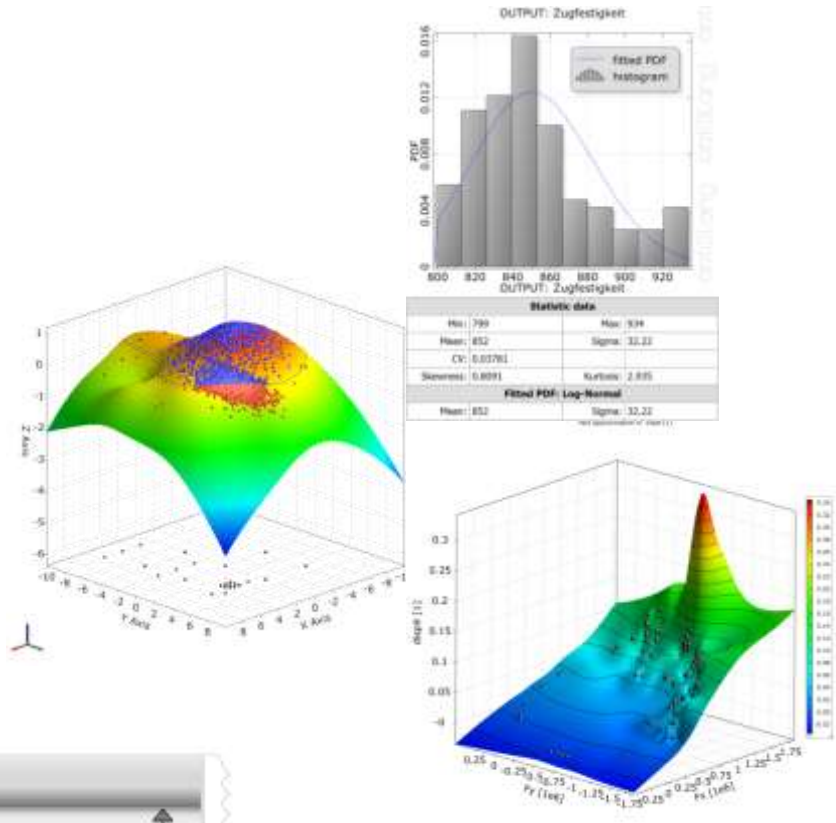
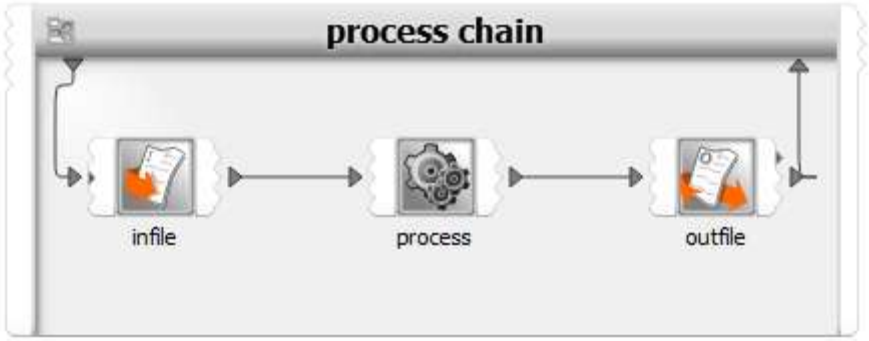
optiSLang v4

v3 Compatibility

- **Support v3 problem definition**
  - **Load or**
  - **Double - click .pro**

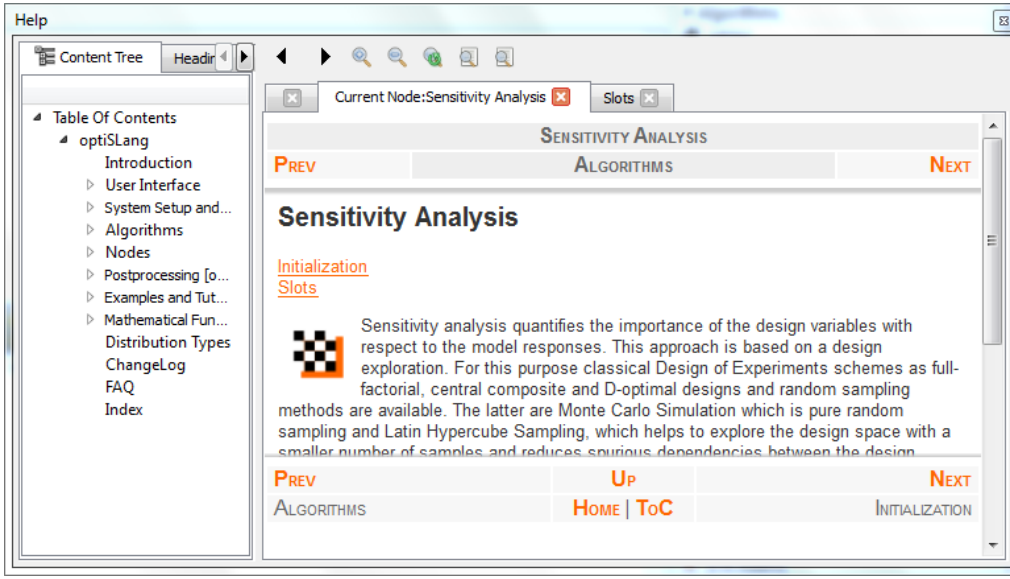
- **Postprocessing**
  - **v4.0 uses v3 postprocessing**
  - **Double - click .bin**

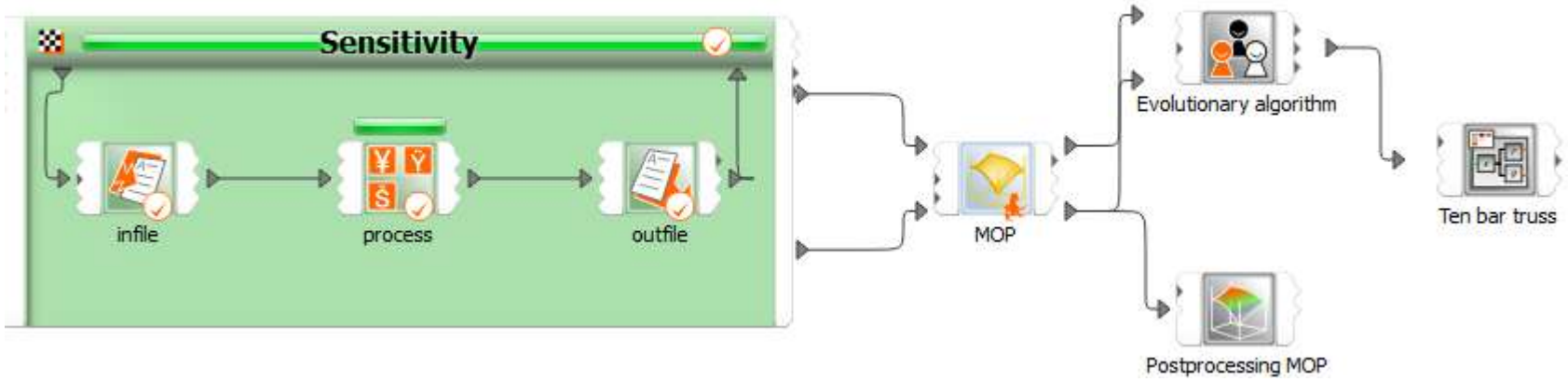




Documentation

- **Tutorials**
- **Examples**
- **Supportmail**
- **Method documentation**
- **Context sensitive help**
  - **Tabs**
  - **Search**





Wizards

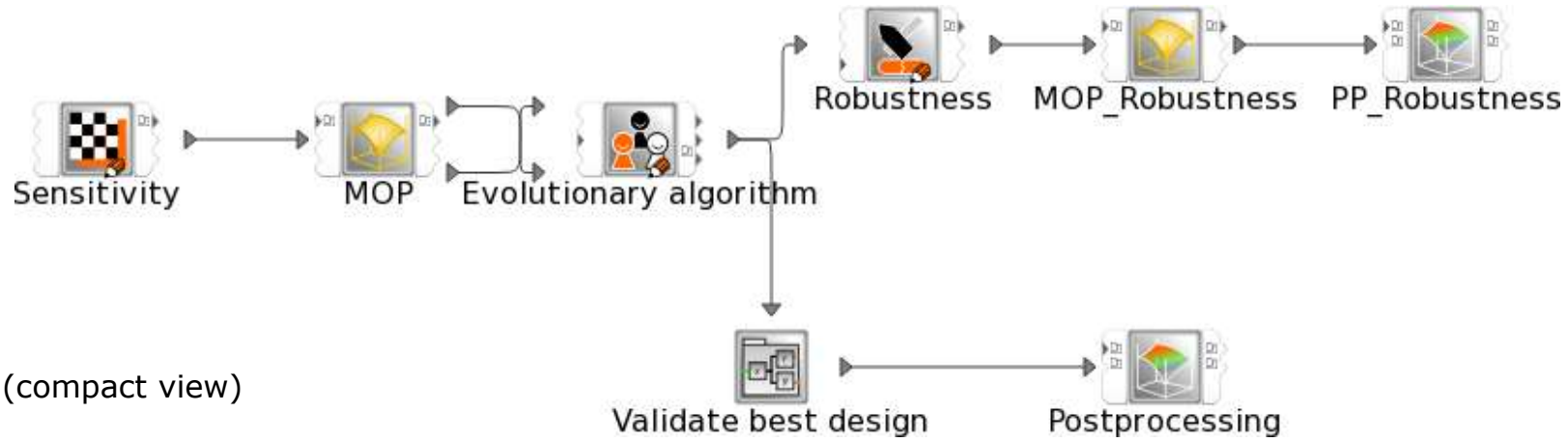
- **Fast and easy creation of:**
  - **Process chain**
  - **Sensitivity analysis**
  - **Optimization task**
  - **Robustness evaluation**

- **Supported by:**
  - **Principle of wizards**
  - **Defaults**
  - **Decision help**

Wizard based flow creation

- **Use templates**
- **Drag & Drop**
- **Automatic connect (Parameter, Designs, ...)**

- **Algorithms are nodes with inputs and outputs**

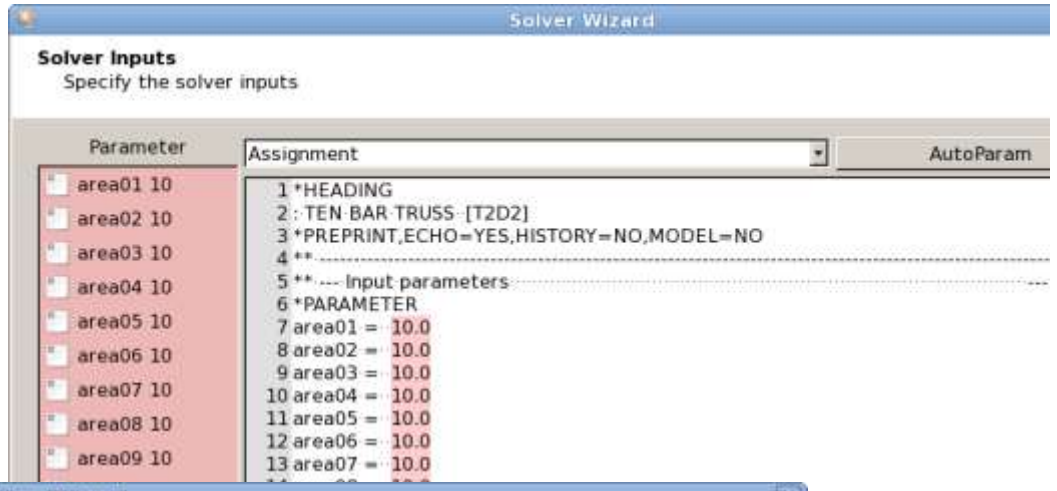


(compact view)

Input parametrization

- **Automatic detection**
  comfortable for a lot of parameters

- **Parameter table**
  for fast and easy modification

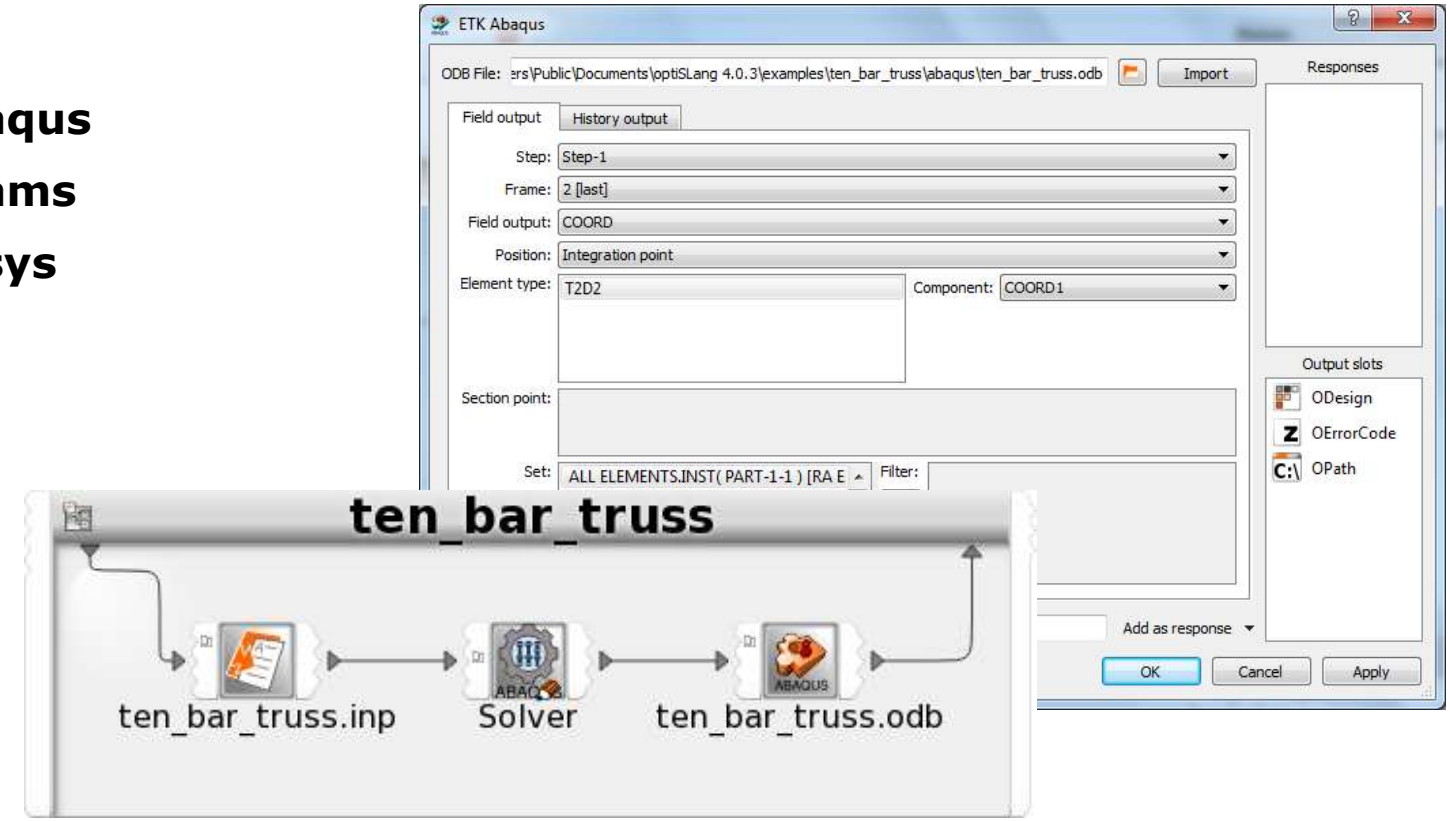


Output extraction with ETK

Comfortable extraction of known output file formats
To be integrated in process chain

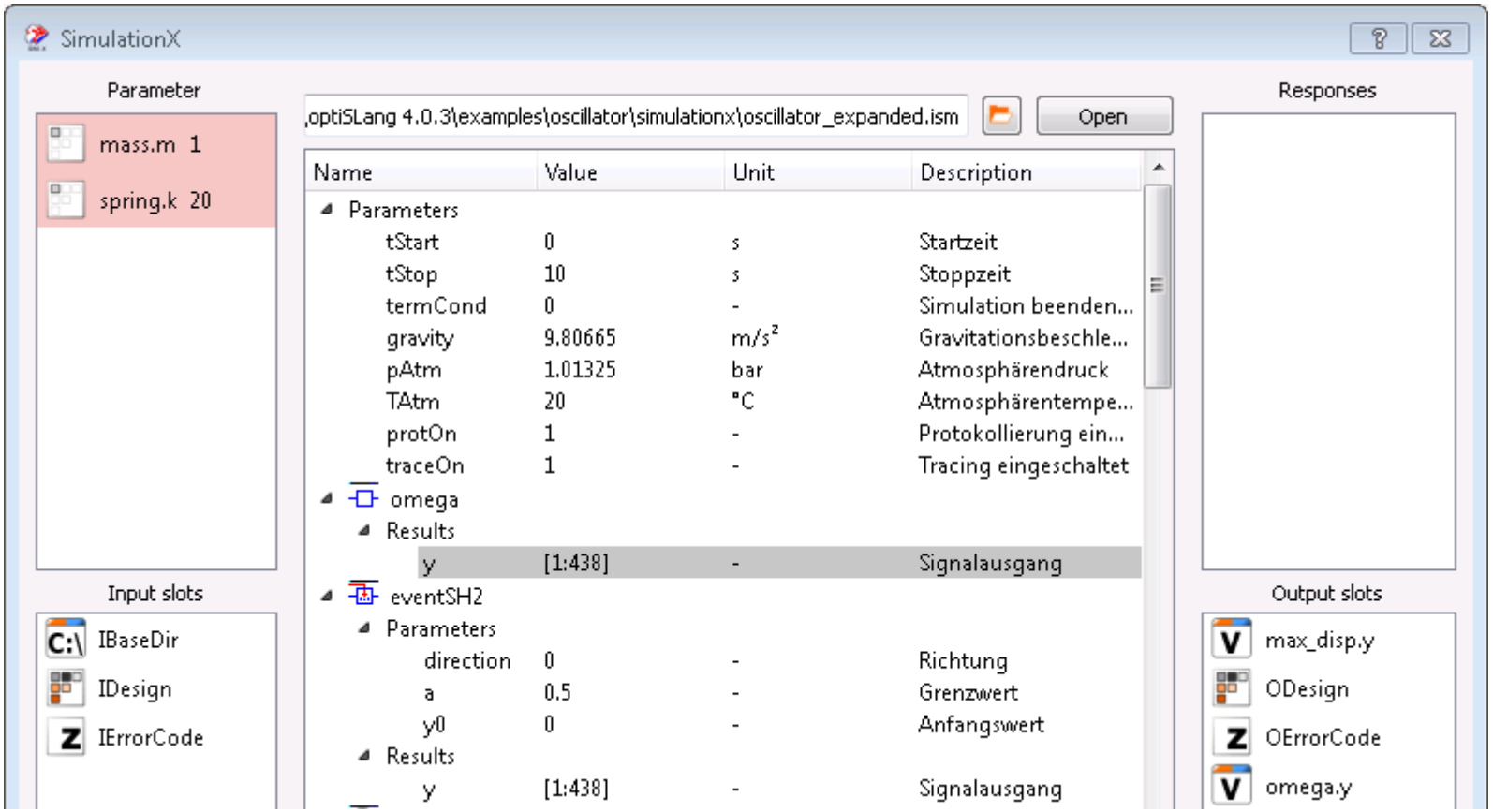
Nodes for:

- **Abaqus**
- **Adams**
- **Ansys**



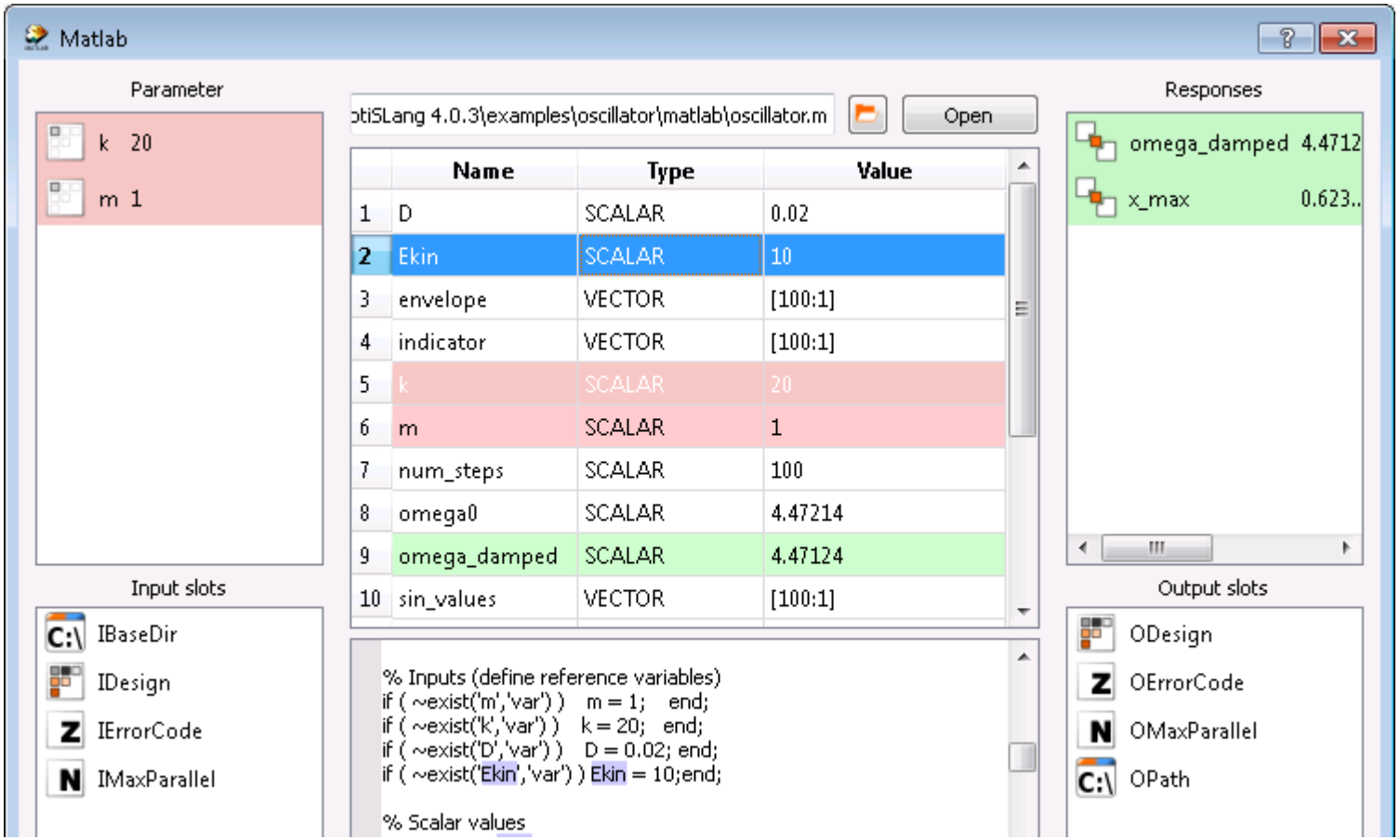
Simulation X

Solver specific dialog and execution



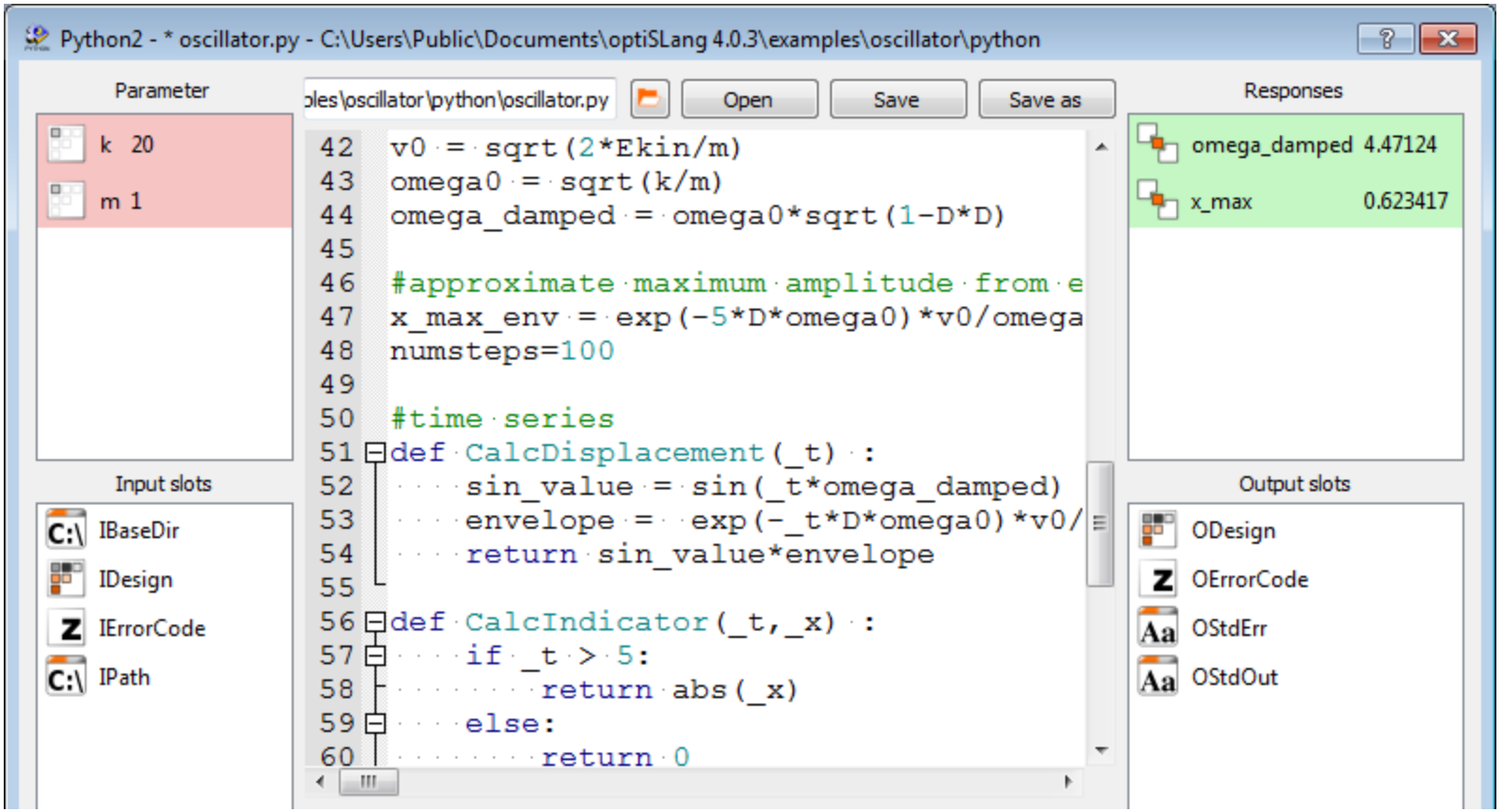
Matlab

Solver specific dialog and execution



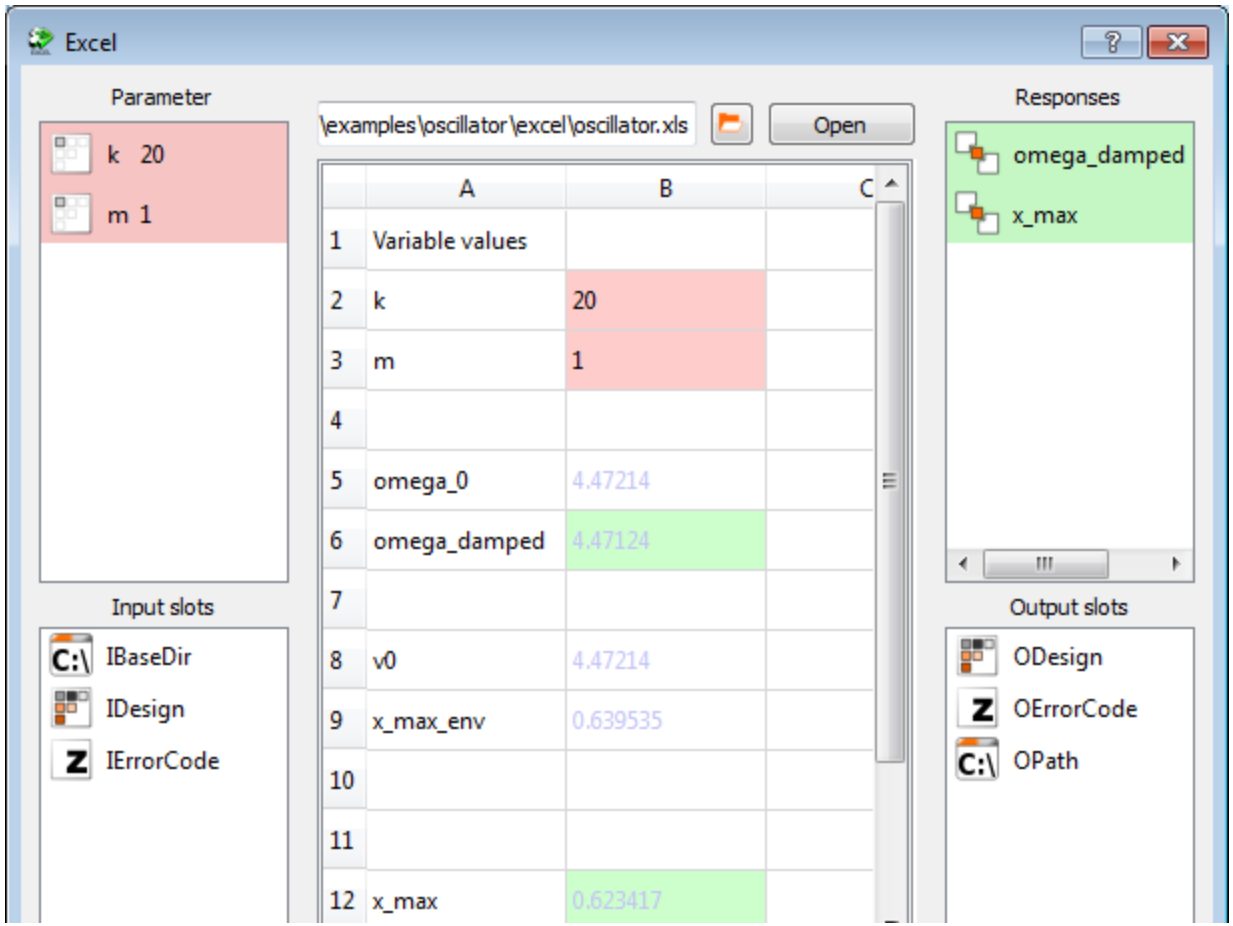
Python

Solver specific dialog and execution



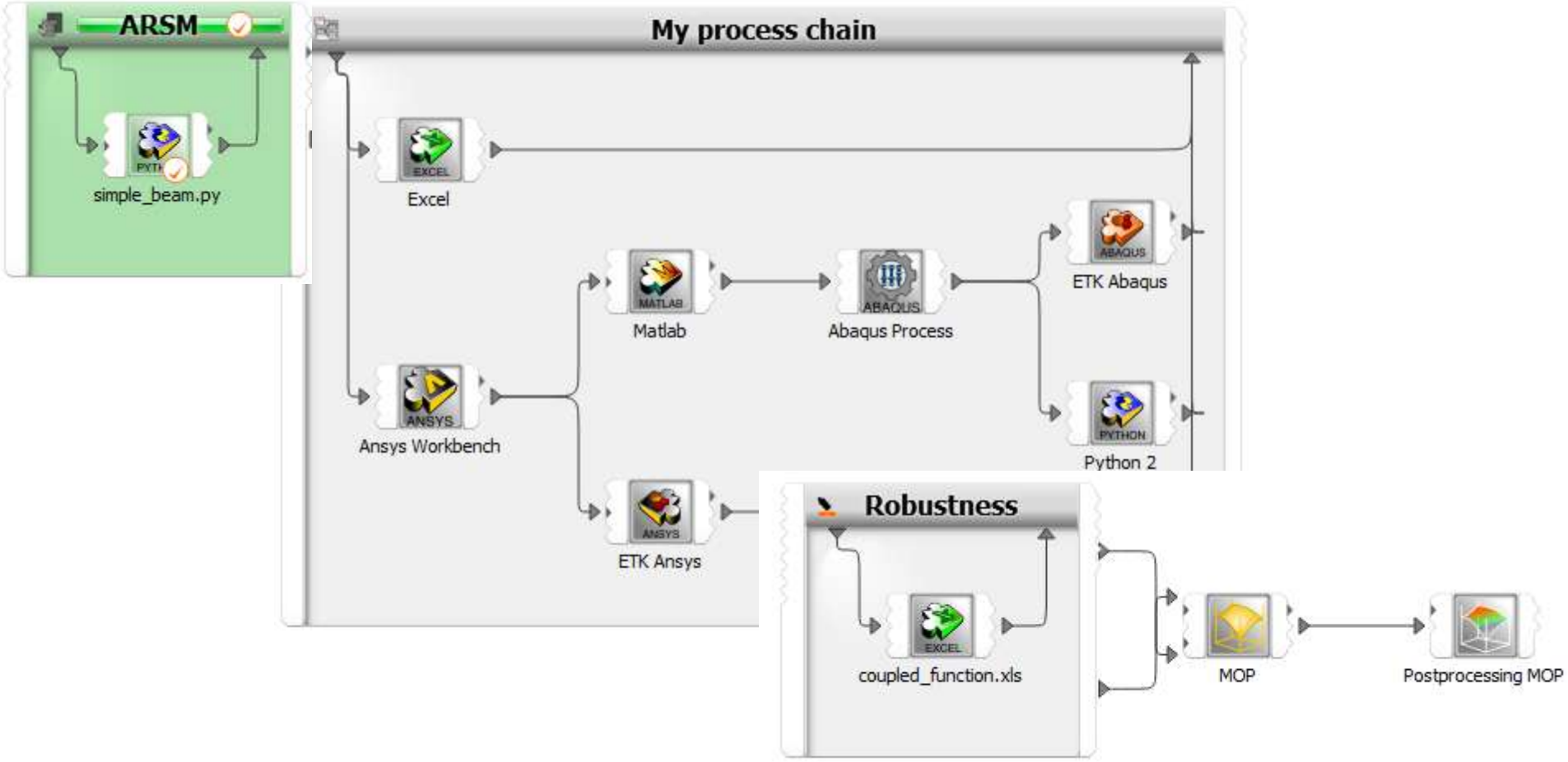
Excel

Solver specific dialog and execution



Integrations

Appear as nodes with inputs and outputs



Command Line Interface (CLI)

- **Create or modify a project**
  - **In GUI**
  - **With python script**

- **Run the project**
  - **In batch**
  - **No graphical environment needed**

optislang – – batch myproject.opf

Python modules and C++ libraries

Use

- **DOE**
- **Robustness evaluation**
- **Optimization algorithms**
- **MOP / MOP – Solver**

In

- **External Code (Matlab, Simplorer, …)**
- **Script / own application**
- **Customized Application**

Through

- **Import of dynardo Python modules**
- **Use of dynardo C++ libraries (.dll, .so)**

Release

optiSLang inside Ansys Workbench

ACUM

26. 10. 2012

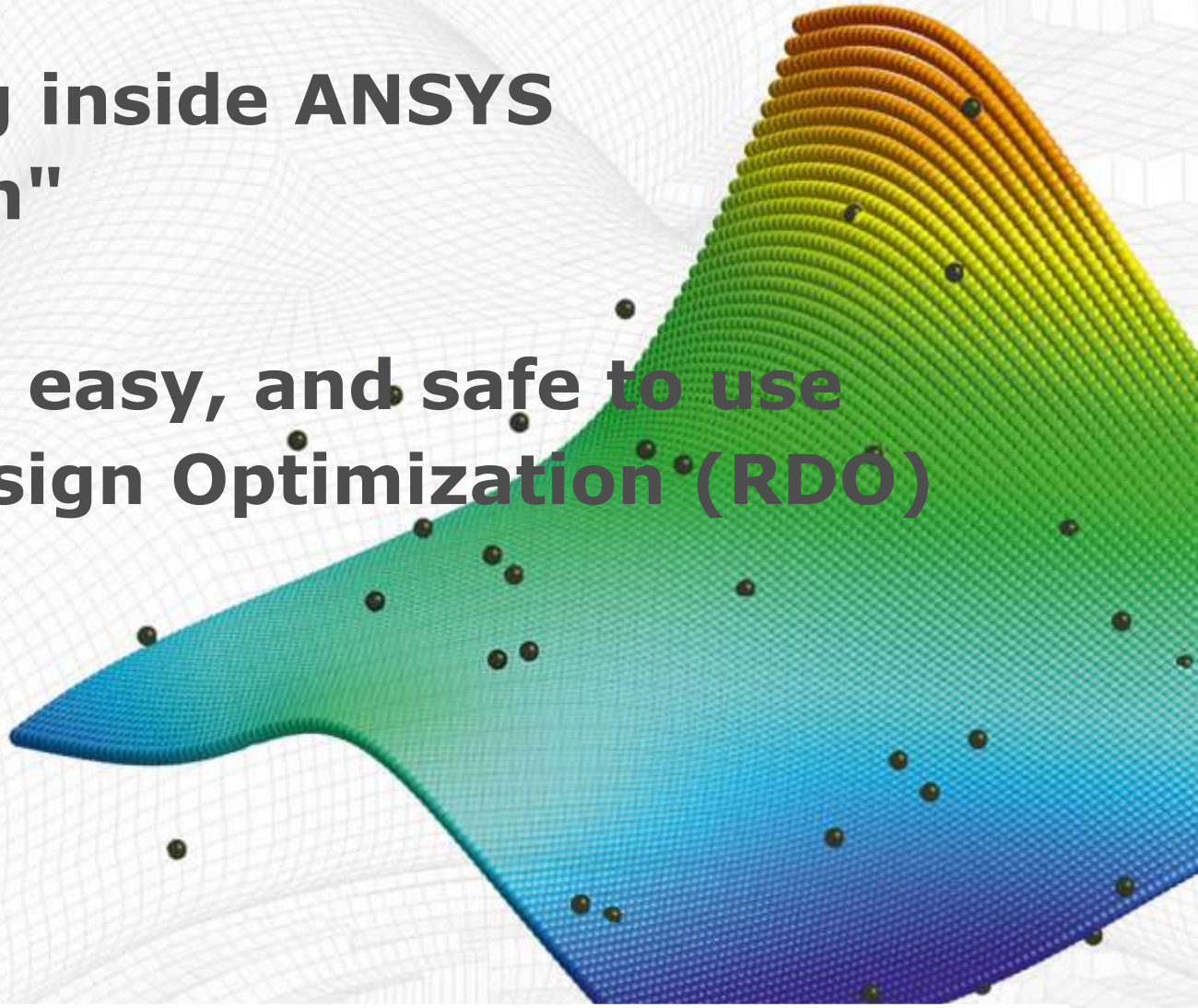
optiSLang v4

WOST 9.0

27. 11. 2012

"optiSLang inside ANSYS Workbench"

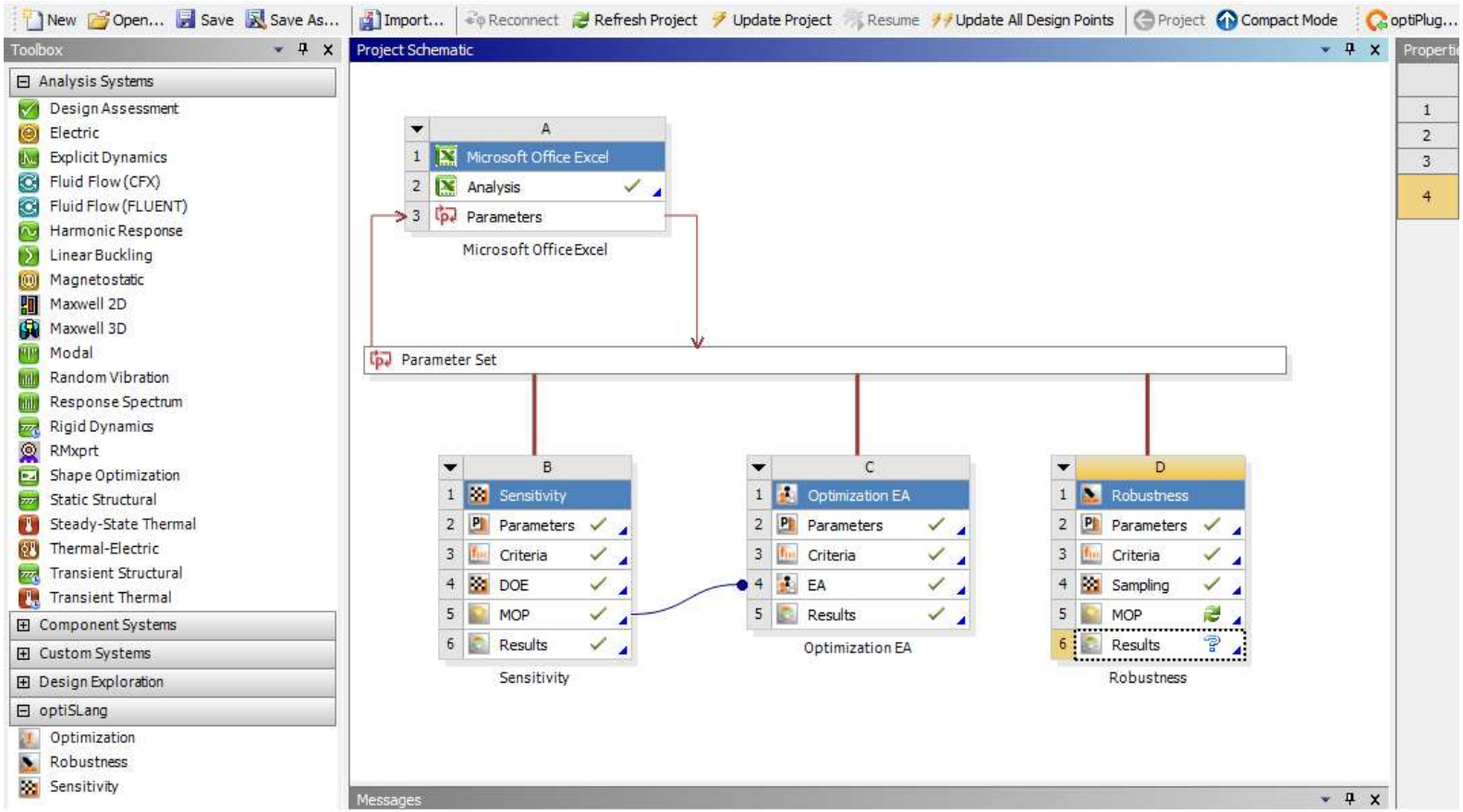
– efficient, easy, and safe to use Robust Design Optimization (RDO)



Dynardo GmbH

optiSLang inside ANSYS Workbench v14

Modules Sensitivity+MOP, Optimization and Robustness+MOP provide „best practise" optiSLang functionality

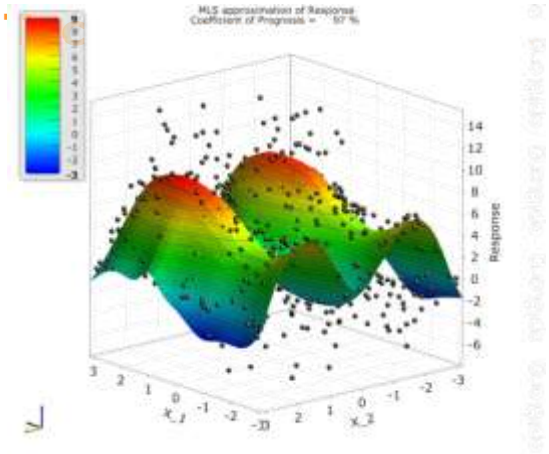


optiSLang Flows of best Praxis

Safe to use.

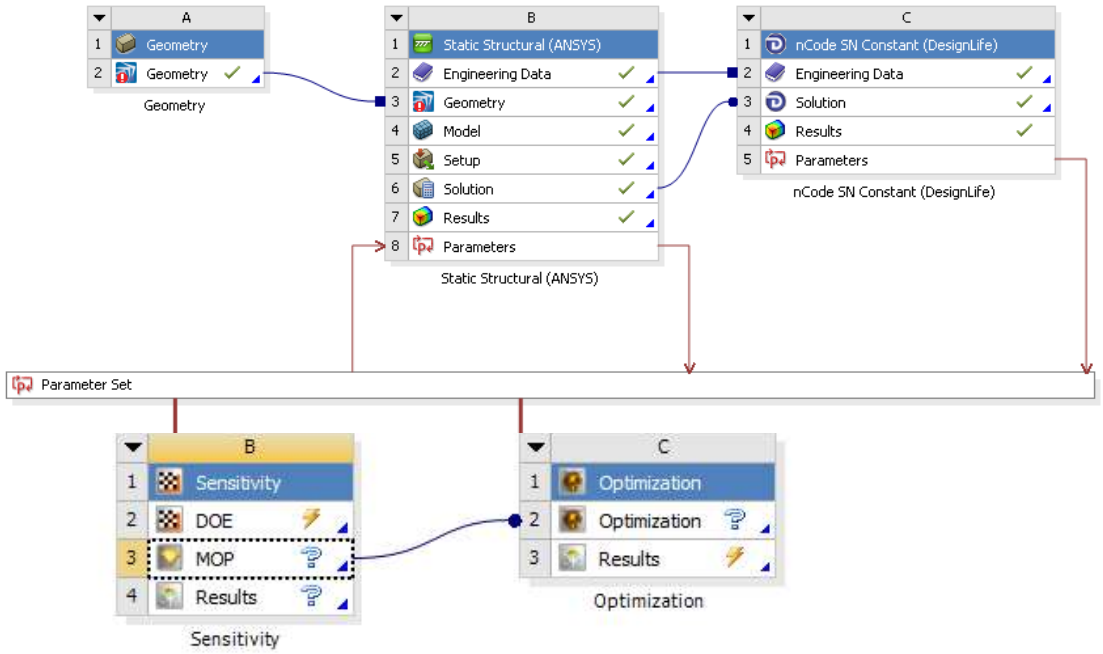
- automate best practice to „black box" flows
- minimize the risk to miss better designs (optimization)
- minimize the risk to estimate misleading measures for robustness, safety and reliability
- offer easy to use measurements of (response variation) prognosis quality

That task requires sophisticated technology with carefully balance between number of solver calls and safety to reach the RDO goal.



that "non expert" can use it!

optiSLang inside ANSYS Workbench



ANSYS Workbench parametric set up of complex simulations

Easy to use:

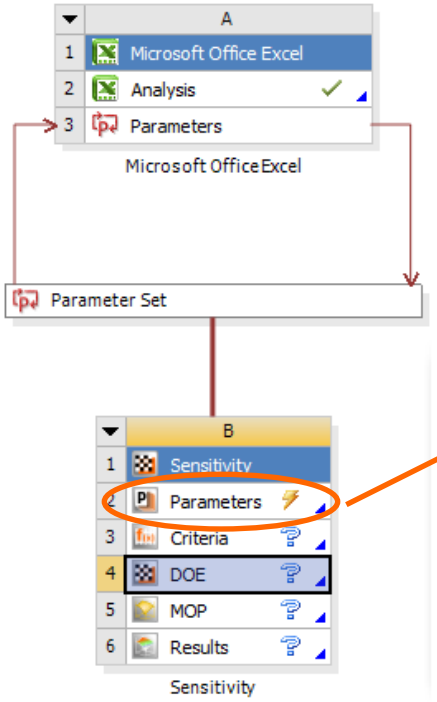
- minimize user input
- offer best practise defaults for modules
- offer pre defined post processing modes

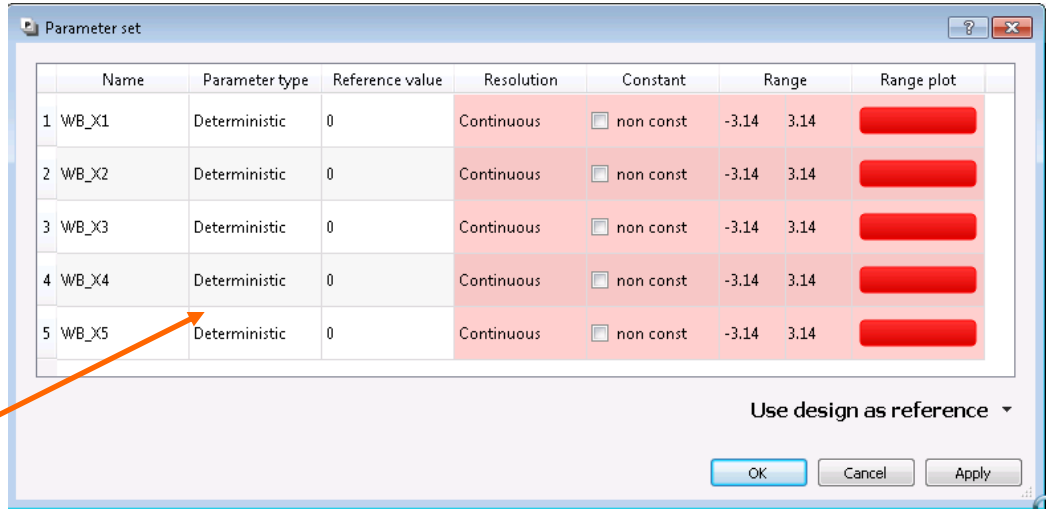
User-friendliness takes care that it will be used!

Sensitivity Module

Minimal required user input:

- Definition of parameter variation





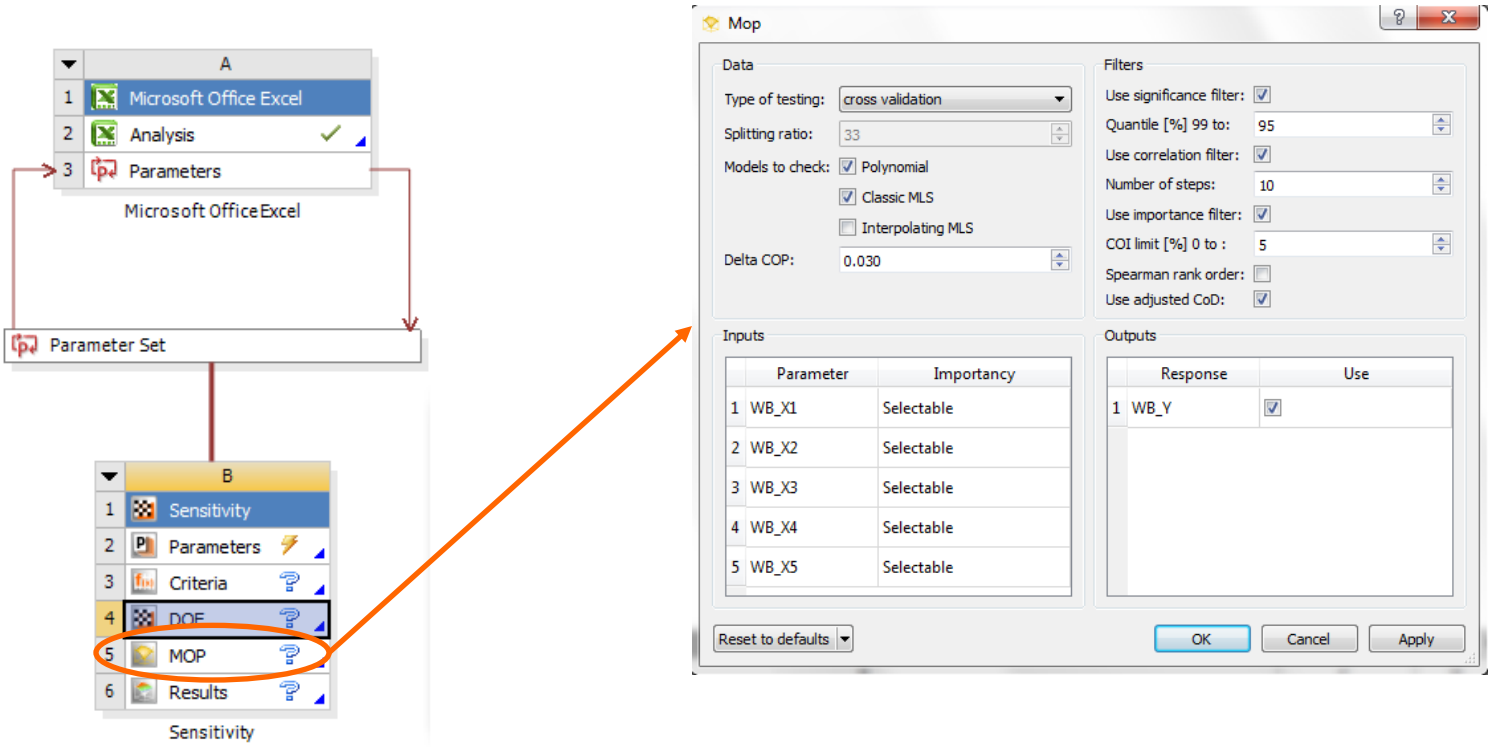
Sensitivity Module

The Meta Model of optimal Prognosis (MOP) is automatically created out of the DOE-Sampling

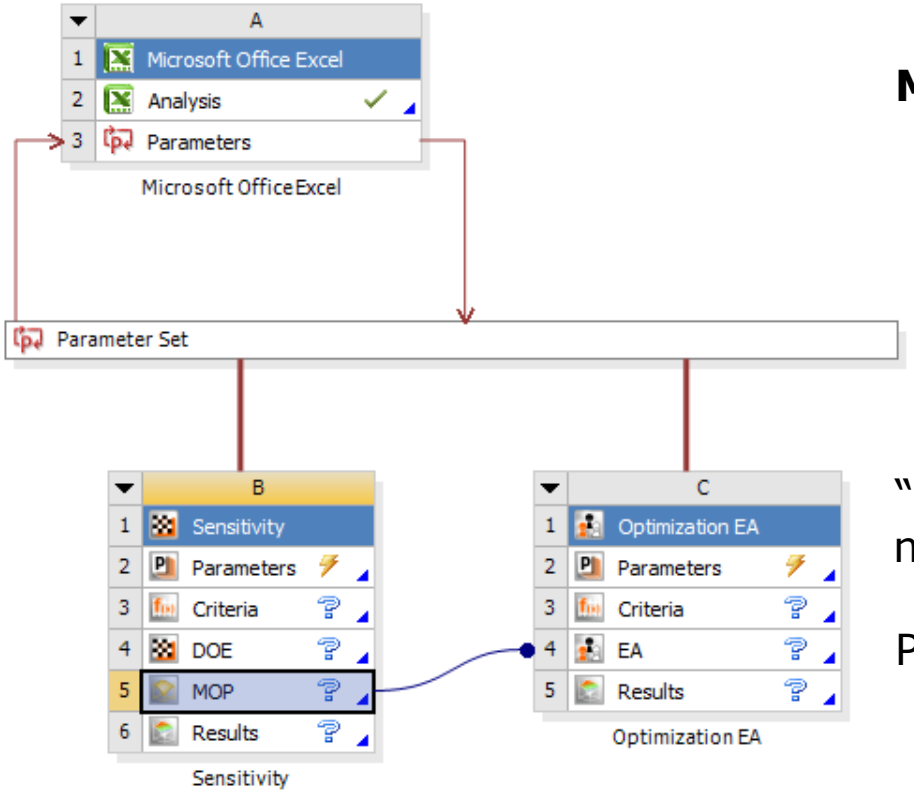
Minimal required user input: non

Additional features:

- supports removing designs out of DOE Post Processing



Optimization using MOP



After sensitivity optimization using of MOP is supported.

Minimum required user input:

- drop the optimization module onto MOP
- defining objective and constraints

"Optima" which are based on meta models need to be verified!

Proof optima:

- Automatic verification with real ANSYS call
- Check differences in post processing

Optimization Wizzard

optiSLang helps you to select a suitable optimization algorithm. Support the underlying (automatic) selection process with some additional information about the solver and the problem itself.

Exampel for using MOP and best_design_Sensitivity:

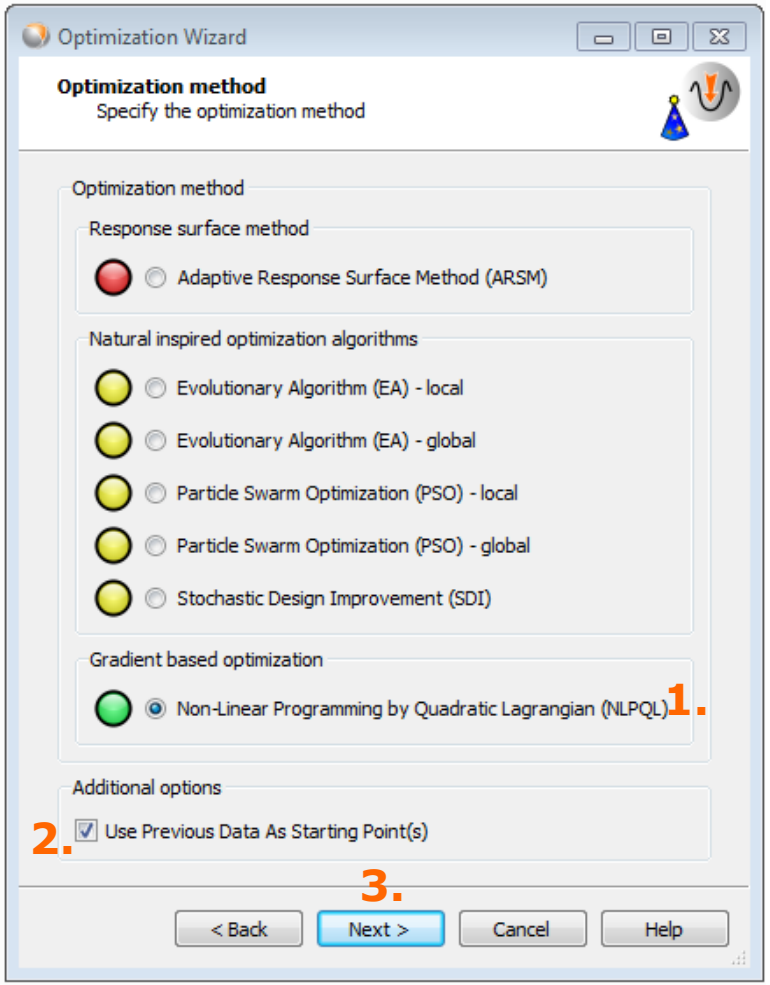
1. Set the analysis status as “Preoptimized” (best design from Sensitivity)
2. Set the constraint violations to “Seldom”
3. Set failed designs to “None” (MOP gives always response values)
4. Set solver noise to “None” (MOP gives a smooth surface)

Optimization Wizzard using MOP

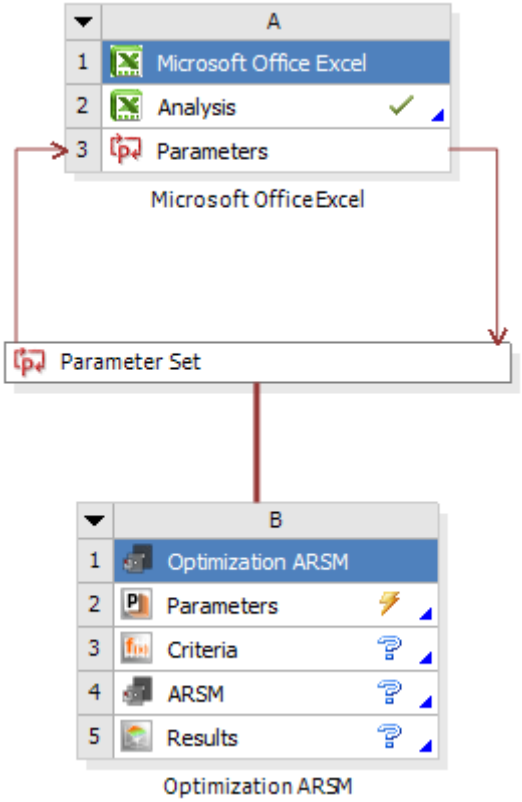
Suggested algorithm is NLPQL

Start point is automatically selected

Press “Next”



Optimization with real design calls



After sensitivity und optimization on MOP the user can continue with gradient-based, NOA-based optimization or ARSM optimization.

Minimum required user input:

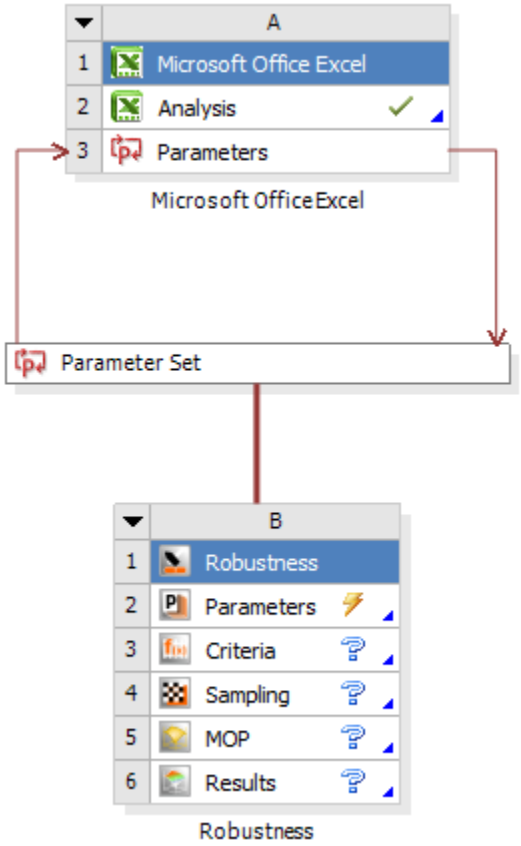
- define objectives and constraints
- choose method (Gradient-based including start design, NOA-based including best designs's out of sensitivity/MOP, ARSM in the domain of the most important optimization parameter)

For all optimizer robust default settings are provided.

NOA - Nature inspired optimization contains: evolutionary, genetic, particle swarm optimization

ARSM – Adaptive Response Surface Method

Robustness Evaluation



Minimum required user input:

- definition of input variation /scatter
- definition of robustness criteria
- number of samples for ALHS

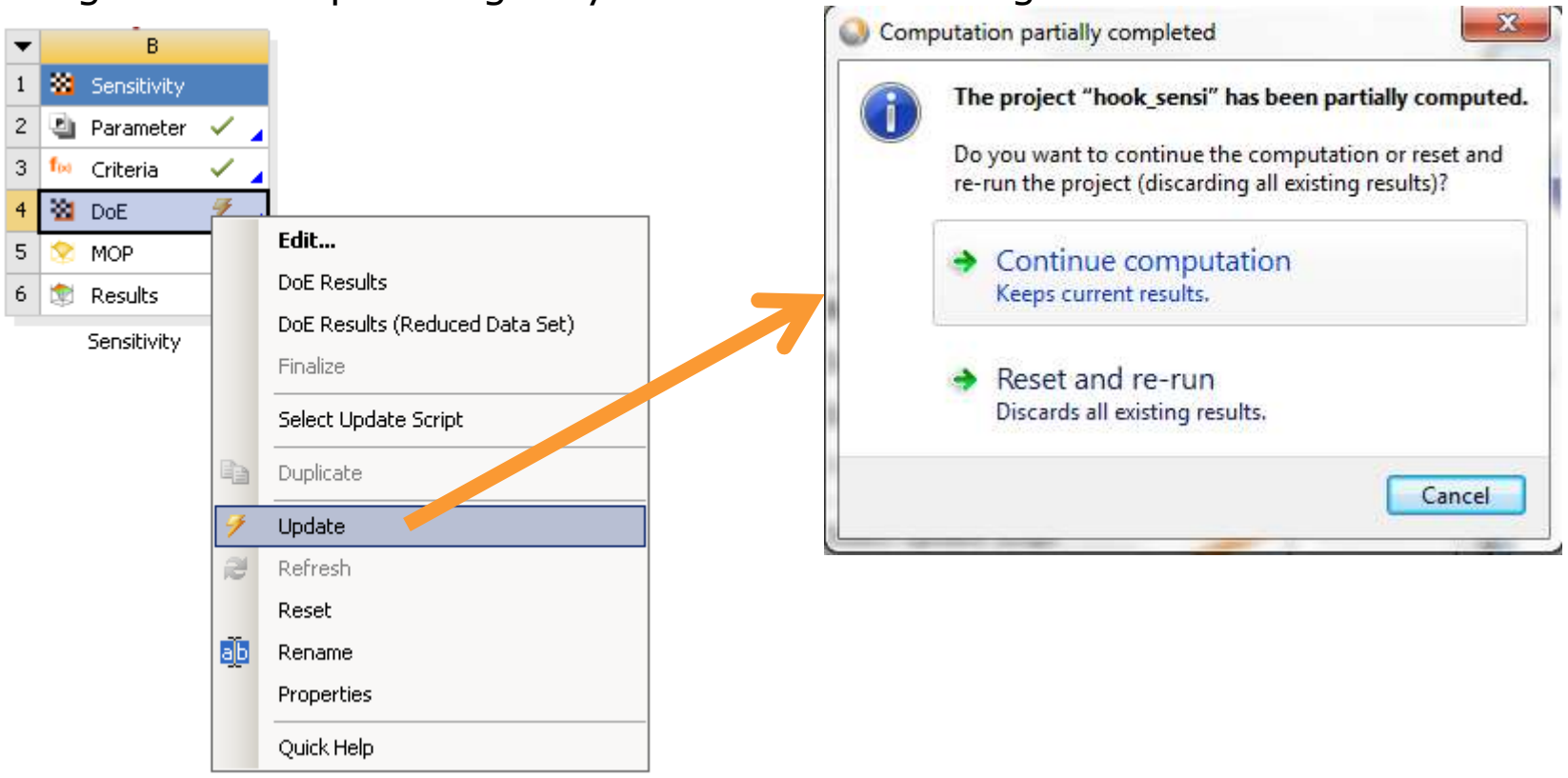
Update Functionality

Continue crashed session option inside ANSYS

optiSLang saves every design which was successfully calculated

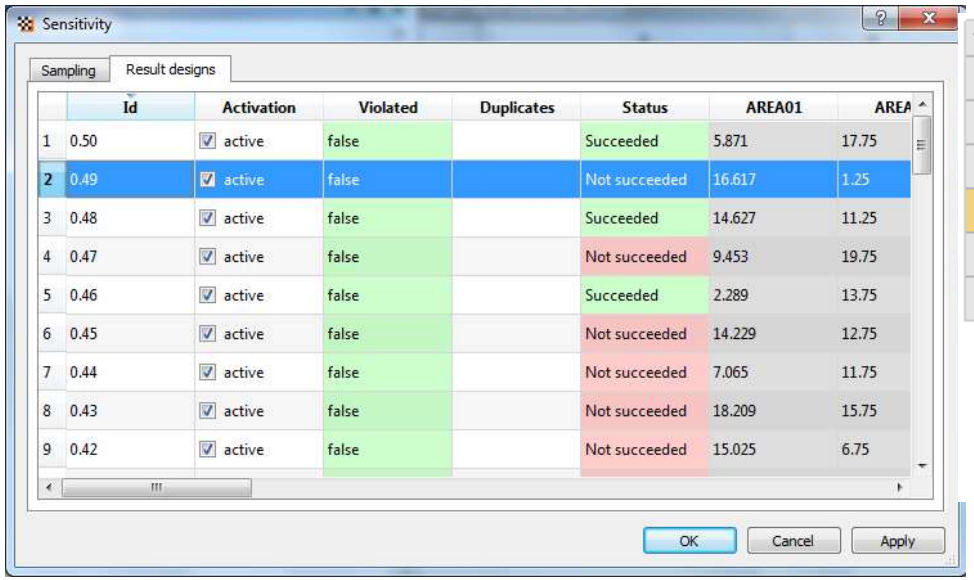
using update at optiSLang container continue or reset can be chosen

using continue optiSLang only send unsolved designs

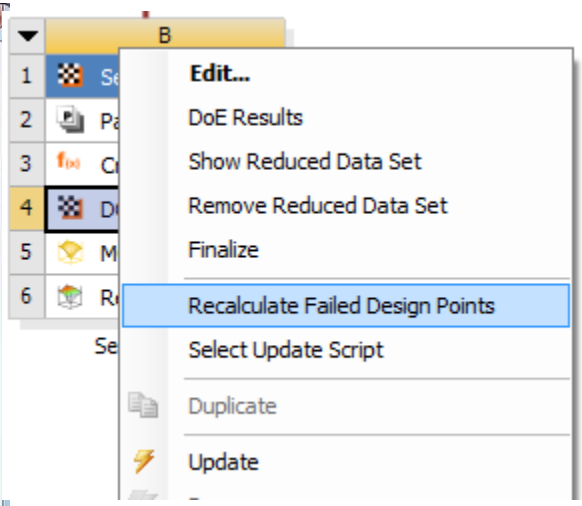


Recalculate failed designs

- Due to different reasons design evaluations may fail
- With “Recalculate Failed Design Points” you can start them again



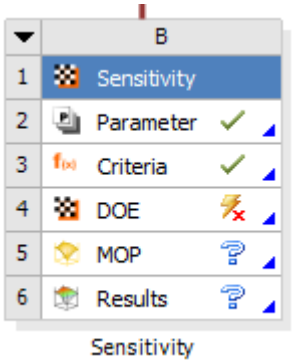
	Id	Activation	Violated	Duplicates	Status	AREA01	AREA
1	0.50	active	false		Succeeded	5.871	17.75
2	0.49	active	false		Not succeeded	16.617	1.25
3	0.48	active	false		Succeeded	14.627	11.25
4	0.47	active	false		Not succeeded	9.453	19.75
5	0.46	active	false		Succeeded	2.289	13.75
6	0.45	active	false		Not succeeded	14.229	12.75
7	0.44	active	false		Not succeeded	7.065	11.75
8	0.43	active	false		Not succeeded	18.209	15.75
9	0.42	active	false		Not succeeded	15.025	6.75



Interrupt, save, send & continue

- Stop your analysis
  - At the end of the day
  - If licenses are not available
  - …
- Save the Workbench project
- Continue later

- saves every successful design run!
- external optimization using MOP possible!



Update via Python scripting

- ANSYS initialize per default an update mechanism, which updates a complete ANSYS Workbench project
- Mechanism can be overridden via python file
- optiSLang provides this feature for optiSLang design evaluations
- user has full access to his ANSYS model update

Properties of Schematic B4: DoE

	A	B
1	Property	Value
2	⊟ General	
3	Component ID	DoE (optiSLang)
4	Directory Name	Sensitivity
5	Open Postprocessing during Solver Run	☐
6	Save Design Point Directories	☑
7	⊟ Update Options	
8	Run Python Script for Update	☑
9	Python Script:	C:\Users\Verwalter\Desktop\Python Scripts\update.py

Parallel evaluation using Ansys RSM

- ANSYS RSM is the powerful tool to distribute jobs
- optiSLang can fill the Workbench design table with a predefined number of designs
- ANSYS RSM organizes distribution of jobs

- If ANSYS RSM is installed you only need to:
  - Choose RSM Mode
  - Set max. number of parallel jobs

Properties of Schematic B4: DOE

	A	B
1	Property	Value
2	General	
3	Component ID	DOE (optiSLang)
4	Directory Name	Sensitivity
5	Open Postprocessing during Solver Run	☑
6	Open Postprocessing after Calculation	☐
7	Save Design Point Directories	☐
8	Update Options	
9	Use RSM Mode	☑
10	Preferred Number of Design Points in Parallel	25

ANSYS HPC Parametric Pack
optiSLang inside Ansys Workbench v14.5

optiSLang Algorithm Settings

- **Select "Use RSM Mode" to enable parallel design point submission**
- **Set the "Preferred Number of Design Points in Parallel" to the intended RSM job size**

Properties of Schematic B4: DOE

	A	B
1	Property	Value
2	General	
3	Component ID	DOE (optiSLang)
4	Directory Name	Sensitivity
5	Open Postprocessing during Solver Run	☐
6	Open Postprocessing after Calculation	☐
7	Save Design Point Directories	☑
8	Notes	
9	Notes	
10	Update Options	
11	Use RSM Mode	☑
12	Preferred Number of Design Points in Parallel	4