(polyColl,EnumerateFunction(inputNbr))
```

# How to build a Polynomial Chaos Expansion?

## Script

```
################################
## SECOND STEP: Truncature
################################
maximumConsideredTerms = consideredTerms(inputNbr, polynomialOrder)
# The maximum number of considered polynomials
mostSignificant = 10
# Defining the significance factor considered
significanceFactor = 1.e-6
truncatureBasisStrategy = CleaningStrategy(OrthogonalBasis(multivariateBasis),
    maximumConsideredTerms, mostSignificant, significanceFactor, True)
################################
## THIRD STEP: Evaluation
################################
evaluationCoeffStrategy = LeastSquaresStrategy(FixedExperiment(Z))
model = NumericalMathFunction(statModel(inputNbr, jobNbr, Z, outVar))
polynomialChaosAlgorithm =
    FunctionalChaosAlgorithm(model, Distribution(myDistZ),
    AdaptiveStrategy(truncatureBasisStrategy),
    ProjectionStrategy(evaluationCoeffStrategy))
################################
## LAST STEP: Run!
################################
polynomialChaosAlgorithm.run()
polynomialChaosResult = polynomialChaosAlgorithm.getResult()
```

# Parameterization of the PCE

## Choice of the polynomial basis

Choice guided by the distribution of the variable. Here, normal distribution $\rightsquigarrow$ **Hermite polynomials** basis.

## Truncature: maximum size of the basis

If we want all polynomials of degree D, with d inputs:

$$N = \frac{(D+d)!}{D!d!} \tag{5}$$

## Truncature: number of considered terms

Linked to the size of the training database, should be validated a posteriori. Here, we choose 10.

## Type and size of the training database

Linked to the computational budget, here 100. Choice between an already existing database, MC, LHS, etc.

# Validation process

## Chosen validation process

- Use a part of the database to train the model
- Use another part of the database to compare the surrogate model with the real one
- Different criterion can be chosen

$$d_2(f,\tilde{f}) = \frac{1}{|\mathcal{V}|}\left(\sum_{x\in\mathcal{V}}\left(\frac{f(x)-\tilde{f}(x)}{f(x)}\right)^2\right)^{1/2} \quad (6)$$

$$d_\infty(f,\tilde{f}) = \max_{x\in\mathcal{V}}\left|\frac{f(x)-\tilde{f}(x)}{f(x)}\right| \quad (7)$$

## Results

- For all surrogate models built, $d_2(f,\tilde{f}) < 1e-5$
- For all surrogate models built, $d_\infty(f,\tilde{f}) < 1e-4$

# Use the surrogate model to propagate uncertainty

## Compute the quantiles

- Empirical quantile with a very large sample
- Possible with the surrogate model

## Sensitivity analysis

With the polynomial chaos expansion, the computation of Sobol indices is easy.

$$Var(Y) = \sum_i Var\left[E[Y|X_i]\right] + \sum_{i<j} Var\left[E[Y|X_i, X_j]\right] + \cdots + Var\left[E[Y|X_1, \dots, X_n]\right]$$

$$S_{i_1,\dots,i_k} = \frac{Var\left[E[Y|X_{i_1}, \dots, X_{i_k}]\right]}{Var[Y]}$$

- For most outputs, for one manoeuvre, only 2 or 3 variables are involved
- For all manoeuvres agregated, usually all variables are involved

# Outline

# Interests

## Advantages of a surrogate model

- Faster computations
- Has to be goal-oriented (when possible)

## Special advantages of PCE

- Possibility to export the expression to another tool
- Warning: the polynomial is not the metamodel, but the polynomial part, need to compose with the distribution transformation
- Easy computation of all Sobol indices

# Difficulties

### Parameterization

- Choice of the basis: not difficult for parametric uncertainties
- Size of the basis: how to choose the maximum degree of the polynomials?
- Number of terms in the decomposition: difficult to guess?
- Size of the training database: how to be sure that the size of the database is large enough?
- Is the training database is correct for the use PCE?

### Solution

Does the Sparse Polynomial Chaos Expansion answer all these questions?