Open TURNS
Demonstration Software

CEA-EDF-INRIA Summer school

6th July 2011

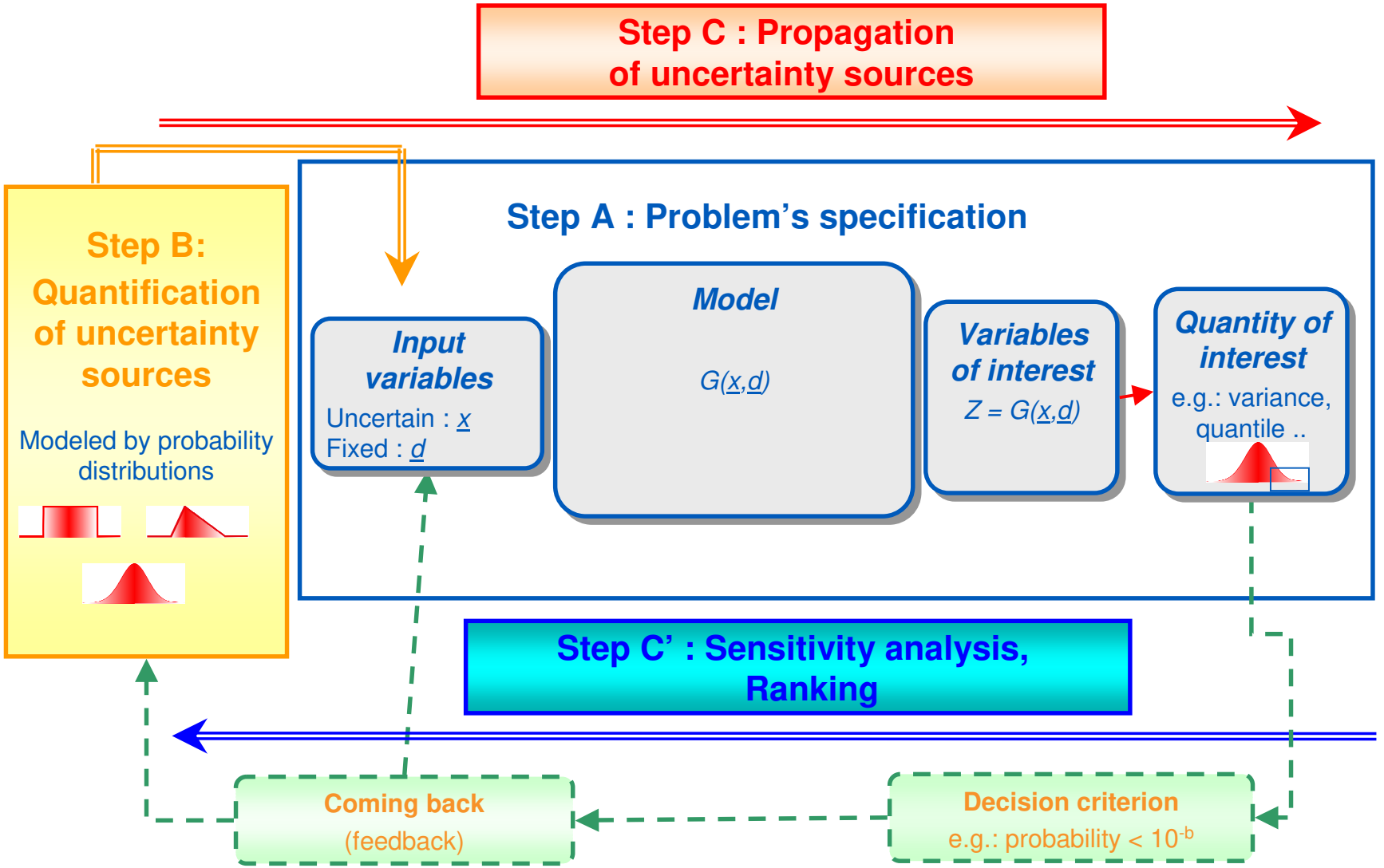
Anne-Laure Popelin (EDF – R&D)
Vincent Feuillard (EADS – IW)
Géraud Blatman (EDF – R&D)
Bertrand Iooss (EDF – R&D)
Merlin Keller (EDF –R&D)

EDF

Summary

- Presentation of Open TURNS (~25 min.)
- Presentation of the study case (~5 min.)
- Exercises (~2h)
- Questions and discussion
- Demonstration of additional functions (wrapper, GUI, website…)

Uncertainty management - the global methodology



Open TURNS : a software tool adapted to the industrial practice methodology

- TURNS : Treatments of uncertainties, risk'n statistics
- Open : Open source software (LGPL license)
  - Environment : Linux, Windows
  - Languages : C++ (libraries), Python (command scripts)
  - « Eficas » GUI helping the redaction of command scripts
- Partnership EDF-EADS-Phimeca since 2005
  - ✓ Transparency : to be understood and challenged by outside authorities and experts
  - ✓ Genericity : to allow a consistent treatment of multi-physical problems
  - ✓ Calculation performance : number of simulations generated by uncertainty treatment

Open TURNS : scientific content

Step A : Problem specification

- Model (under Linux or Windows)
- Enables to evaluate its grandient or not
- Criteria :
  - Min / Max,
  - Central Dispersion,
  - Probability to exceed a threshold

- The model G can be :
  - An analytical function (Python, see exercise)
  - An external code (see wrapper demonstration)

Open TURNS : scientific content

Step B : Quantification of uncertainty

- Available data ?
  - ✓ Parametric fitting : Maximum Likelihood and Moments based fitting methods, ...
  - ✓ Non parametric fitting : Kernel smoothing
  - ✓ Construction (nD), Empirical CDF, Empirical Copula...
  - ✓ Tests validation : Kolmogorov, Anderson-Darling, Cramer, ...
  - ✓ Graphics validation : QQ-Plot, Kendall plot, Henri line, ...

- No data?
  - ✓ Usual nD distributions : more than 40!
  - ✓ Dependence modelisation based on copula : Independent, Frank, Normal, Gumbel, Sklar Copula
  - ✓ Different ways to build a distribution :
    - ✓ (Marginals, Copula) : F(x1, ..., xn) = C(F1(x1), ..., Fn (xn))
    - ✓ Linear combination of pdf
    - ✓ Linear combination of variables

Open TURNS : scientific content

- Step C : Propagation of uncertainty sources
  - Simulation Methods : MC, LHS, Importance Sampling, Directional Sampling, ...
  - FORM / SORM methods with the Generalized Nataf transformation (any elliptical copula) or the Rosenblatt transformation
  - Taylor decomposition of variance
  - Experiment planes (composite, factorial, axial, …)

- Meta-models
  - Polynomials, Projection on any function basis (least squares)
  - Functional Chaos Expansion with advanced functionalities based on the LAR method

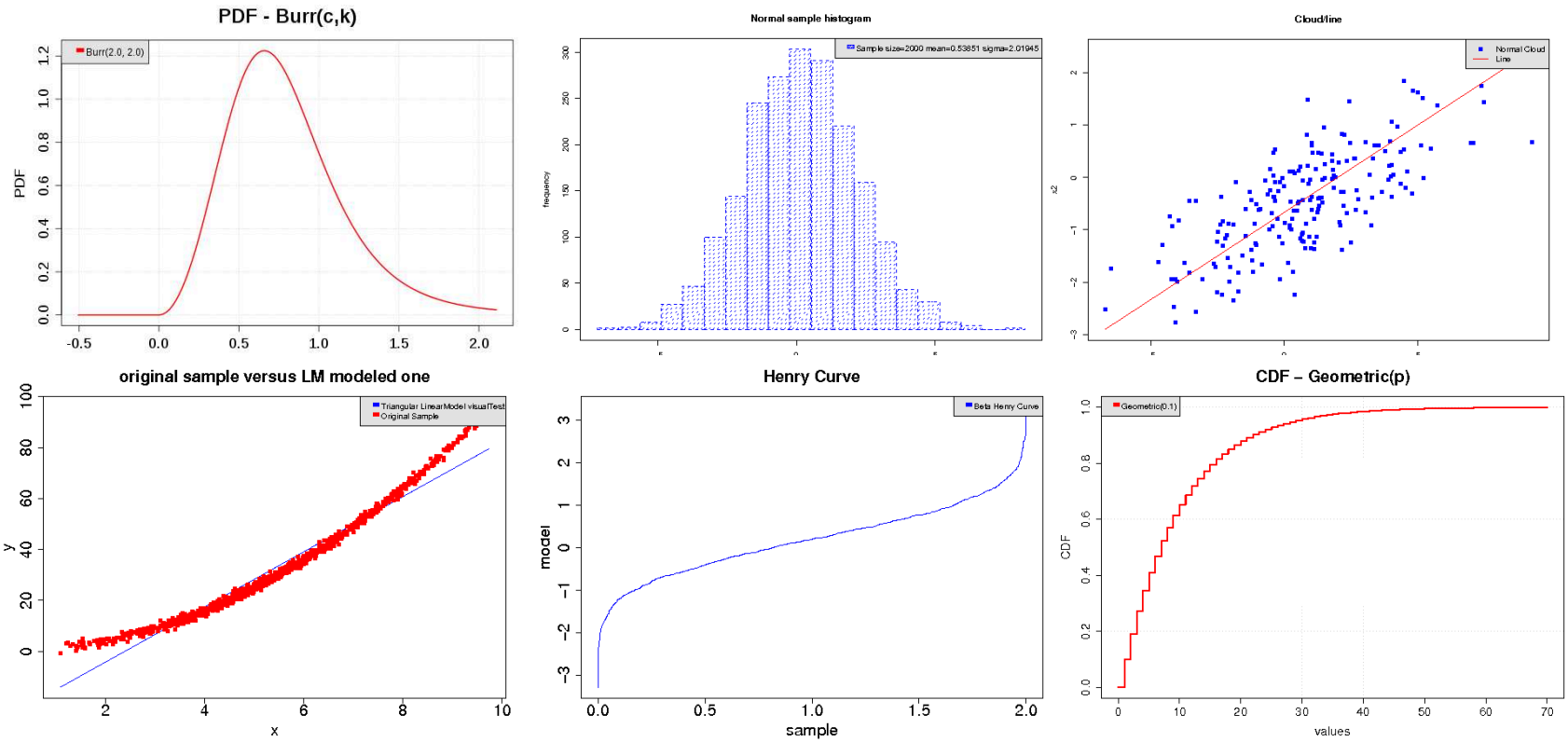
Open TURNS : scientific content

- Step C' : Sensitivity analysis, hierarchisation of uncertainty sources
  - Importance and Sensitivity Factors
  - Sobol Indices
  - Statistical coefficients : Pearson, Spearman, SRC, SRRC, PCC, PRCC
  - Regression analysis

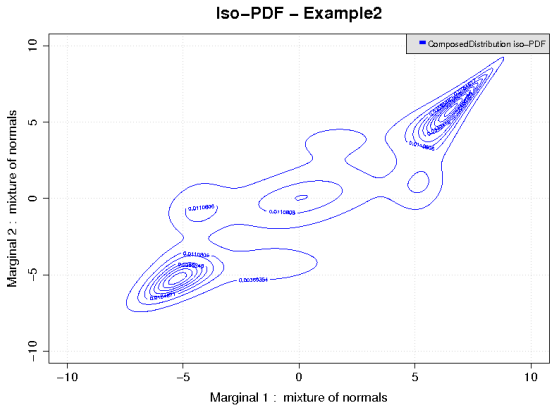
- Open TURNS integrates recent research results, such as :
  - Functional Chaos Expansion with advanced functionalities based on the LAR method (G. Blatman) → will be in the 0.15.0 release (summer 2011)
  - Accelerated simulation algorithm for low probability estimation, M. Munoz-Zuniga (soon)

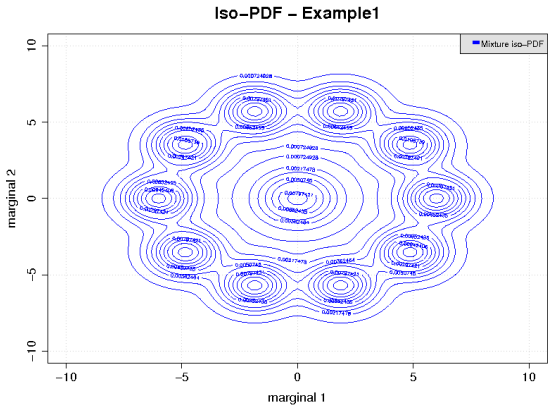
- More details and precise references in the « Reference Guide » of Open TURNS documentation

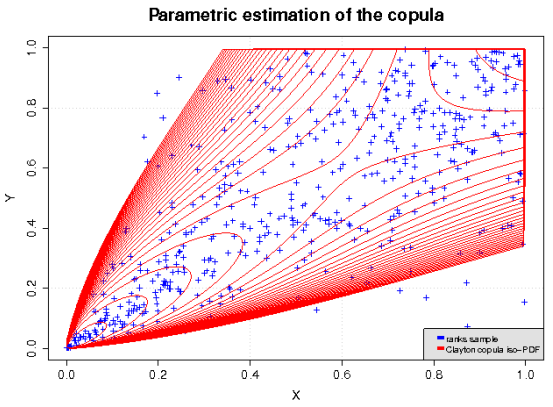
Open TURNS : pictures

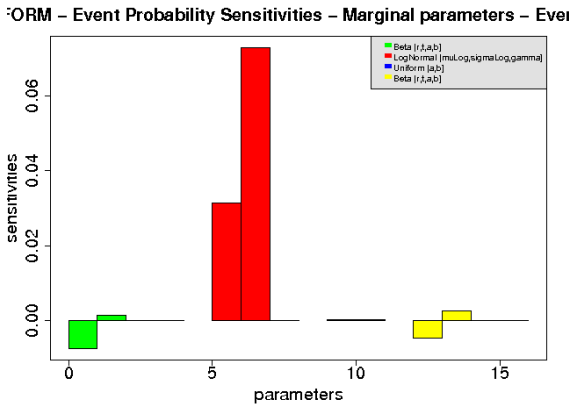


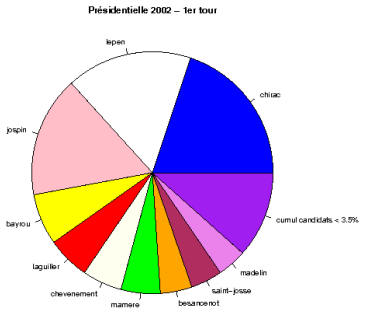
Open TURNS : pictures

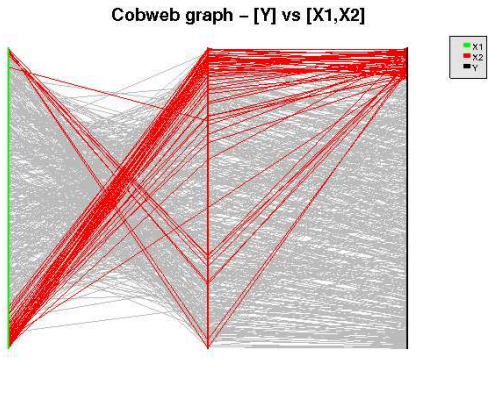




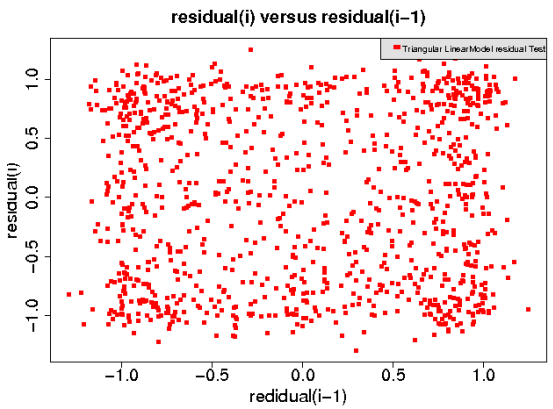


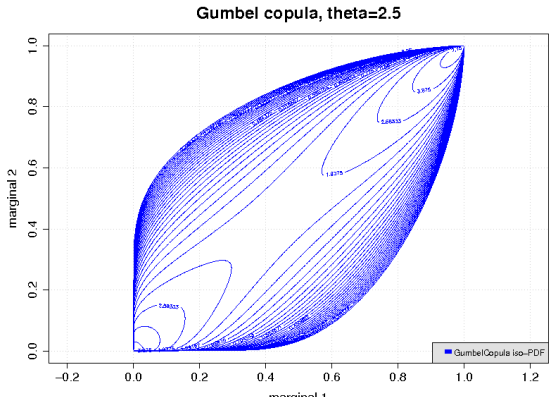


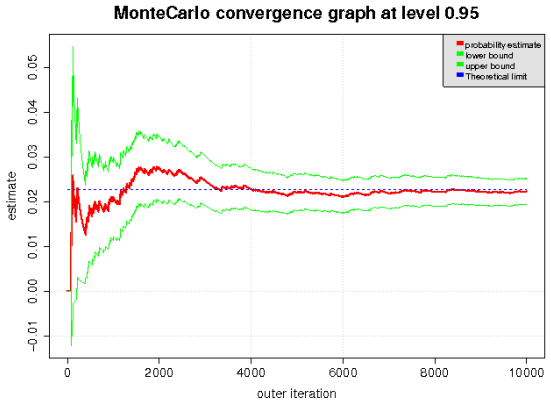


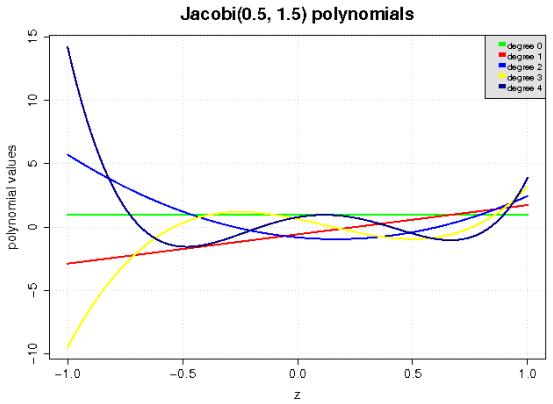


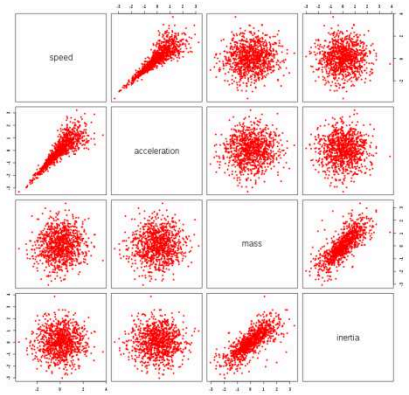
Open TURNS : pictures

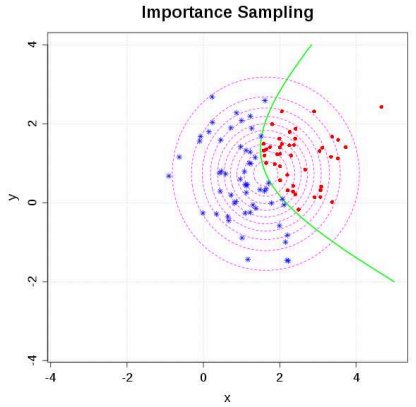












Open TURNS : software, doc and Users

What is Open TURNS? …

- ✓ A C ++ library with high level calculation operators
- ✓ An application with a GUI
- ✓ A Python module

And a documentation :

- ✓ scientific : Reference Guide,
- ✓ User : Use Cases Guide, User Manual, Examples Guide
- ✓ technical : Architecture Guide, Wrapper Guide, Contribution Guide, Windows port doc.

… and a users communauty :

- ✓ Openturns.org : official website
- ✓ « share » part : users have a « blog »

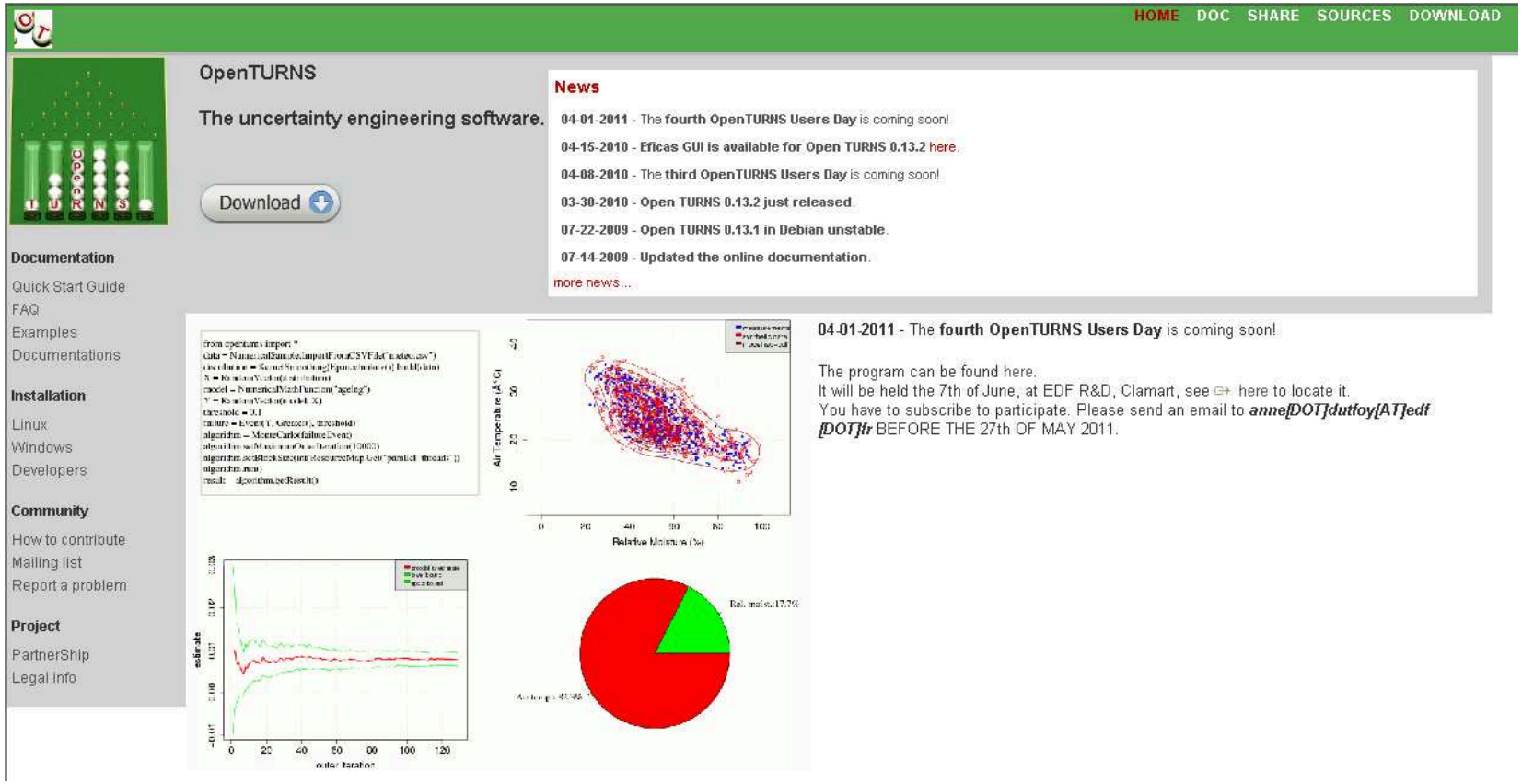
Annual Users Day (Users #4 : 7th Juny 2011, ~50 pers., 13 industries, 3 research labs)

↓

… welcome !

Open TURNS : Website

- **www.openturns.org**



Open TURNS : contributions

- **How to contribute to Open TURNS?**
  - ✓ **Level 1 : the user wants to share about the software, proposing additional functionalities (developed as a C++ or Python module), a particular script (pre / post treatment of data, wrapping to an open source tool, …)**
    - **No particular exigency**
    - **The author is responsible of the contribution.**
    - **➔ « Share » part of the Open TURNS website**
  - ✓ **Level 2 : the contributor would like the « OpenTURNS label » for the contribution**
    - **Contribution to the C++ library, to the Python TUI, or in R language**
    - **Consequence : the responsibility is for the Open TURNS team : maintenance, diffusion, …**
    - **➔ Quality criteria Open TURNS : source code + documentation (see Coding Rules Guide + Contribution Guide)**
  - **➔ Please do not hesitate to contact a core team member for any question, project, …**
- **Contacts :**
  - ✓ **bugs : bugs@openturns.org**
  - ✓ **users : users@openturns.org**
  - ✓ **developers : developers@openturns.org**

… or one of us of course!

Open TURNS : contribution as a module

- **Module mechanism of Open TURNS**
  - ✓ An easy way to develop for Open TURNS
  - ✓ Development cycle faster (compilation, tests, etc.)
  - ✓ For the new functionalities

- **Examples of 2 modules producted since 2009 :**

  - ✓ **Module OT-Agrum : link with the open source library aGrUM**
    - allows to model and simulate graphical models (including Bayesian networks)
    - developed by the computer science laboratory of Paris 6 : LIP6
    - Realised in collaboration with Pierre-Henri Wuillemin

  - ✓ **Module OT-MixMod : link with the open source library MixMod**
    - allows to reconstruct a Gaussian or multinomial mixture on a multi-variate sample for discriminant analysis and classification
    - developed by Université de Franche Comté
    - realised by Nolwenn Balin (EADS IW)

Prospects (for the longer term)

- **Stochastic processes : development begins in September 2011**
  - ✓ **The objective is to provide some modelisation, estimation and propagation functionalities in OpenTURNS, when input variables are described as stochastic processes or fields.**
    - ✓ **Could be scalar or vectorial field.**

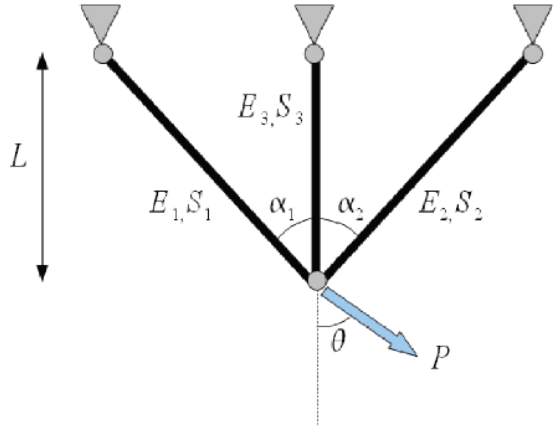
- **Bayesian approach**
  - ✓ **Modelisation of the joint distribution by conditioning**
  - ✓ **Bayesian regularization of prior distribution by data assimilation**

Some news of the 0.14.0 release

- **Technological uploads :**
  - ✓ **Generical wrapping mechanism**
  - ✓ **High Performance Computing**
  - ✓ **…**

- **Scientific uploads :**
  - ✓ **New distributions :**
    - ✓ 1 D continues: ArcSine, ArcSineFactory, Burr, BurrFactory, Chi, ChiFactory, FisherSnedecor, InverseNormal, InverseNormalFactory, NonCentralChiSquare, Rice , Trapezoidal, TrapezoidalFactory
    - ✓ • 1 D discrètes: Bernoulli, BernoulliFactory, Binomial, BinomialFactory, ZipfMandelbrot
    - ✓ • N D continues: Dirichlet, DirichletFactory
    - ✓ • N D discrètes: Multinomial, MultinomialFactory
  - ✓ **New graphs:**
    - ✓ Pairs for the 2D pairs of an nD sample
    - ✓ CobWeb plot
    - ✓ Kendall plot (graphical validation test for copulas)
    - ✓ …
  - ✓ **New low discrepancy sequences :**
    - ✓ InverseHaltonSequence
    - ✓ FaureSequence
  - ✓ **Chaos expansion for vector-valued model response**
  - ✓ **…**

Presentation of the study case : Elastic three-bar truss



$$\delta_v = \left[\frac{\alpha_v^2}{E_1 S_1 \cos(\alpha_1)} + \frac{\alpha_v^2 \sin^2(\alpha_1)}{E_2 S_2 \cos(\alpha_2) \sin^2(\alpha_2)} + \frac{\left(1 - \frac{\sin(\alpha_1+\alpha_2)}{\sin(\alpha_2)}\right)^2}{E_3 S_3} \right] P_v \, L$$

$$\delta_h = \left[\frac{\alpha_v^2}{E_1 S_1 \cos(\alpha_1)} + \frac{\left(\alpha_h \frac{\sin(\alpha_1)}{\sin(\alpha_2)}\right)^2}{E_2 S_2 \cos(\alpha_2)} + \frac{\left(\alpha_h \frac{\sin(\alpha_1+\alpha_2)}{\sin(\alpha_2)} - \frac{\cos(\alpha_2)}{\sin(\alpha_2)}\right)^2}{E_3 S_3} \right] P_h \, L$$

$$\delta = \sqrt{\delta_h^2 + \delta_v^2}$$

Variable	Distribution	Mean	Coef. of variation
E_i	Lognormal	210 GPa	10%
S_i	Normal	$1.5 \cdot 10^{-3}$ m^2	5%
P	Gumbel	$2.5 \cdot 10^5$ N	20%
α_j	Normal	45°	3%
θ	Normal	45°	3%

Presentation of the study case (0.13.2 release)

- 6 Python files :
  - StepA.py : implementation of the physical problem as a Python function
  - StepB_fromData.py : let's consider we have data for the « P » variable and let's try to adjust some distributions, then choose the « best » one
  - StepB.py : we define the probabilistic model
  - StepC_CentralTendancy.py : Quadratic Cumul and Kernel Smoothing
  - StepC_FORM.py : threshold exceedance estimation using FORM
  - StepC_MonteCarlo.py : threshold exceedance estimation using MonteCarlo
- Follow the documentation. No need to know Python!
  - ExampleGuide
  - UseCasesGuide
  - UserManualTUI
  - ReferenceGuide

To start

- « source export/opt/env.sh » in the console
- Copy the files in your repertory to have the rights
- To run a file : « python myfile.py »