NURESAFE WP1.1
TESTBED FOR INTEGRATED COUPLING AND UNCERTAINTY QUANTIFICATION METHODS

COBRA-TF sensitivity analysis with URANIE

J. Jimenez, N. Trost, L. Mercatali, V. Sanchez

Presented by J. Jimenez

victor.sanchez@kit.edu
or
javier.jimenez@kit.edu

Outline

- NURESAFE COBRA-TF
- Description of the URANIE software
- Scripts for Steady state and transient analysis
- Application to the O2 -1999 Feedwater transient
- Comparison of results against SUSA
- Conclusion and Outlook

NURESAFE COBRA-TF version

- COBRA-TF is a thermal-hydraulic simulation code designed for Light Water Reactor (LWR) vessel analysis.
- It is being developed and improved by the Reactor Dynamics and Fuel Management Group (RDFMG) at the Pennsylvania State University (PSU).
- COBRA-TF uses a two-fluid three fields formulation for the simulation of two phase flow (9 equation system for the conservation of mass, momentum and energy).
- Within NURESAFE, COBRA-TF is being delivered by GRS to the partners.
- Code license agreement need to be signed between PSU and each interested partner.
- The same source is used as for the CASL program (USA).

URANIE Software

- URANIE is a software dedicated to uncertainty and optimization.
- It allows to perform studies on uncertainty propagation, sensitivity analysis or model calibration in an integrated environment.
- Based on ROOT, a software developed at CERN for particle physics data analysis. As a result, URANIE benefits from the numerous features of ROOT, among which:
  - a C++ interpreter (CINT)
  - a Python interface (PyROOT)
  - access to SQL databases
  - many advanced data visualization features
- URANIE training course attended 2-4th April 2013 in Saclay.

URANIE Software

- At KIT, URANIE software has been applied to perform sensitivity analysis based on stochastic sampling (MC).
- Advantages of a Statistical Methodology:
  - Sound mathematical basis.
  - Reduction of Expert Opinion to the minimum needed.
  - There is no limit in the number of variables and models that can be used (No need for a previous PIRT).
  - The actual BE Code is used for the calculations (No need for regression based surfaces to replace the code).
  - The uncertainty can be quantified in transient analysis.
  - Objective determination of important variables and models.
- The methodology is concerned mainly with the uncertainty in:
  - Code´s input variables.
  - Code´s correlations and physical models.
- Confidence level determined by Wilk´s formula (95% of probability with 95% confidence level with 93 runs).

URANIE Software

- **The stochastic nature (PDF) of the uncertainty in the input variables and in the physical models induces a statistical nature in the results of the code.**
- The idea behind the statistical Methodology is quite simple
  - Uncertainties in code inputs are treated as Stochastic Variables.

Input Sample of Size N

Random Sampling

Output Sample of Size N

$$\begin{pmatrix}(\boldsymbol{X})_1\\(\boldsymbol{X})_2\\\vdots\\(\boldsymbol{X})_N\end{pmatrix} \equiv \begin{pmatrix}(X_1,\dots,X_k)_1\\(X_1,\dots,X_k)_2\\\vdots\\(X_1,\dots,X_k)_N\end{pmatrix} \xrightarrow[\text{to Input}]{\text{Assign Values}} \begin{pmatrix}(x_1,\dots,x_k)_1\\(x_1,\dots,x_k)_2\\\vdots\\(x_1,\dots,x_k)_N\end{pmatrix} \rightarrow \begin{pmatrix}Code\left[(x_1,\dots,x_k)_1\right]\\Code\left[(x_1,\dots,x_k)_2\right]\\\vdots\\Code\left[(x_1,\dots,x_k)_N\right]\end{pmatrix} \rightarrow \begin{pmatrix}(y_1,\dots,y_m)_1\\(y_1,\dots,y_m)_2\\\vdots\\(y_1,\dots,y_m)_N\end{pmatrix} \equiv \begin{pmatrix}Code\left[(\boldsymbol{X})_1\right]\\Code\left[(\boldsymbol{X})_2\right]\\\vdots\\Code\left[(\boldsymbol{X})_N\right]\end{pmatrix} \rightarrow \begin{pmatrix}(\boldsymbol{Y})_1\\(\boldsymbol{Y})_2\\\vdots\\(\boldsymbol{Y})_N\end{pmatrix}$$

Input Multivariate
k stochastic input variables

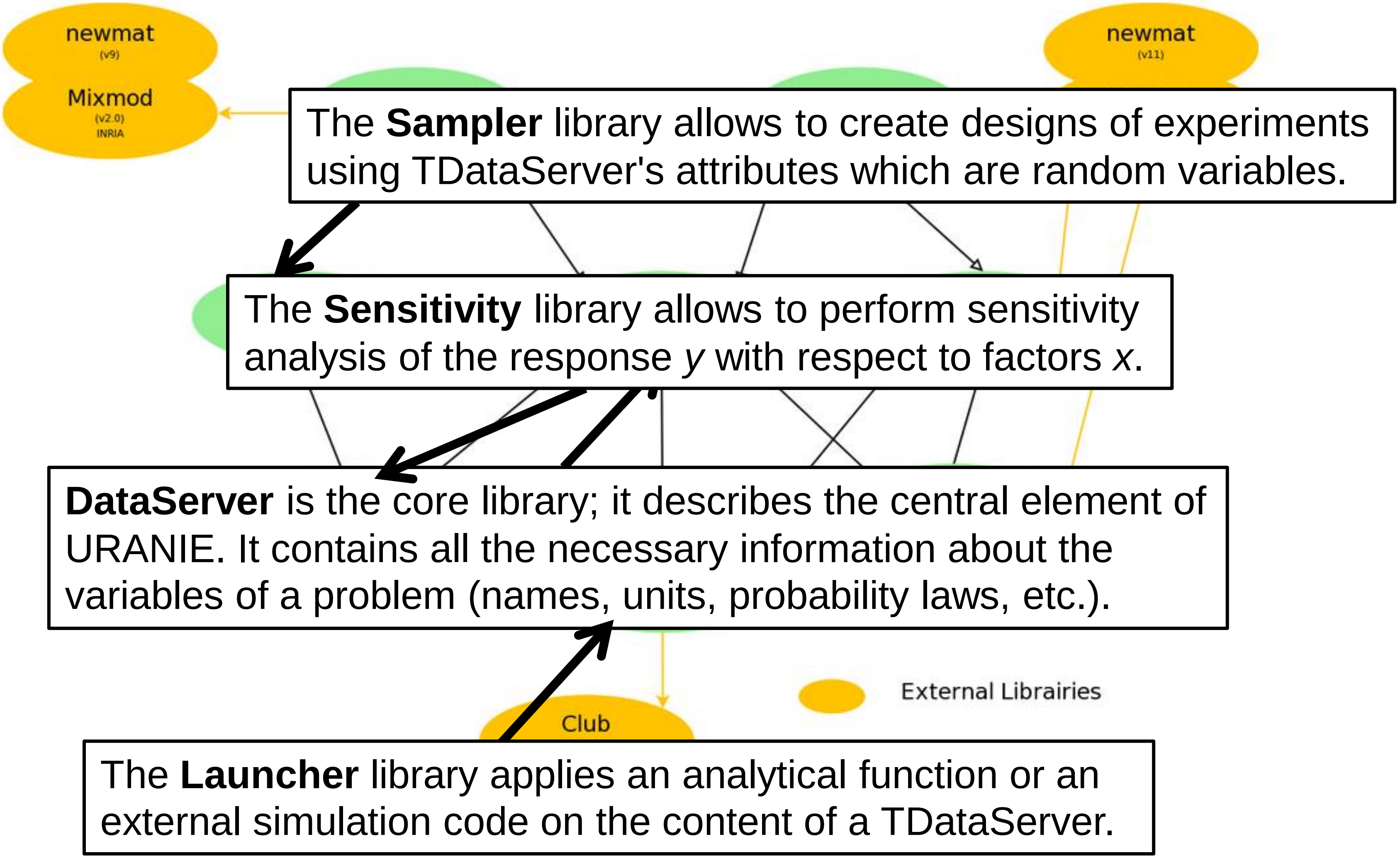
Input Variables for N
Code Executions

N Code Executions

Output Multivariate
m stochastic output variables

  - Deterministic Code transforms Stochastic INPUT in Stochastic OUTPUT.
  - Uncertainty in INPUT is PROPAGATED to OUTPUT.
  - Statistical Methods extract uncertainty information from OUTPUT.

URANIE Software Functional Diagram



The **Sampler** library allows to create designs of experiments using TDataServer's attributes which are random variables.

The **Sensitivity** library allows to perform sensitivity analysis of the response *y* with respect to factors *x*.

DataServer is the core library; it describes the central element of URANIE. It contains all the necessary information about the variables of a problem (names, units, probability laws, etc.).

The **Launcher** library applies an analytical function or an external simulation code on the content of a TDataServer.

Script for COBRA-TF run

```
void BFBT_P6x(Int_t nS = 100) {

  TDataServer *tds = new TDataServer("tdsCTF", "BFBT_P6x");

  TNormalDistribution *tnp = new TNormalDistribution("OutletPressure", 7.16e6, 71600.0);  // +-1%
  TNormalDistribution *tnf = new TNormalDistribution("MassFlowRate", 5.61, 0.0561);       // +-1%
  TUniformDistribution *tut = new TUniformDistribution("InletTemperature", 276.3, 279.3); // +-1.5K
  TNormalDistribution *tnl = new TNormalDistribution("Power", 1.951e6, 29265.0);          // +-1.5%

  tnp->setBounds(7.0884e6, 7.2316e6);
  tnf->setBounds(5.5539, 5.6661);
  tnl->setBounds(1.921735e6, 1.980265e6);

  tds->addAttribute(tnp);
  tds->addAttribute(tnf);
  tds->addAttribute(tut);
  tds->addAttribute(tnl);

  TString sFileName = TString("deck.inp");
...
```

Specify the number of simulation runs

Create DataServer to hold all informations

Specify the distribution for each parameter

Set upper and lower bound to avoid non-physical values

Fill up the DataServer

Specify the input file of COBRA-TF to be parsed by URANIE and replaced by the sampled random data

Script for COBRA-TF run

```
...
  tds->getAttribute("pref")->setFileFlag(sFileName, "@Pref@");
  tds->getAttribute("massf")->setFileFlag(sFileName, "@Massf@");
  tds->getAttribute("hin")->setFileFlag(sFileName, "@Hin@");
  tds->getAttribute("aflux")->setFileFlag(sFileName, "@Aflux@");

  TSampling *sampling = new TSampling(tds, "lhs", nS);
  sampling->generateSample();

  TOutputFileKey *fout = new TOutputFileKey("result_channels.out");

  TAttribute *avgpres = new TAttribute("total_ax_pres_loss_uranie");
  avgpres->setDefaultValue(-200.0);
  fout->addAttribute(avgpres);

  TCode *mycode = new TCode(tds, "COBRA-TF");
  mycode->addOutputFile(fout);

  TLauncher *tlch = new TLauncher(tds, mycode);
  tlch->setSave();
  tlch->setClean();
  tlch->setWorkingDirectory(gSystem->Getenv("PWD") + TString("/tmpUranie/cobratf"));
  tlch->setVarDraw("MassFlowRate:total_ax_pres_loss_uranie","","");
...
```

Give URANIE the strings of COBRA-TF input to be changed

Generate the random data

Specify the output file to extract the data for post-processing

Tell URANIE which value has to be extracted from CTF output and initialize it with a non-physical value for easier error detection

Initiate a COBRA-TF simulation run

Script for COBRA-TF post-processing

```
 TCanvas *Canvas = new TCanvas("c1", "Graph",5,64,1270,667);
  c1->Divide(2, 2);
  c1->cd(1);
  tlch->run();
  c1->cd(3);
  tds->draw("OutletPressure:total_ax_pres_loss_uranie");
  c1->cd(2);
  tds->draw("InletTemperature:total_ax_pres_loss_uranie","","");
  c1->cd(4);
  tds->draw("Power:total_ax_pres_loss_uranie","","");

  TCanvas *Canvas2 = new TCanvas("c2", "Graph",5,64,1270,667);
  tds->draw("total_ax_pres_loss_uranie");

  tds->exportData("BFBT_P6x_Sampling.dat");

}
```

Visualize the output of URANIE

Plot a histogram of the output data

Export the output as well as the sampled random numbers into a file for post-processing.

Script for COBRA-TF statistics

```
{
  using namespace URANIE::DataServer;
  using namespace URANIE::Sampler;

  TDataServer *tds = new TDataServer();

  tds->fileDataRead("COBRA-TF_O2.dat");

  tds->addAttribute("OutletPressure", "OutletPressure");
  tds->addAttribute("MassFlowRate", "MassFlowRate");
  tds->addAttribute("InletTemperature", "InletTemperature");
  tds->addAttribute("Power", "Power");
  tds->addAttribute("AxialPressureLoss", "total_ax_pres_loss_uranie");

  tds->computeStatistic();
  tds->computeCorrelationMatrix()->Print();

  std::cout << "mean: " << tds->getAttribute("AxialPressureLoss")->getMean()    << std::endl;
  std::cout << "min: "  << tds->getAttribute("AxialPressureLoss")->getMinimum() << std::endl;
  std::cout << "max: "  << tds->getAttribute("AxialPressureLoss")->getMaximum() << std::endl;
  std::cout << "std: "  << tds->getAttribute("AxialPressureLoss")->getStd()     << std::endl;
}
```

Read in an URANIE file that contains sampled random data

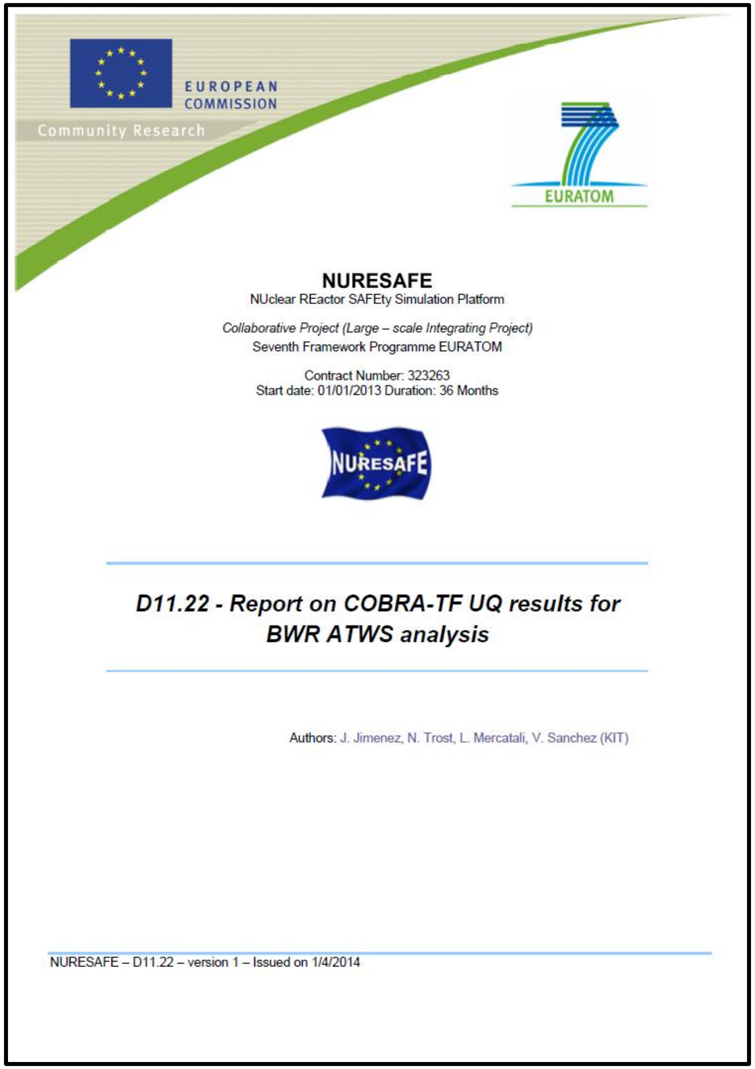
Fill up the DataServer structure

Compute the statistics and print the correlation matrix

Print mean, min, max, and standard deviation

Summary of the COBRA-TF URANIE scripts

- The same scripts can be used for:
  - Steady state
  - Transient analyses
- Dedicated general script for post-processing.
- Around 500 lines of URANIE code.
- Documentation reported under D11.22 deliverable (WP1.1).



EUROPEAN COMMISSION

Community Research

EURATOM

NURESAFE

NUclear REactor SAFEty Simulation Platform

Collaborative Project (Large – scale Integrating Project)

Seventh Framework Programme EURATOM

Contract Number: 323263

Start date: 01/01/2013 Duration: 36 Months

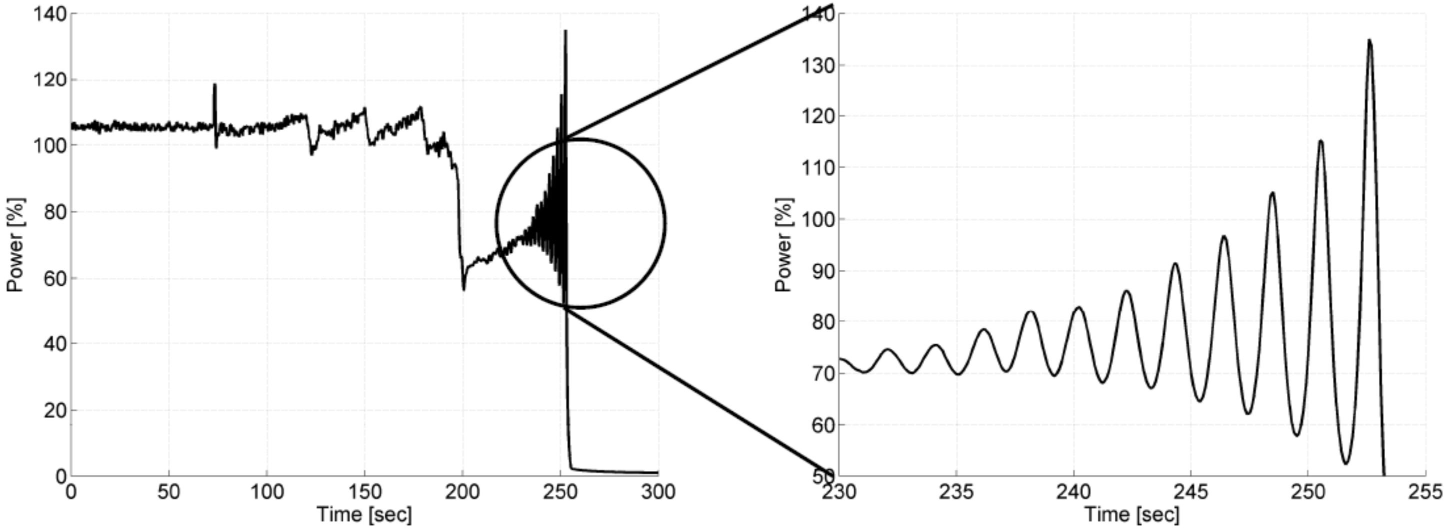
D11.22 - Report on COBRA-TF UQ results for BWR ATWS analysis

Authors: J. Jimenez, N. Trost, L. Mercatali, V. Sanchez (KIT)

NURESAFE – D11.22 – version 1 – Issued on 1/4/2014

Application to the O2 -1999 FW transient

- Power oscillation during the event (feedwater transient)



Oskarshamn-2 February 25, 1999 feedwater transient

- Boundary conditions taken from TRACE/PARCS calculation (KIT model with 444 channels)
- Modeling the O2 core with COBRA-TF using 444 channels (WP1.3)

Sensitivity study, PDFs determination

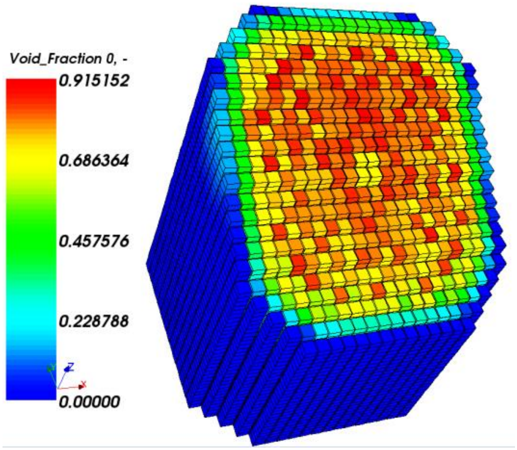
- Sensitivity analysis with parameters taken from the NURESAFE benchmark specifications (D13.11)

No.	Parameter	Range	Distribution
1	Outlet pressure	± 0.5 %	Uniform
2	Mass flow rate	± 0.5 %	Uniform
3	Inlet temperature	± 2.0 %	Normal
4	Power	± 0.75 %	Normal
5	Cladding Wall Roughness	± 30.0 %	Normal
6	Spacer grid pressure drop coefficient	± 5.0 %	Uniform
7	Gap Conductance	± 35.0 %	Uniform
8	Fuel Conductivity	± 10.0 %	Uniform
9	Cladding Conductivity	± 6.25 %	Uniform

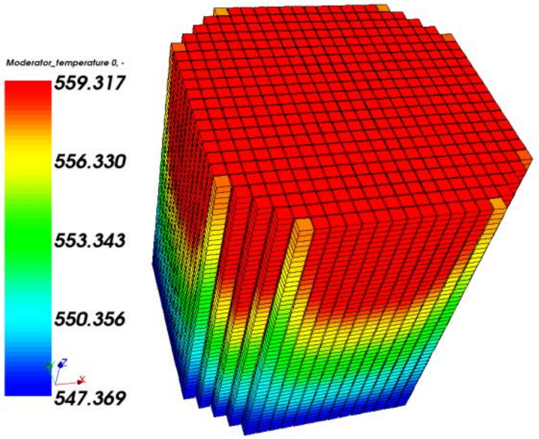
O2 nominal steady state results

- **Axial pressure drop** and **Outlet void fraction** are the output parameters studied (200 runs were used).
- The computed Pearson sensitivity coefficients by URANIE corresponding to a steady state at nominal operating conditions using COBRA-TF.

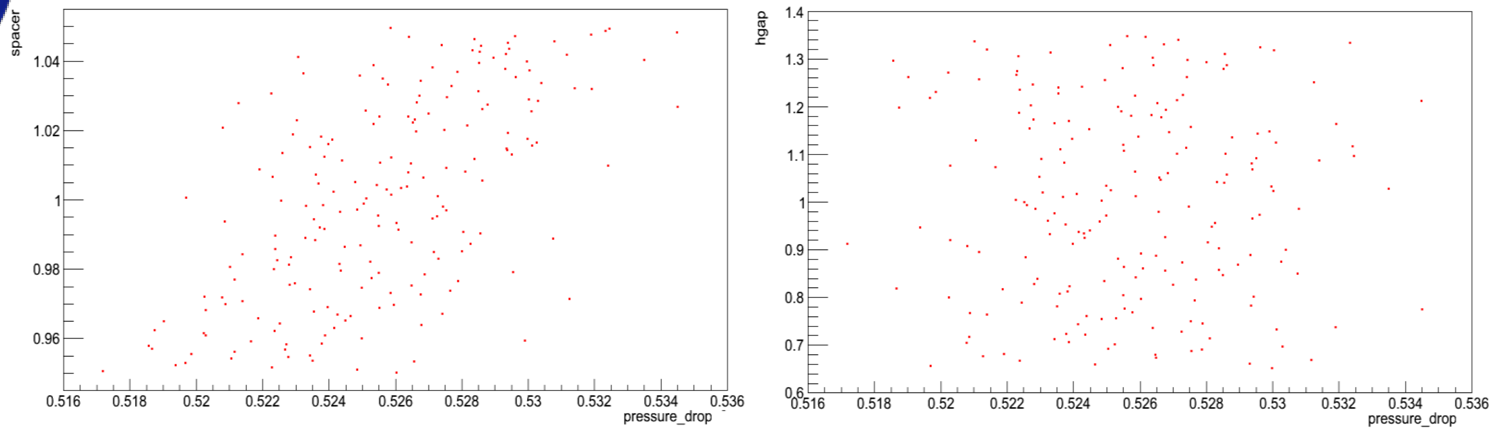
	Mass flow rate	Inlet enthalpy	Pressure	Heat flux	Spacer	Gap conductivity
Axial pressure loss	0.259488	0.384382	-0.52228	0.410298	0.597949	-0.0036
Void Fraction	-0.198526	0.673415	-0.660979	0.275753	-0.0077247	0.02582



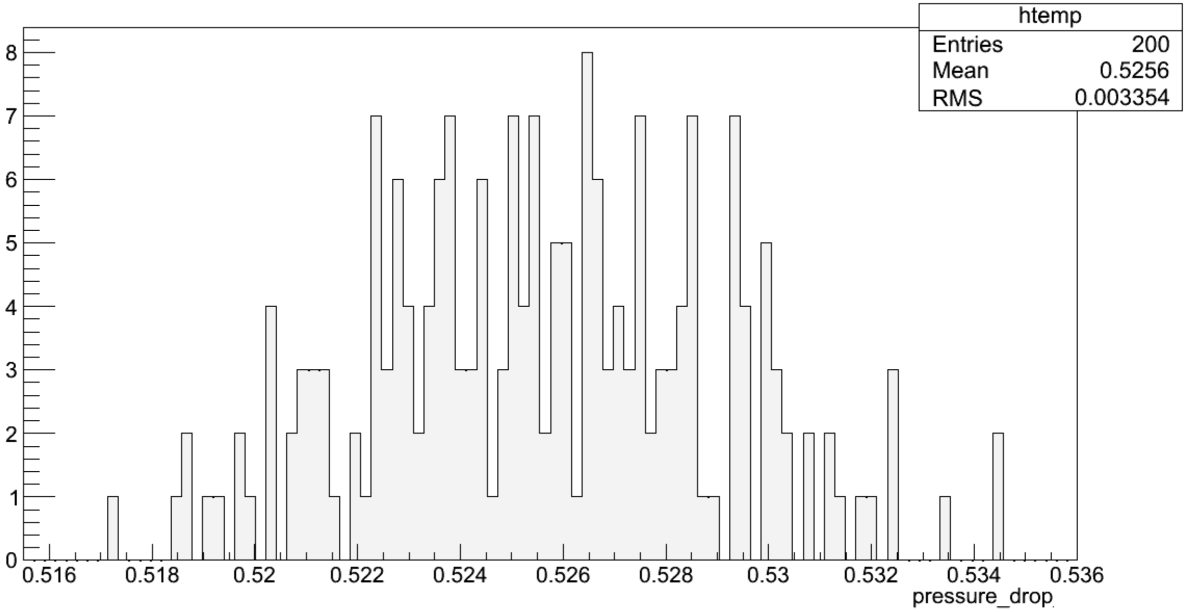
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1																								
2						1	1	1	1	1	1	1	1	1	1	1	1	1	1					
3					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
4				1	1	1	3	1	2	2	1	2	2	2	2	2	1	2	1	1	1			
5			1	1	2	2	1	3	4	1	3	1	4	3	1	4	1	1	2	2	1	1		
6		1	1	2	2	1	4	2	1	3	2	3	1	1	3	1	3	4	1	2	2	1	1	
7		1	1	2	1	4	3	1	1	2	3	1	3	3	1	2	1	3	4	1	3	1	1	
8		1	1	1	2	3	1	4	2	2	2	2	2	2	1	1	3	1	2	3	2	1	1	
9		1	1	4	4	1	3	1	3	2	2	2	3	3	1	3	1	3	1	4	2	1	1	
10		1	1	4	1	3	2	3	1	1	1	3	2	2	2	2	2	2	2	1	2	1	1	
11		1	1	1	2	1	3	1	3	2	3	2	2	3	3	2	2	3	1	3	2	1	1	
12		1	1	1	4	1	3	1	3	3	2	2	2	2	2	1	1	1	3	1	2	1	1	
13		1	1	2	1	3	1	1	1	2	2	2	2	2	3	3	1	3	1	4	1	1	1	
14		1	1	2	3	1	3	2	2	3	3	2	2	3	2	3	1	3	1	2	1	1	1	
15		1	1	2	1	2	2	2	2	2	2	2	3	1	1	1	3	2	3	1	4	1	1	
16		1	1	2	4	1	3	1	3	1	3	3	2	2	2	3	1	3	1	4	4	1	1	
17		1	1	2	3	2	1	3	1	1	2	2	2	2	2	2	4	1	3	2	1	1	1	
18		1	1	3	1	4	3	1	2	1	3	3	1	3	2	1	1	3	4	1	2	1	1	
19		1	1	2	2	1	4	3	1	3	1	1	3	2	3	1	2	4	1	2	2	1	1	
20			1	1	2	2	1	1	4	1	3	4	1	3	1	4	3	1	2	2	1	1		
21				1	1	1	2	1	2	2	2	2	2	1	2	2	1	3	1	1	1			
22					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
23						1	1	1	1	1	1	1	1	1	1	1	1	1	1					
24																								



O2 nominal steady state results



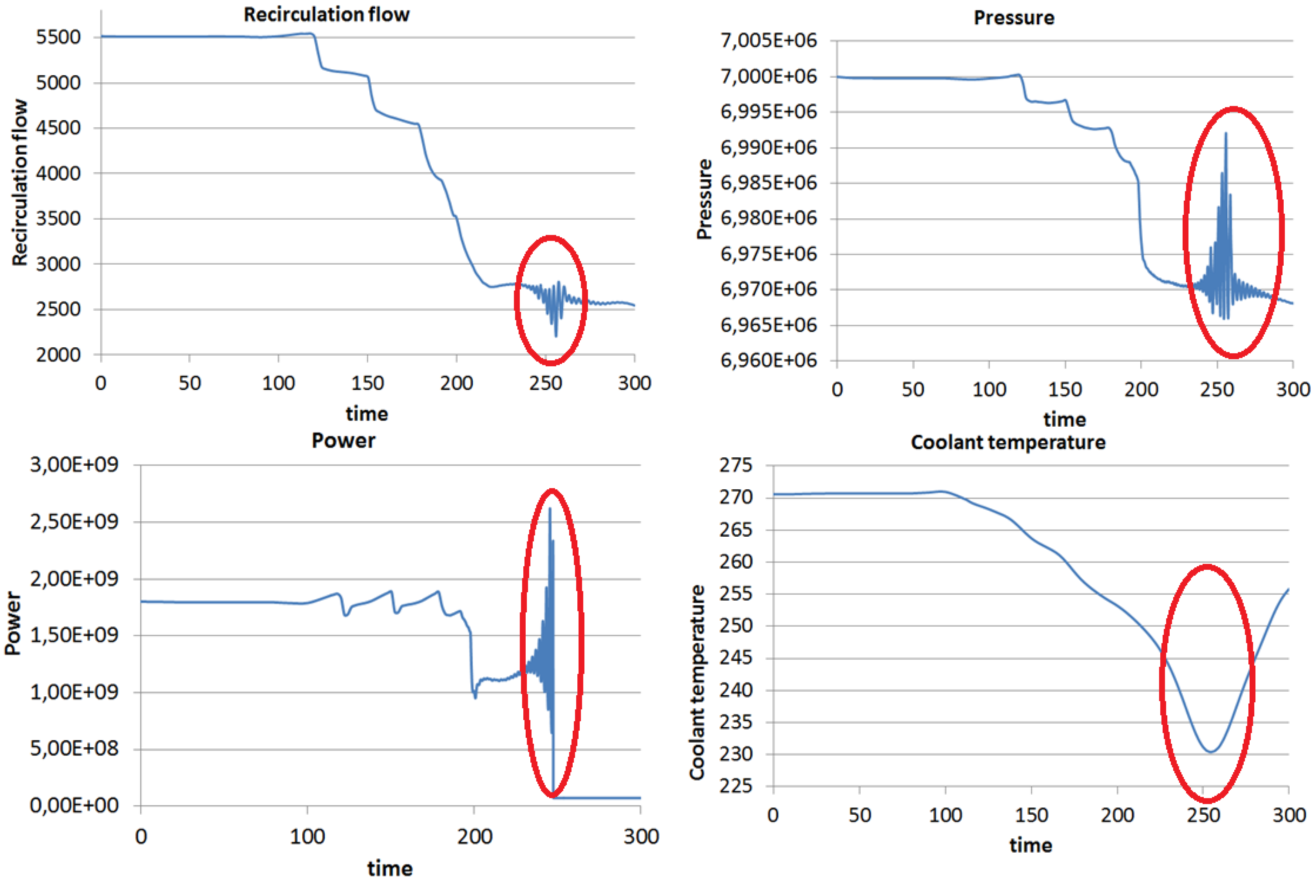
Axial pressure drop for different spacer coefficient and gap boundary conditions



Pressure drop distribution over all COBRA-TF runs

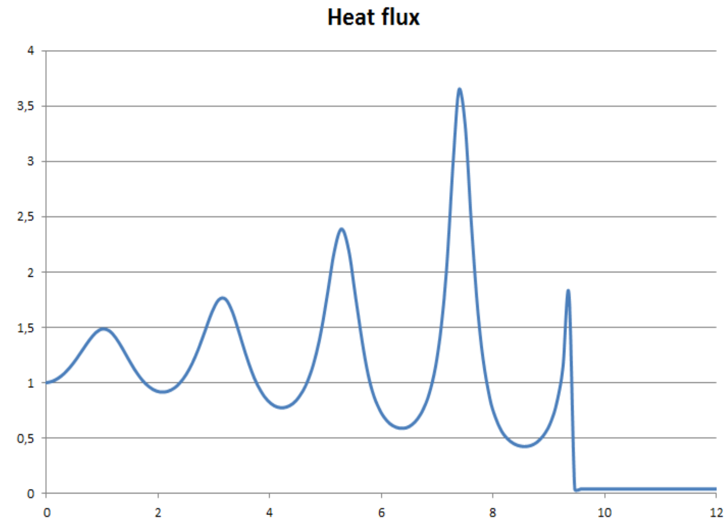
Transient Boundary Conditions applied

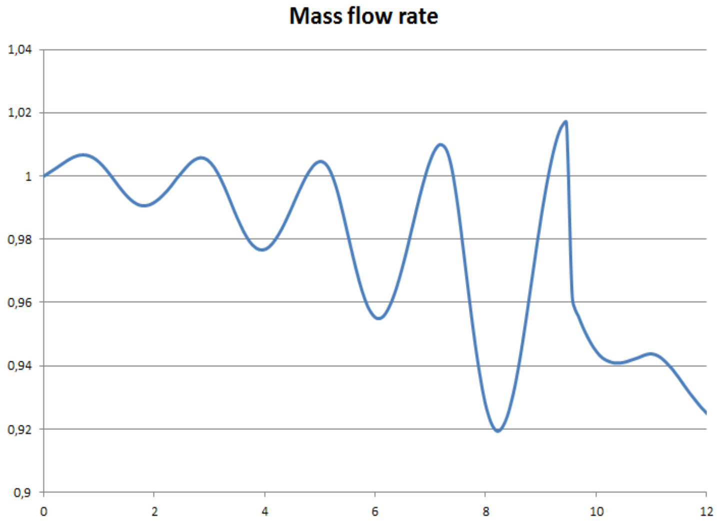
- The next boundary conditions were introduced into CTF for the simulation of the oscillations (only 12s are analyzed).
- They have been extracted from a TRACE5p3/PARCS results
  - Power, inlet temperature, pressure, mass flow rate.

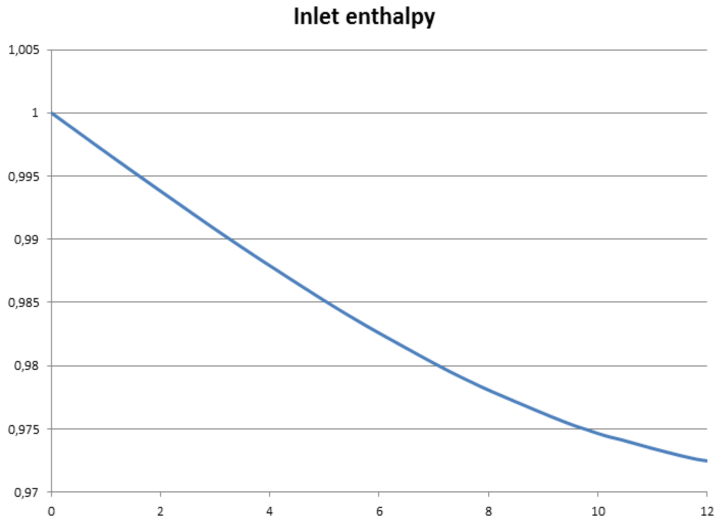


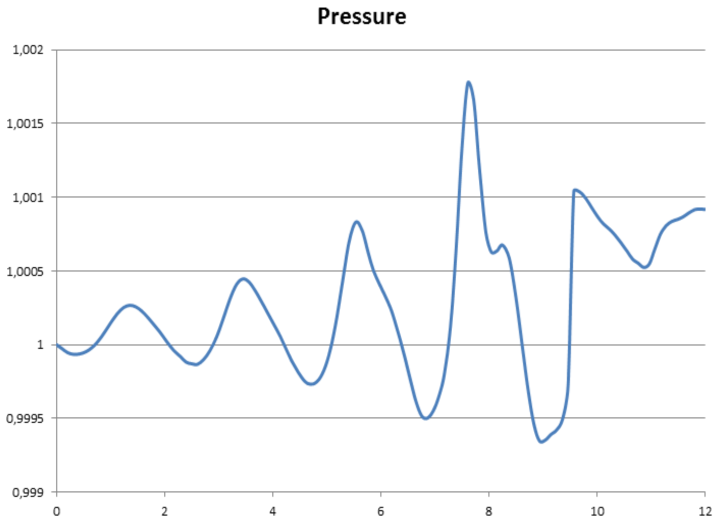
Transient Boundary Conditions applied

- Those BC are representative of a stability event.


Heat flux

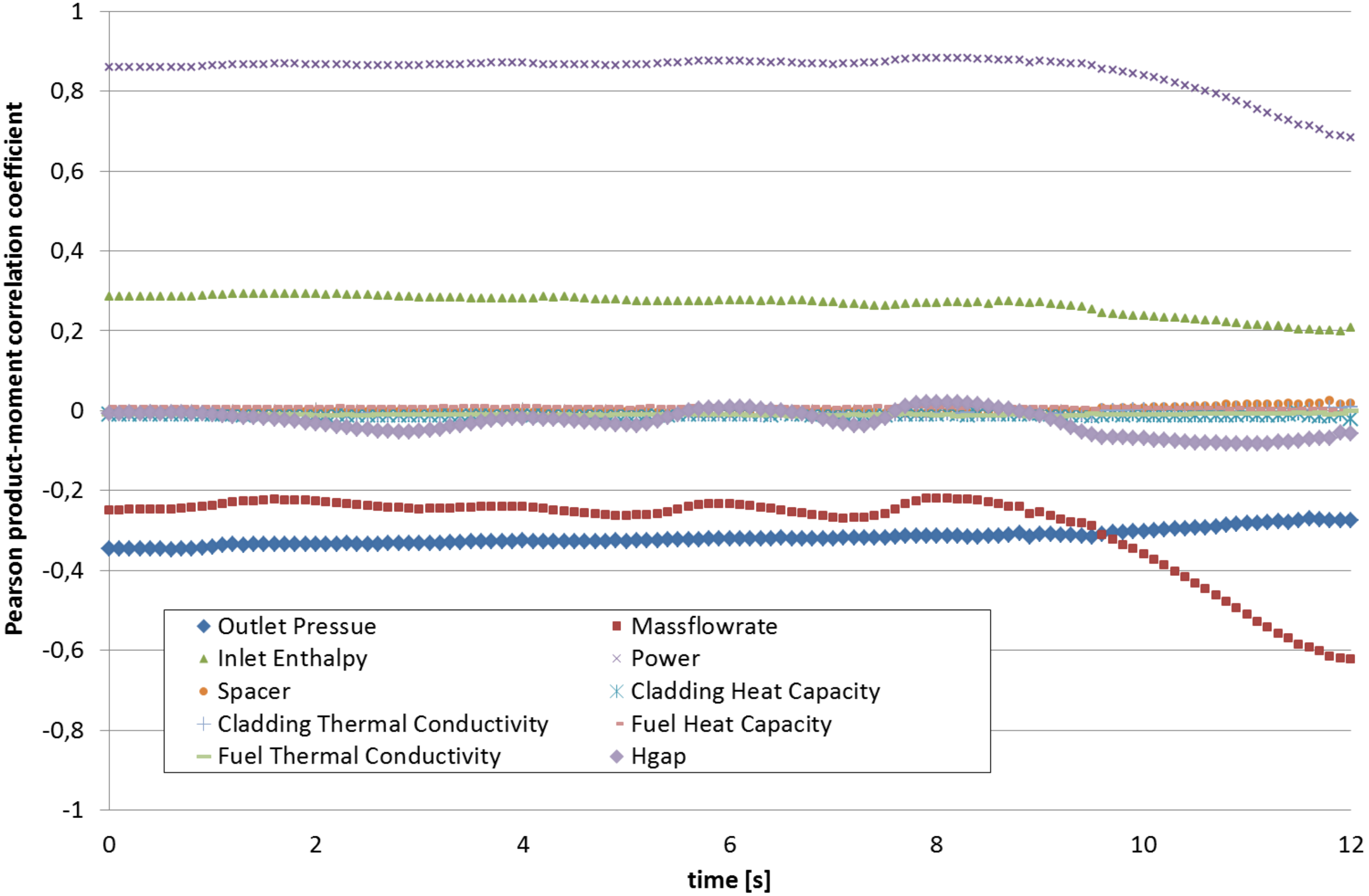

Mass flow rate


Inlet enthalpy


Pressure

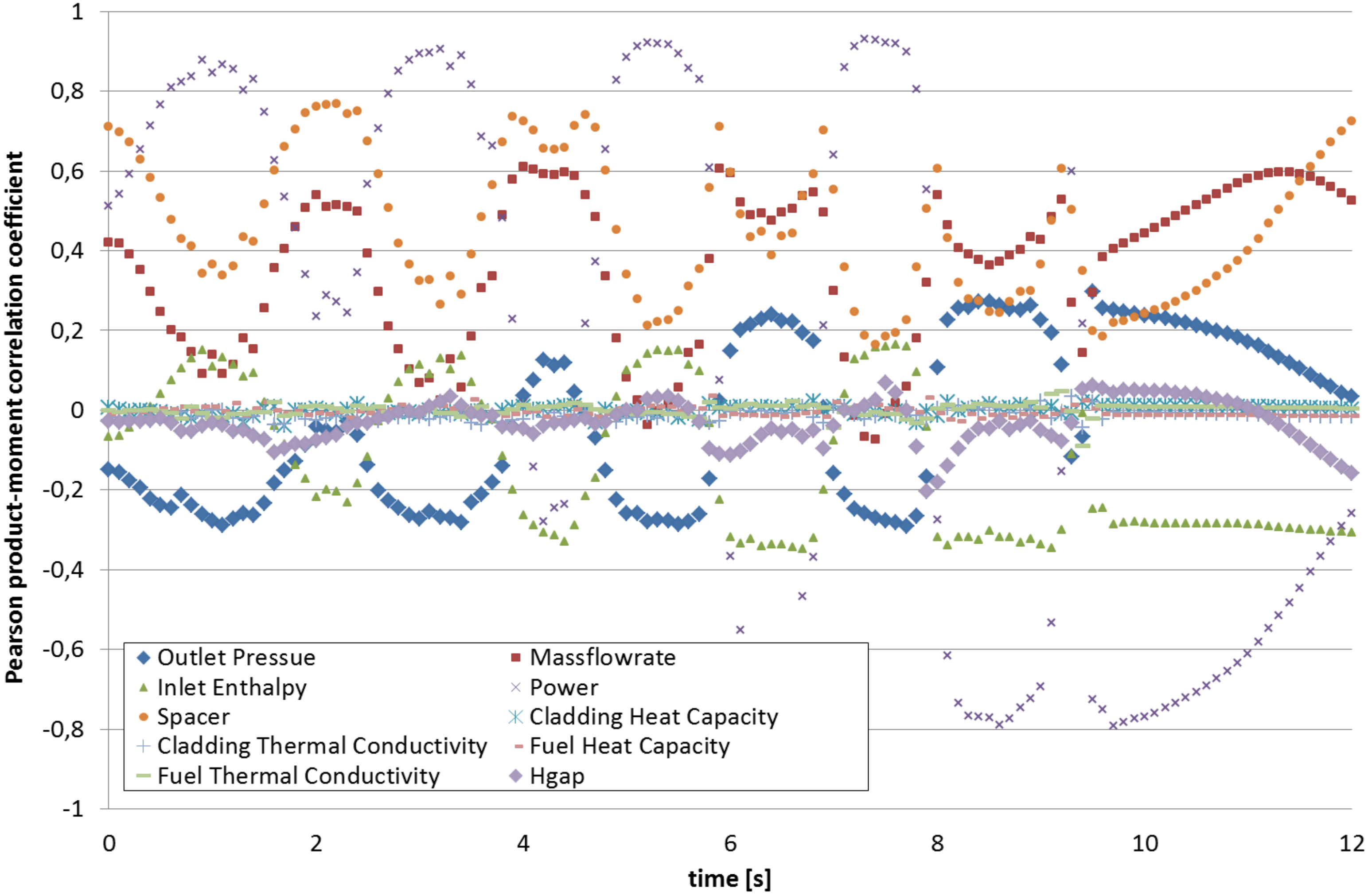
URANIE results in the zooming area

- Pearson sensitivity coefficients of the void fraction



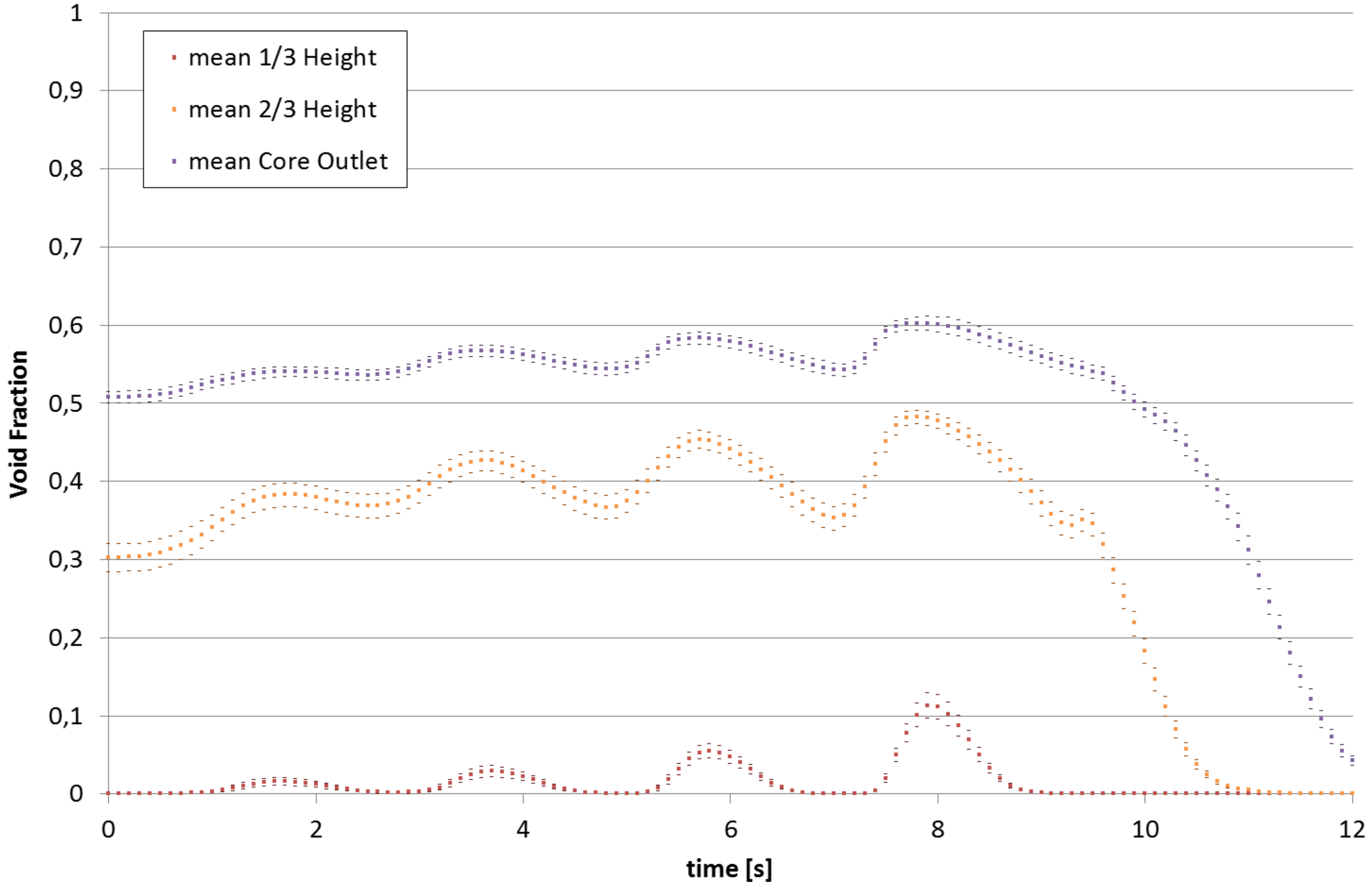
URANIE results in the zooming area

- Pearson sensitivity coefficients of the axial pressure drop



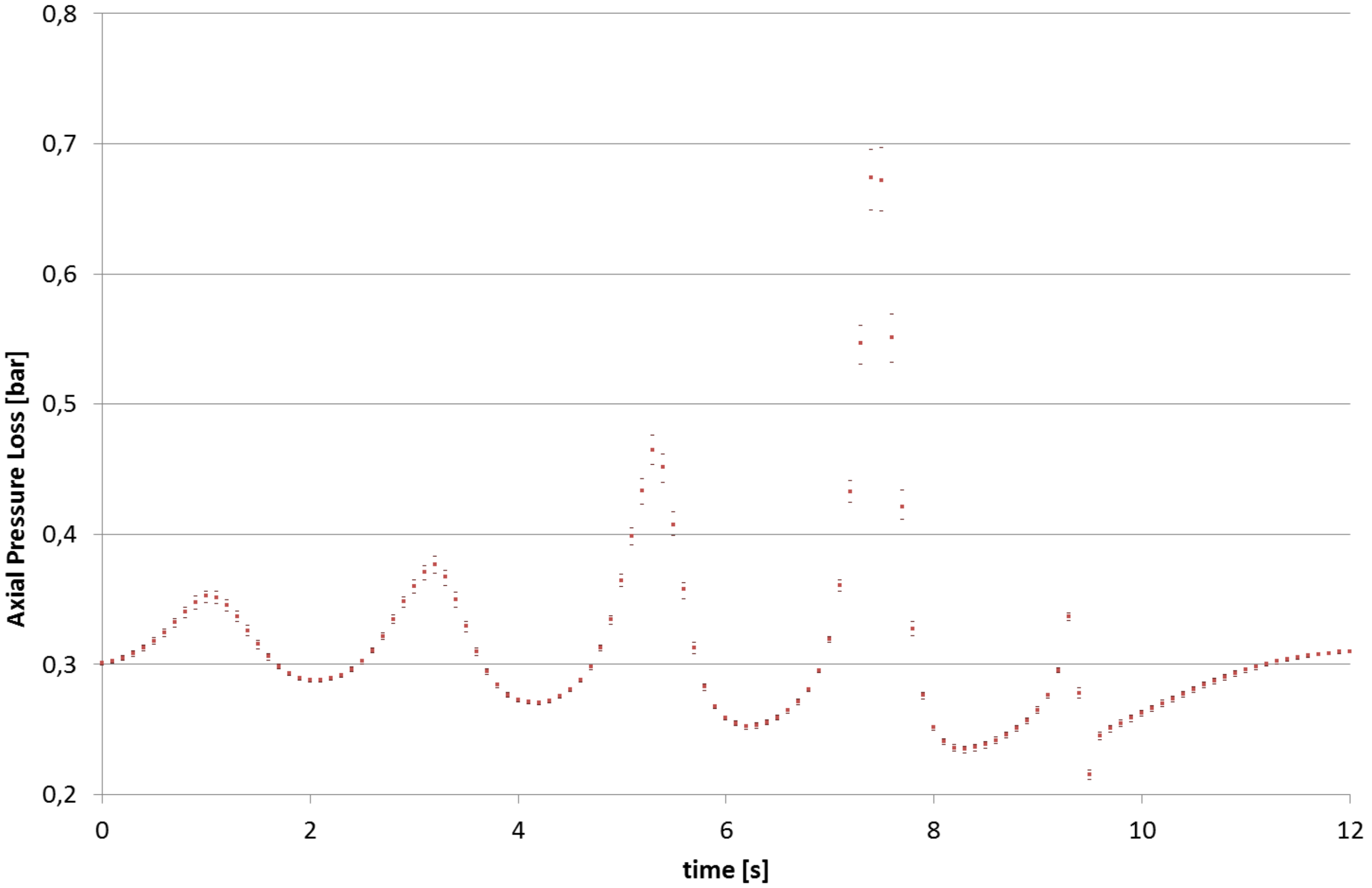
URANIE results in the zooming area

- Mean, min and max value of the void fraction at three different elevations: 1/3, 2/3 and exit



URANIE results in the zooming area

- Mean, min and max value of the axial pressure loss of the bundle average



SUSA Software system for Uncertainty and Sensitivity Analysis

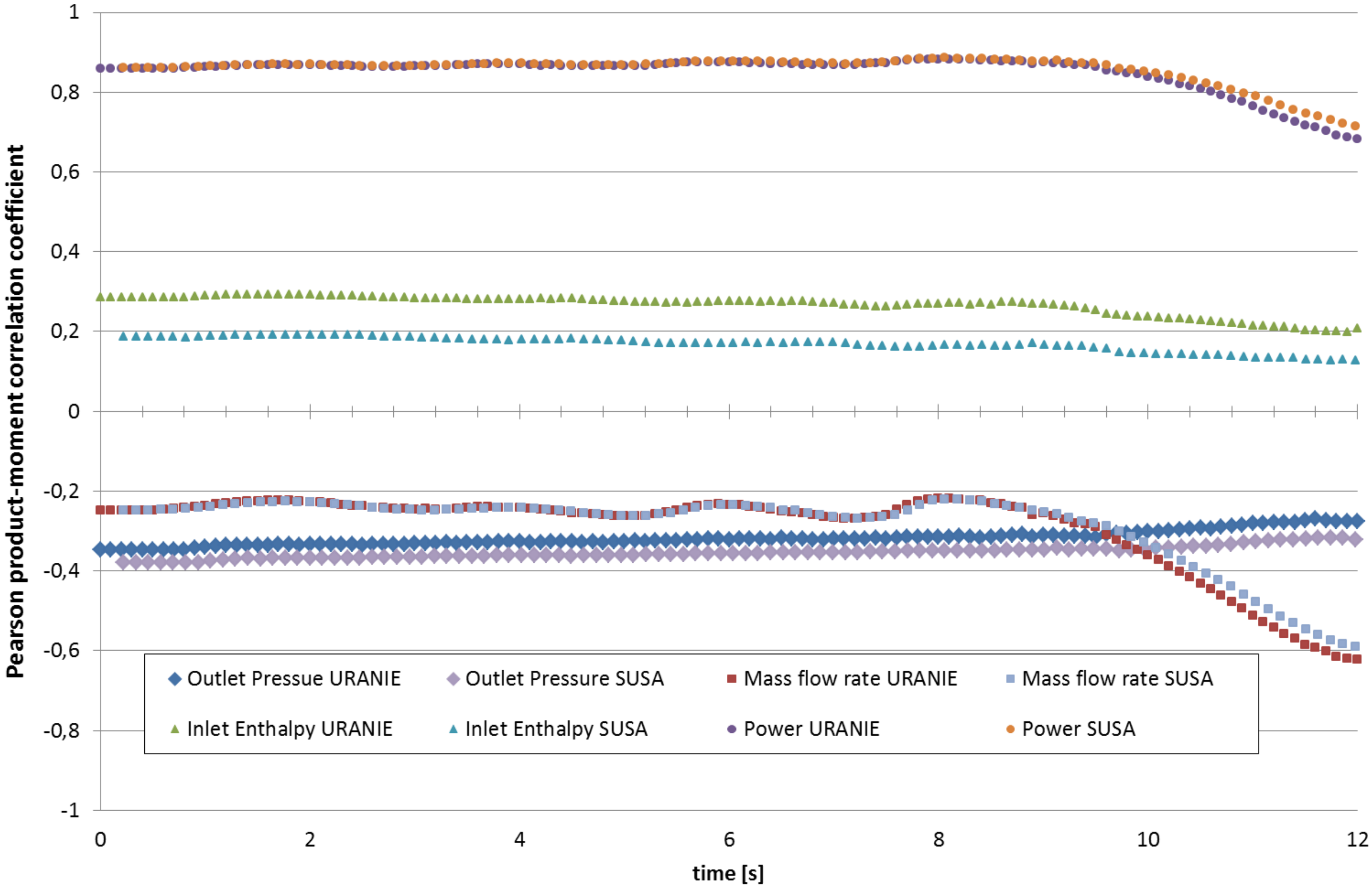
- Under development at GRS
- Probabilistic approach with input error propagation
- Code runs defined by Wilk’s formula
- Monte Carlo (MC) simple random and Latin Hypercube sampling (LHC)
- 4 sensitivity measures are obtanied
- Widely used in nuclear community

B. Krzykacs, E. Hofer and M. Kloos,” A software system for probabilistic uncertainty and sensitivity analysis of results from computer models”, Proceedings of the International Conference on Probabilistic Safety Assessment and Management, San Diego, California, USA, 1994.

R. Macian-Juan, “Uncertainty and sensitivity evaluation for Best Estimate coupled calculations,” FJOT Summer School 2011, Karlsruhe, Germany.

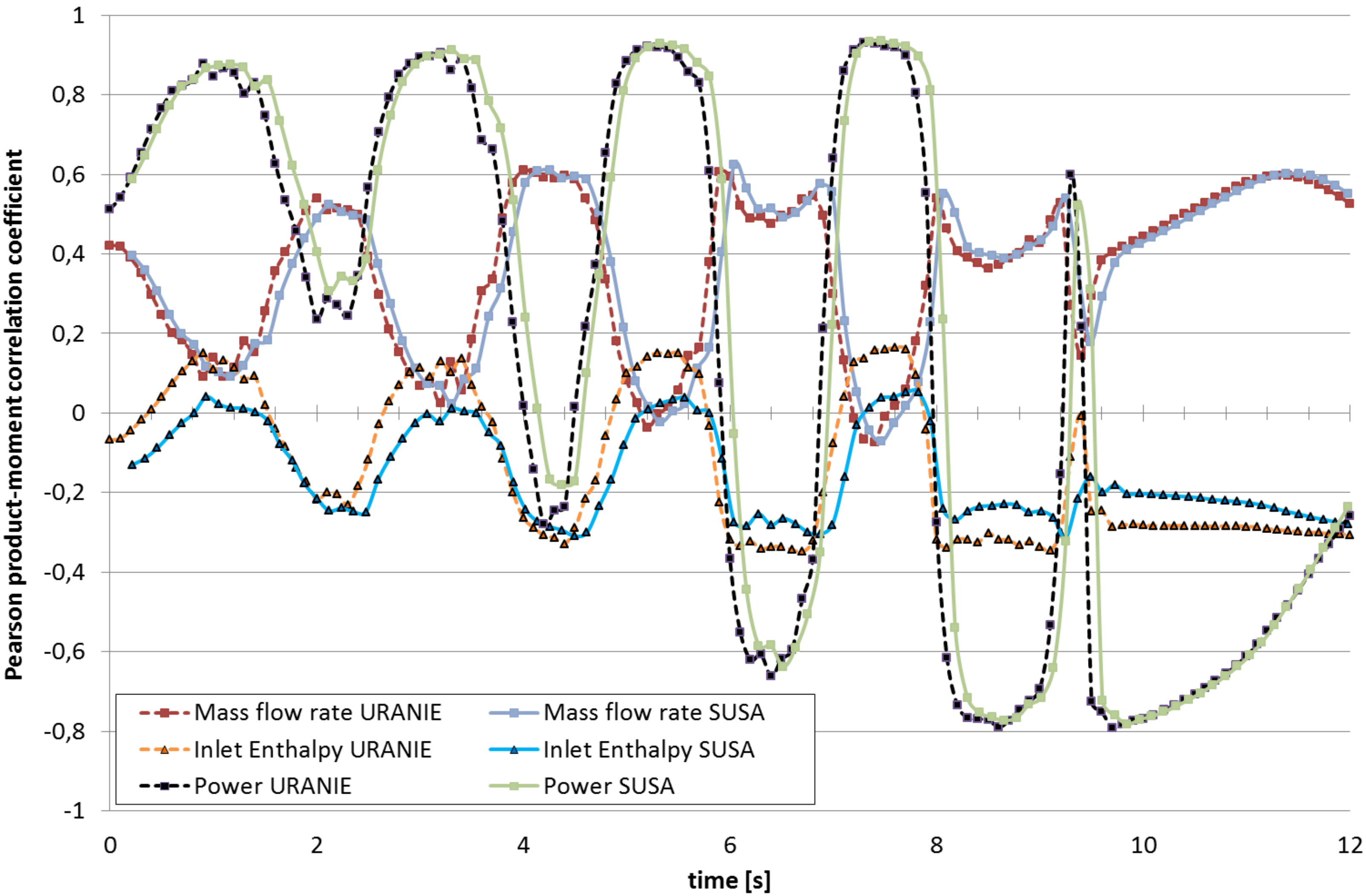
Comparison against SUSA

- Pearson sensitivity coefficients of the void fraction



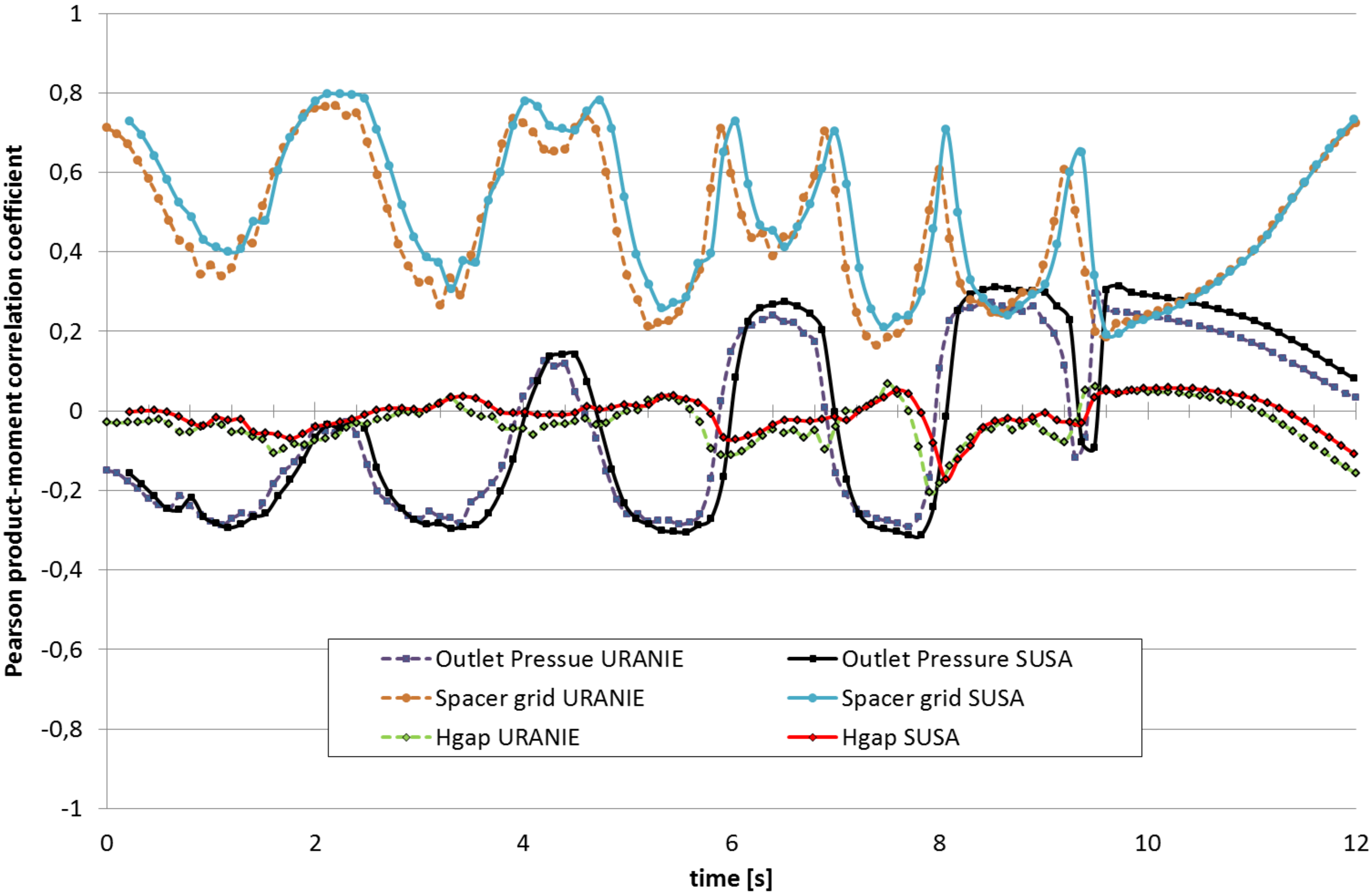
Comparison against SUSA

- Pearson sensitivity coefficients of the axial pressure drop



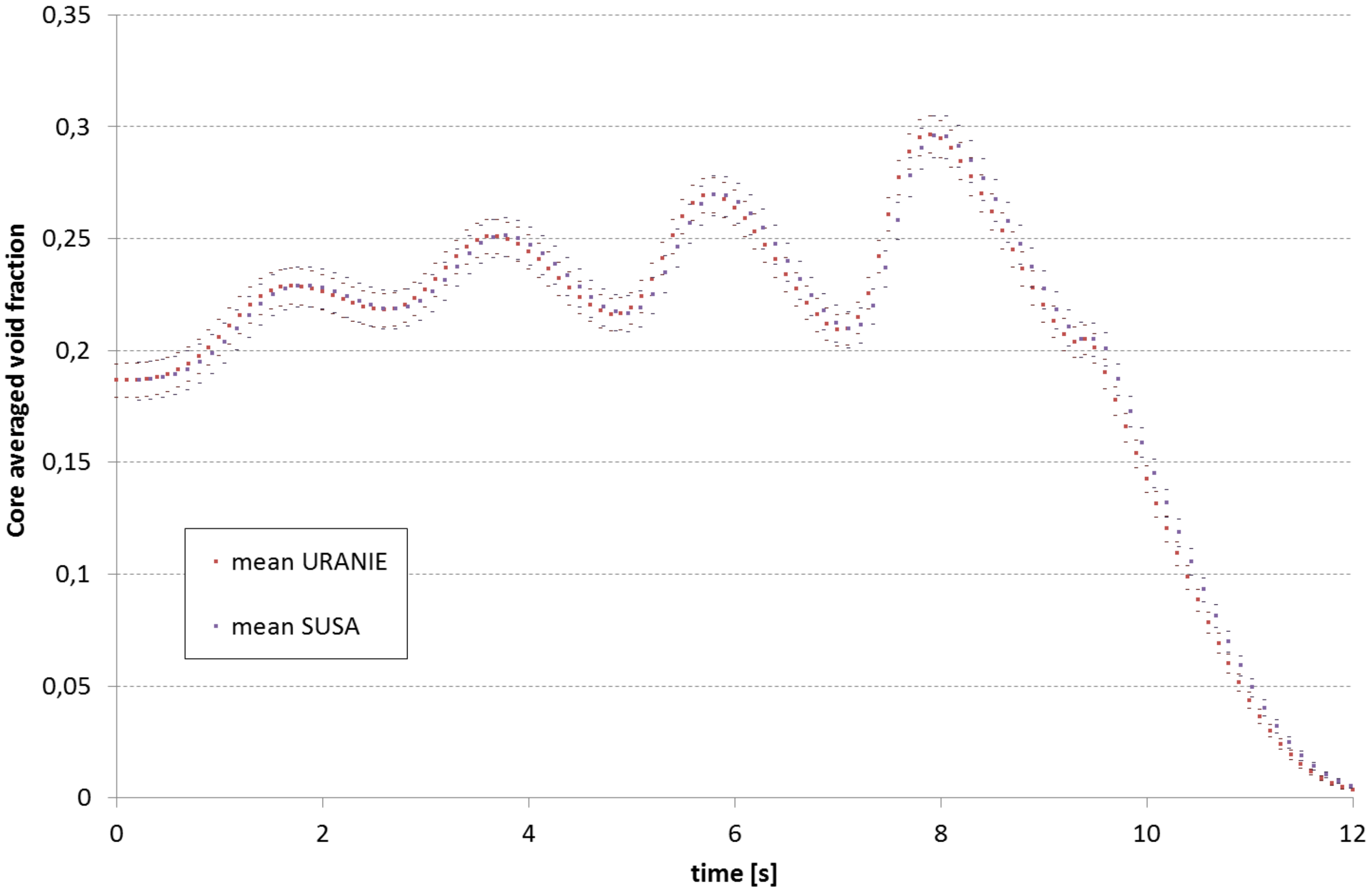
Comparison against SUSA

- Pearson sensitivity coefficients of the axial pressure drop



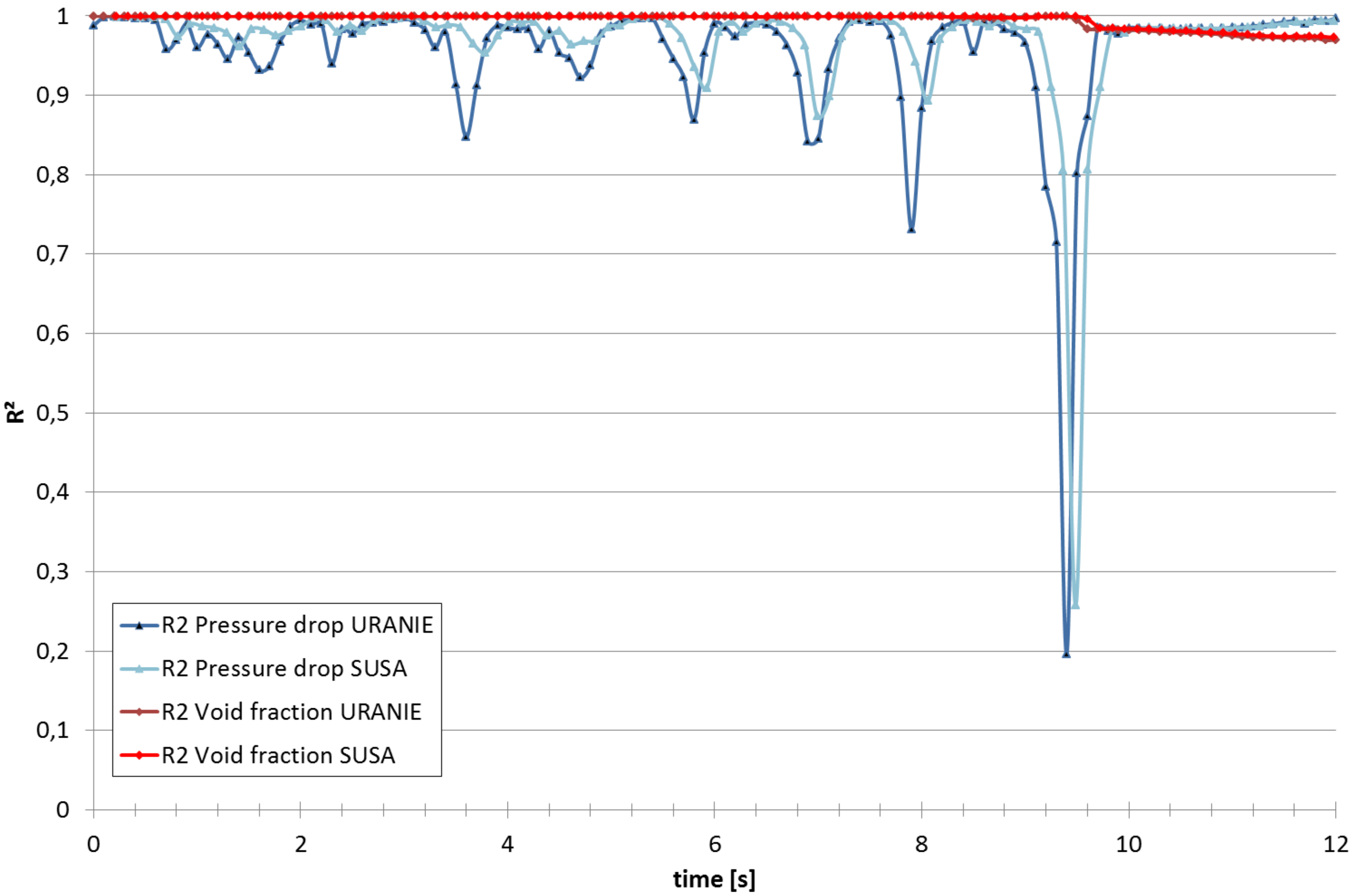
Comparison against SUSA

- ## Core averaged void fraction



Comparison against SUSA

- The experiment is well covered, R²≈1.0



Conclusions and Outlook

- During the first 18 months of the project, investigations on the use of URANIE platform for sensitivity analyses have been conducted.
- Studies using the COBRA-TF code on steady state and transient simulations were carried out.
- Satisfactory results, high degree of flexibility in the URANIE scripts. Similar results to SUSA.
- The scripts can be extrapolated to any code with input text files: FLICA4, DYN3D, COBAYA3, ATHLET, etc, ...

FUTURE WORK

- Application to coupled simulations is foreseen in the next months.
- Uncertainty in model parameters could be considered
- Preparation of a paper to be submitted to NUTHOS-10

THANKS FOR YOUR ATTENTION

Questions?