



dynardo

dynamic software & engineering

optiSLang 5

David Schneider

Product manager

1. Postprocessing

2. optiSLang for ANSYS

6. Algorithms

3. Customization

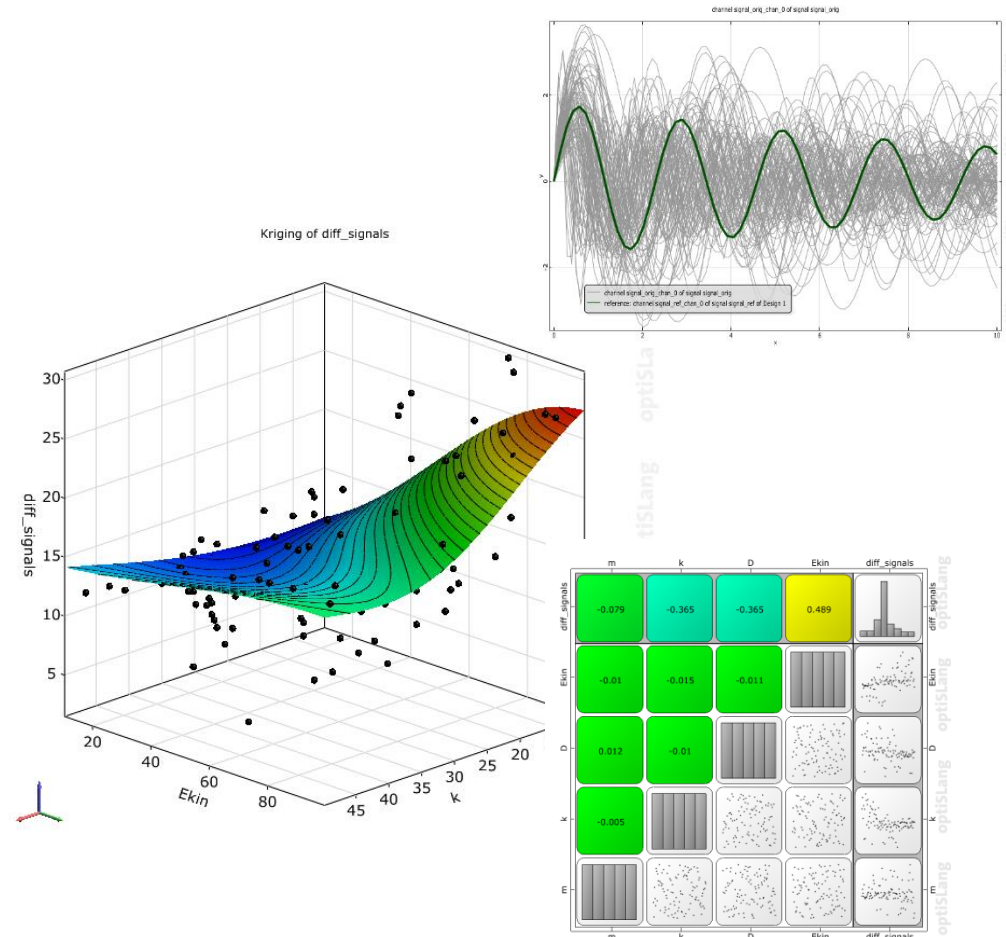
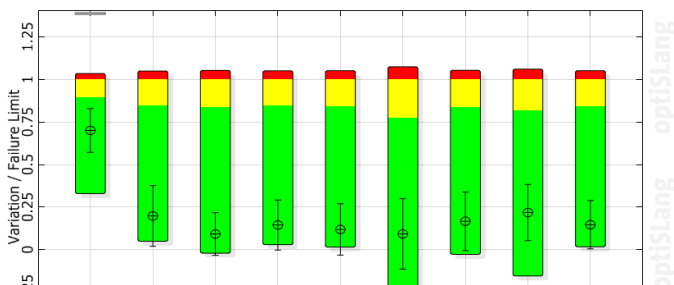


5. SPDM

4. Workflows

Postprocessing

- Predefined modes (installed)
 - *All plots available in all modes*
- Customized modes
 - *Standardized result comparison*
 - *Full access & flexibility*
- More
 - *Report generator*
 - *Automatic Update*
 - *Python console*



1. Post processing

2. optiSLang for ANSYS

6. Algorithms

3. Customization



5. SPDM

4. Workflows

Signal processing inside ANSYS Workbench

- ETK module

Access output parameters which are not "built-in" Workbench – e.g. arbitrary data in text or .rst files

- Nonscalar Data

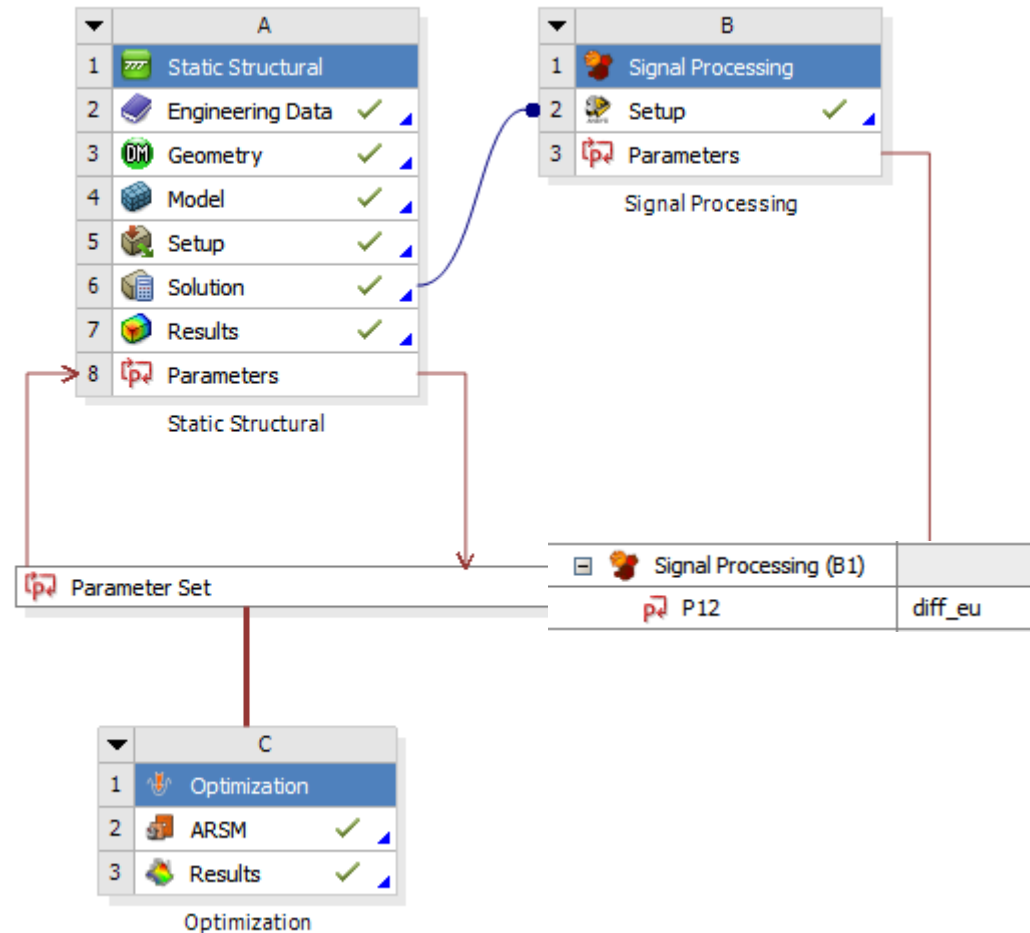
Use functionality of Extraction nodes inside ANSYS Workbench to work with data which is not supported via ANSYS, i.e. vectors, signals, matrices

- Register response values

Use powerful optiSLang calculator functionality to derive scalar values and register them in Workbench Parameter Set

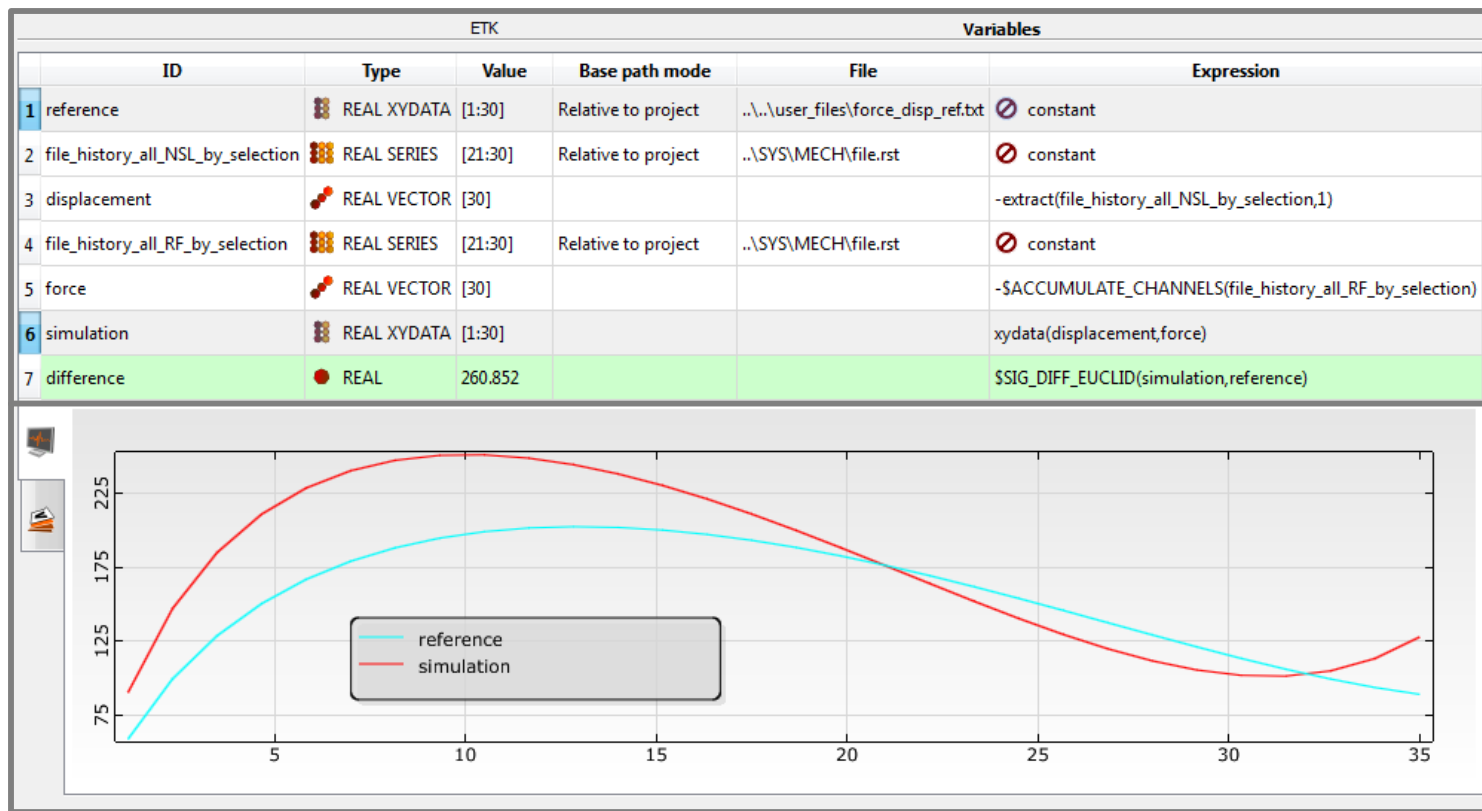
- Parameter identification

Use parameterized values to set up calibration task through optiSLang inside ANSYS Workbench

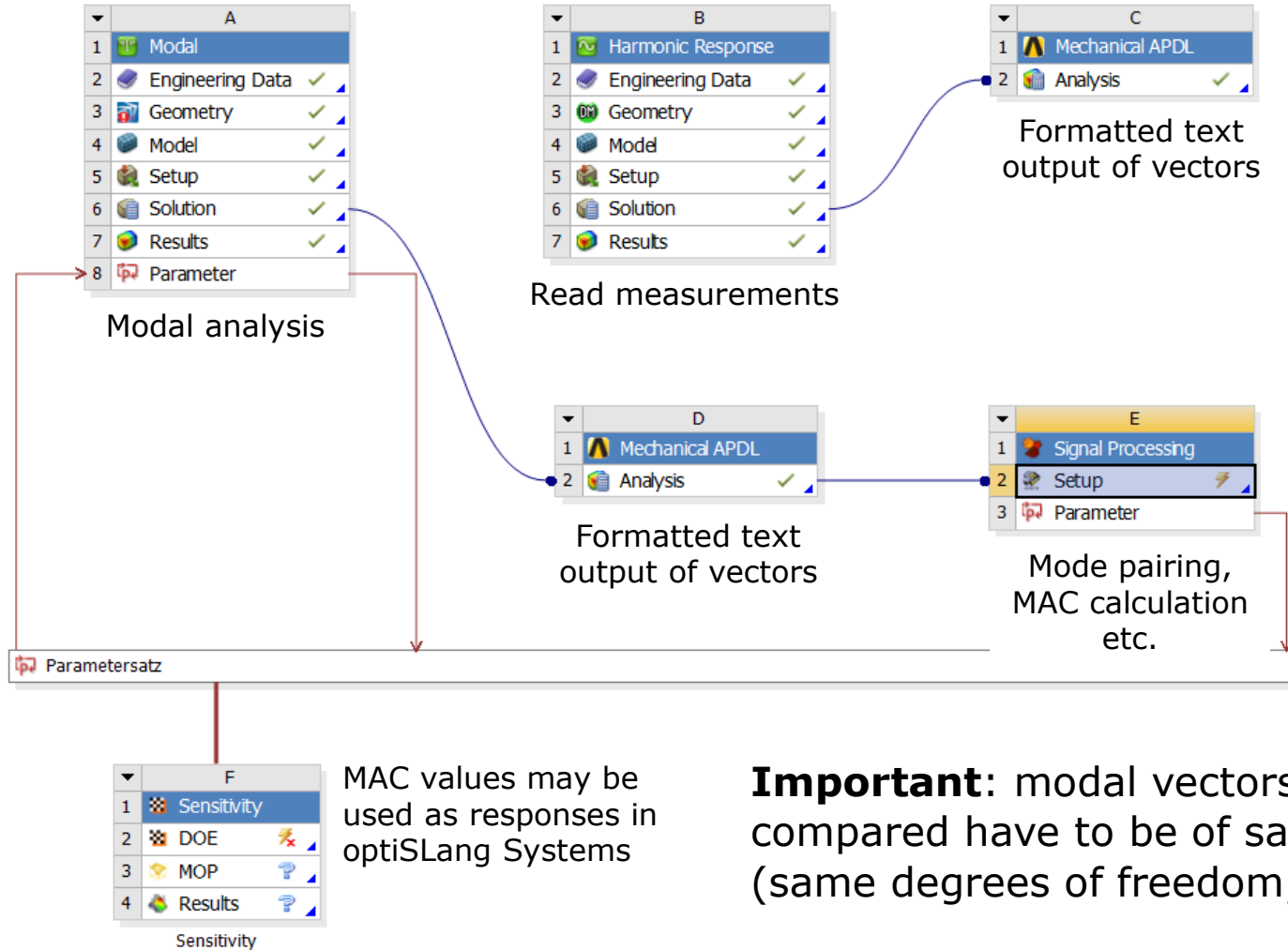


ETK inside ANSYS Workbench

- Signal visualization, extraction and analysis inside ANSYS Workbench



Usage within ANSYS Workbench



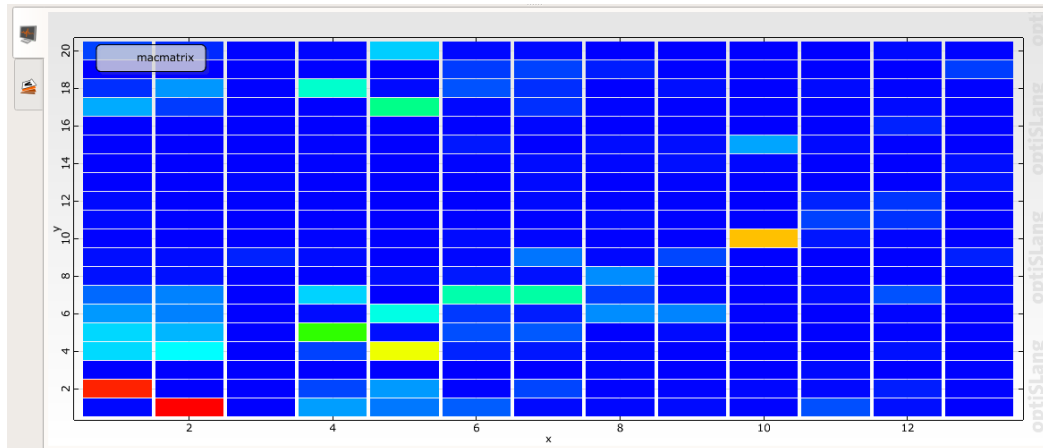
Important: modal vectors to be compared have to be of same format (same degrees of freedom, same order)

Signal Processing / ETK Setup

- Calculation of MAC matrix

ETK			Variables		
ID	Type	Value	Path	File	
41	REAL MATRIX	[20:13]			transpose(matrix(sim_modalbase,ref_modalbase))
42	REAL MATRIX	[20:13]			mac(sim_modalbase,ref_modalbase)

- Instant visualization of MAC matrix



- Mode pairing, extraction of frequencies and MAC values

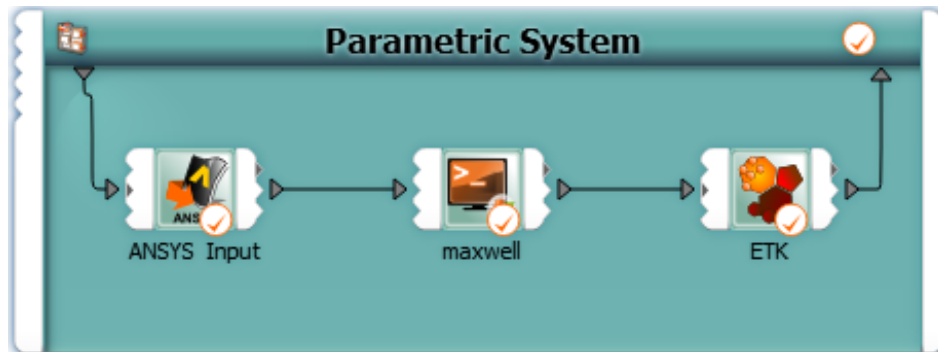
44	pairs_sim_ref	REAL VECTOR	[13]		modepairs(sim_modalbase,ref_modalbase,1e-5)
45	sim_frequencies_paired	REAL VECTOR	[13]		extract(transpose([sim_frequencies_all]),pairs_sim_ref)[,0]
46	mac_1	REAL	0.95341		macmatrix[pairs_sim_ref[0],0]

ANSOFT – Robust Design Optimization

Starting with any Ansoft project

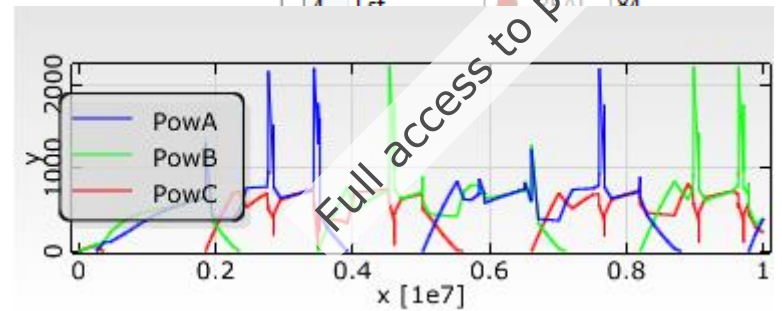


Generate system in optiSLang (<5min)

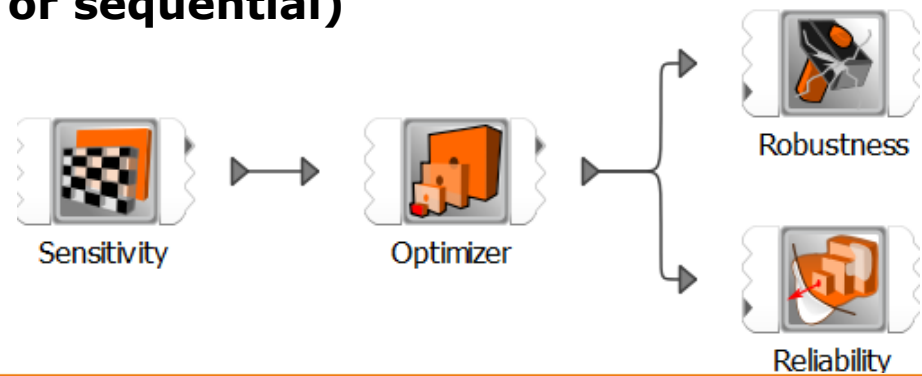
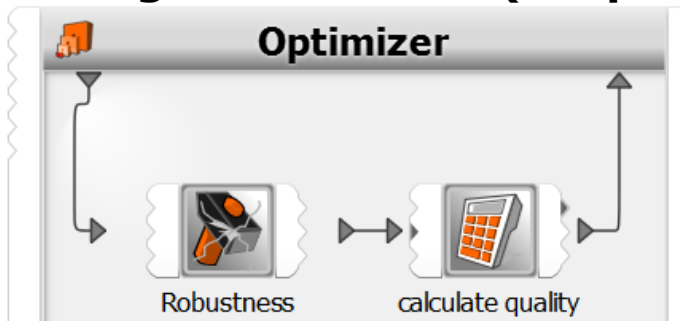


Detected parameter Advanced

	Name	Type	
1	fractions	REAL	6
2	Dr	REAL	77.34
3	Drin	REAL	12.6
4	Let	REAL	84

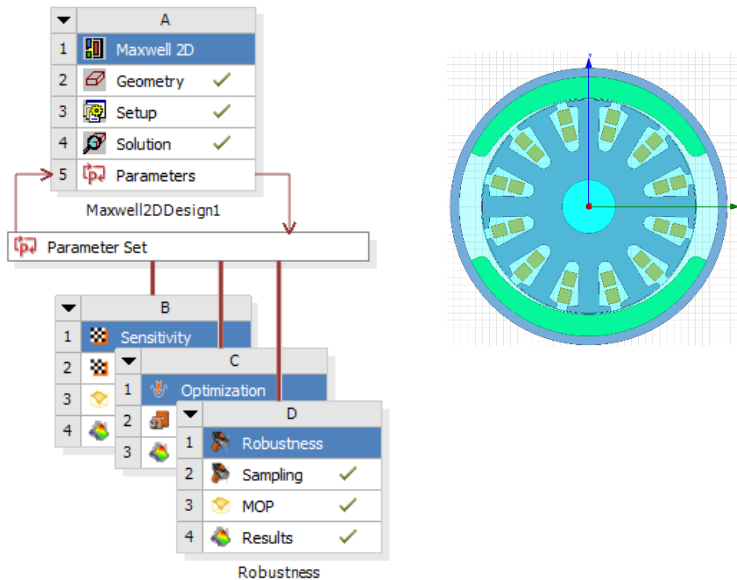


Configure RDO flow (coupled or sequential)

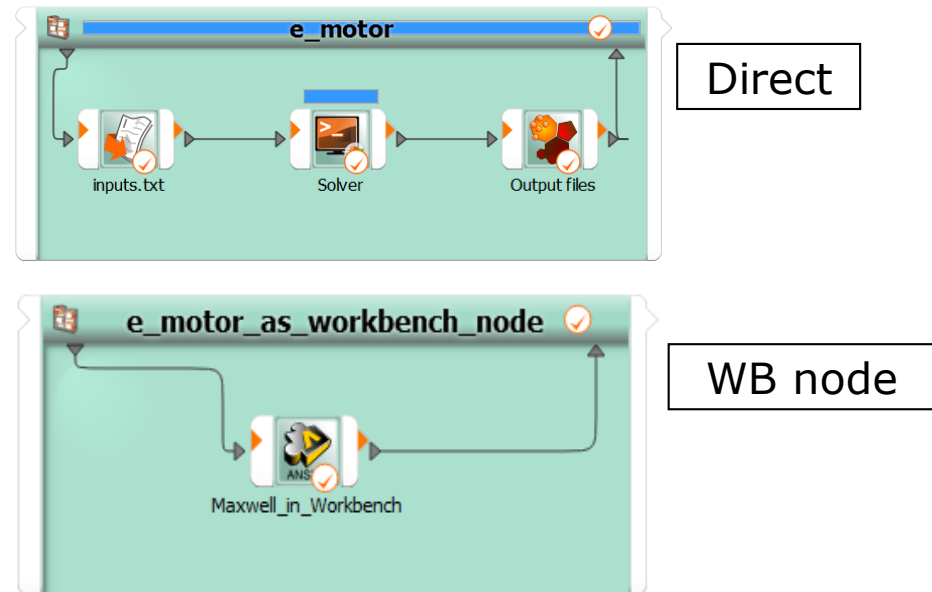


Integration Methods

optiSLang inside Workbench



Inside optiSLang



	Failed designs	Signals	DSO	Other HPC	„multiphysics“
Direct	█	█			
WB node	█	█		RSM	
Inside WB	█	█		RSM	(1)

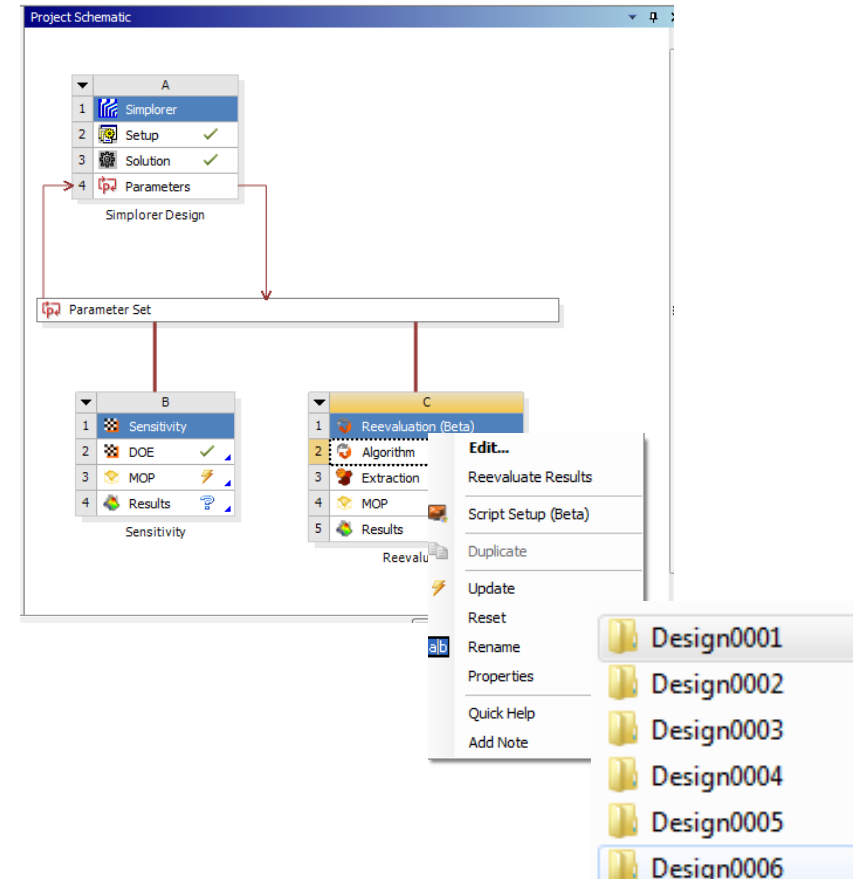
All modes supported

(1) Requires all physics in ANSYS WB

Re-evaluate inside ANSYS

Read designs from directories

- Similar to optiSLang systems
- Connect to Parameter Set
- Define
 - Directories to be considered
 - Data to be merged
 - Post processing commands
 - Additional Responses



1. Post processing

2. optiSLang for ANSYS

6. Algorithms

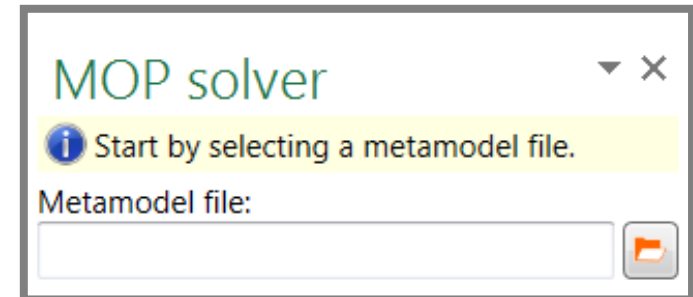
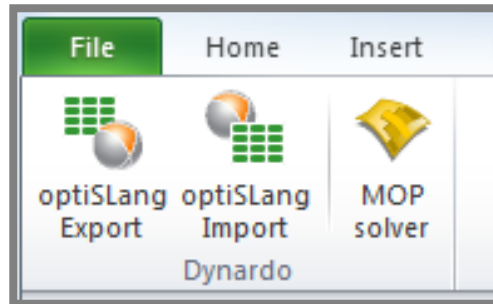
3. Customization



5. SPDM

4. Workflows

Run MOP solver in Excel







	A	B	C	D	E	F	G
1	optiSLang MOP solver version 4.2.3						
2	Meta model database was imported from:						
3	D:\PMOP_coupled_function.omdb						
4	Load time: 00:00:00.0580000						
5		X1	X2	X3	X4	X5	Full model
6	Y	17.40%	29.17%	66.51%	0.00%	0.29%	99.99%
7							
8		Parameters					Responses
9	Lower Bound	-3.1086	-3.1086	-3.1086	-3.1086	-3.1086	
10	Upper Bound	3.1086	3.1086	3.1086	3.1086	3.1086	
11	ID	X1	X2	X3	X4	X5	Y
12	0	1	1	1	1	1	
13							
14							

Custom Integrations

Since version 4.2.1 optiSLang provides plugin mechanisms for CAx tools

Customer can define his own integration in python script or C++:

- Input integration – SetParameters
- Solver – RunSolver
- Output integration – ExtractOutputs

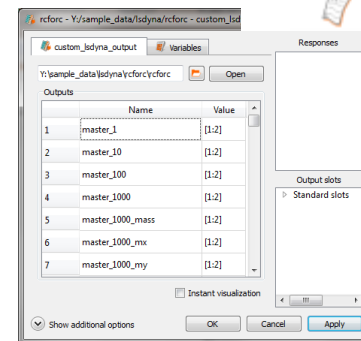
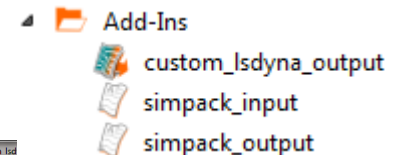
 custom_lsdyna_output_wrapper.png	27.02.2015 00:51	PNG-Bild
 custom_lsdyna_output_wrapper.py	27.02.2015 00:51	PY-Datei
 simpack_input_wrapper.py	27.02.2015 00:51	PY-Datei
 simpack_output_wrapper.py	27.02.2015 00:51	PY-Datei



```

248 #####
249 ## extract results
250 #####
251 def ExtractOutputContainer( args ) :
252     if len( args ) == 3 :
253         file_path = str(args[0])
254         working_dir = os.path.dirname( file_path )
255         reference_dir = str(args[1])
256         codec = str(args[2]) # 'utf-8' - do not seem to need it, but we will see...
257     else :
258         raise Exception( 'ExtractOutputContainer', 'Error: Wrong number of arguments. Expected
259
260     sys.path.append( working_dir )

```

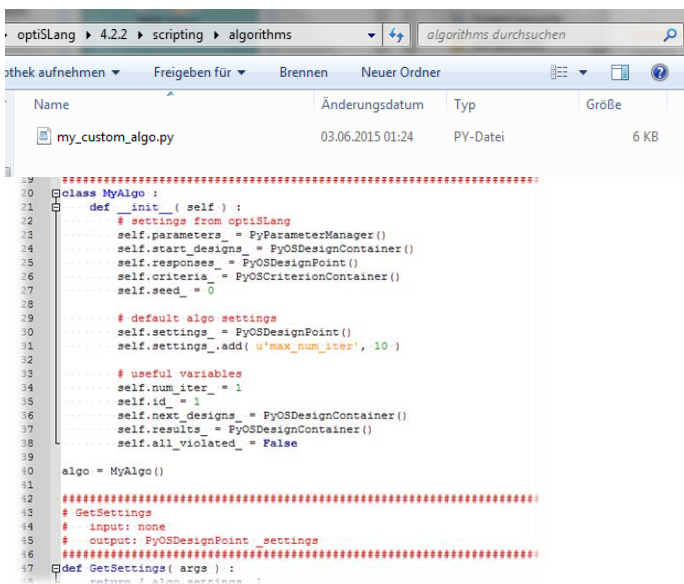


Custom Algorithms

optiSLang provides plugin mechanisms for CAx tools

Customer/3rd party can define his own algorithms in python script:

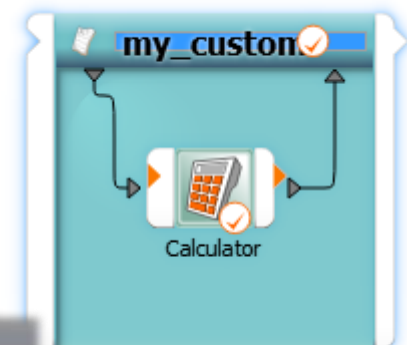
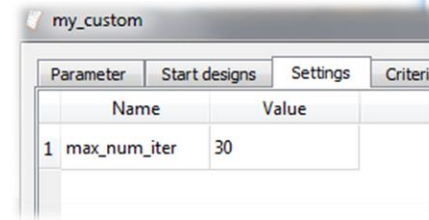
- Algorithms (DOE, Optimizer, ...) (*python in v4.2.2; C++ in v5*)
- Metamodels (*python in v4.2.3; C++ in v5*)



```

19 .....
20 class MyAlgo :
21     def __init__( self ) :
22         ..... # settings from optiSLang
23         self.parameters_ = PyParameterManager()
24         self.start_designs_ = PyOSDesignContainer()
25         self.responses_ = PyOSDesignPoint()
26         self.criteria_ = PyOSCriterionContainer()
27         self.seed_ = 0
28
29         ..... # default algo settings
30         self.settings_ = PyOSDesignPoint()
31         self.settings_.add( u'max_num_iter', 10 )
32
33         ..... # useful variables
34         self.num_iter_ = 1
35         self.id_ = 1
36         self.next_designs_ = PyOSDesignContainer()
37         self.results_ = PyOSDesignContainer()
38         self.all_violated_ = False
39
40 algo = MyAlgo()
41
42 .....
43 # GetSettings
44 # input: none
45 # output: PyOSDesignPoint_settings
46 .....
47 def GetSettings( args ) :
48     return f.algo.settings_

```

Parameter	Start designs	Settings	Criteria				
		<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>1 max_num_iter</td> <td>30</td> </tr> </tbody> </table>	Name	Value	1 max_num_iter	30	
Name	Value						
1 max_num_iter	30						

1. Post processing

2. optiSLang for ANSYS

6. Algorithms

3. Customization

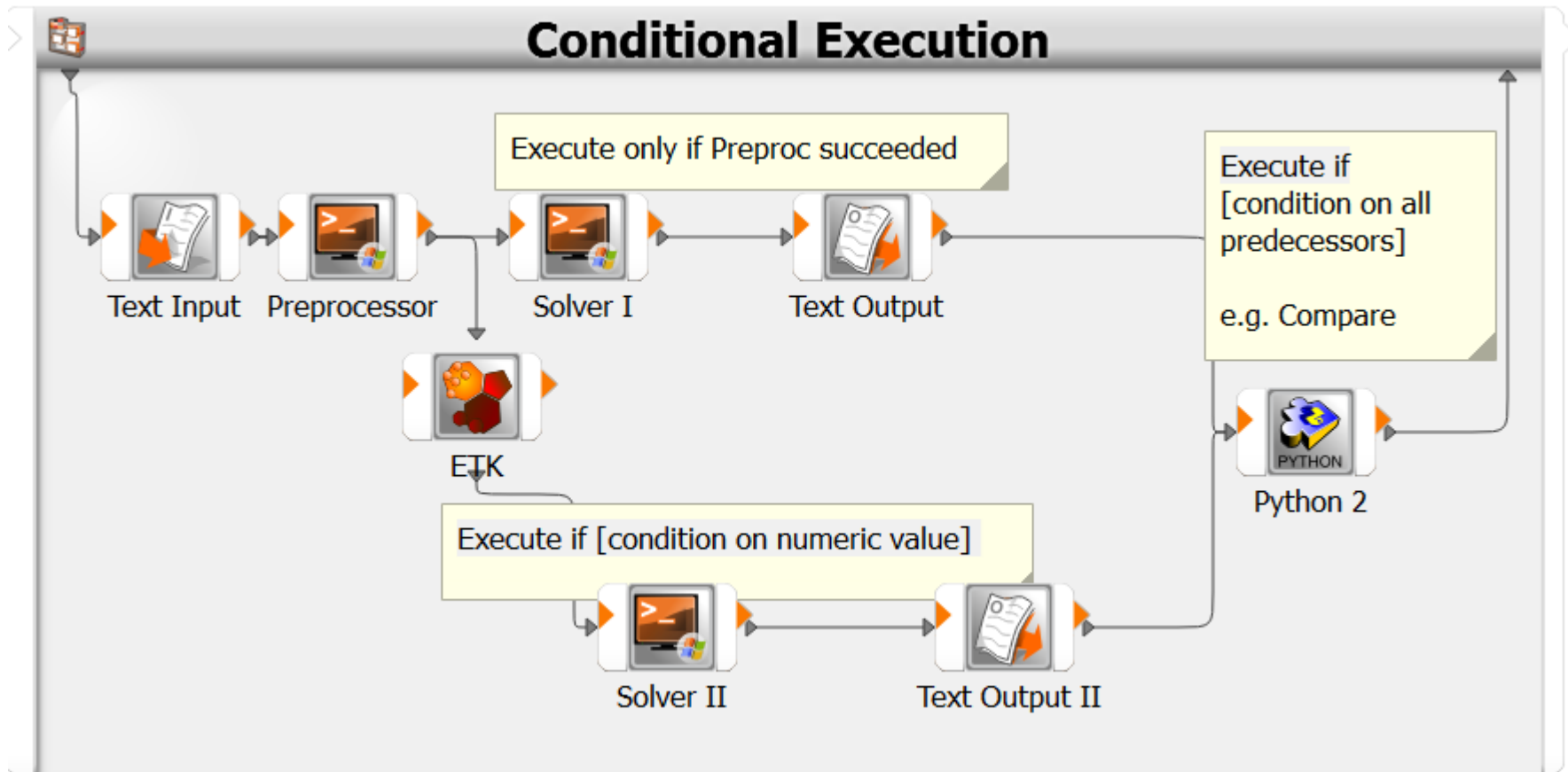


5. SPDM

4. Workflows

Conditional execution

- Do (not) evaluate if ...



1. Post processing

2. optiSLang for ANSYS



3. Customization

6. Algorithms

5. SPDM

4. Workflows

SPDM

optiSLang SPM



+

Database (PDM)



Part of optiSLang:

- Simulation Process Management
 - Automate and standardize CAX – flows
- Connection to Process Data Management
 - Receive data from PDM and send Data to PDM

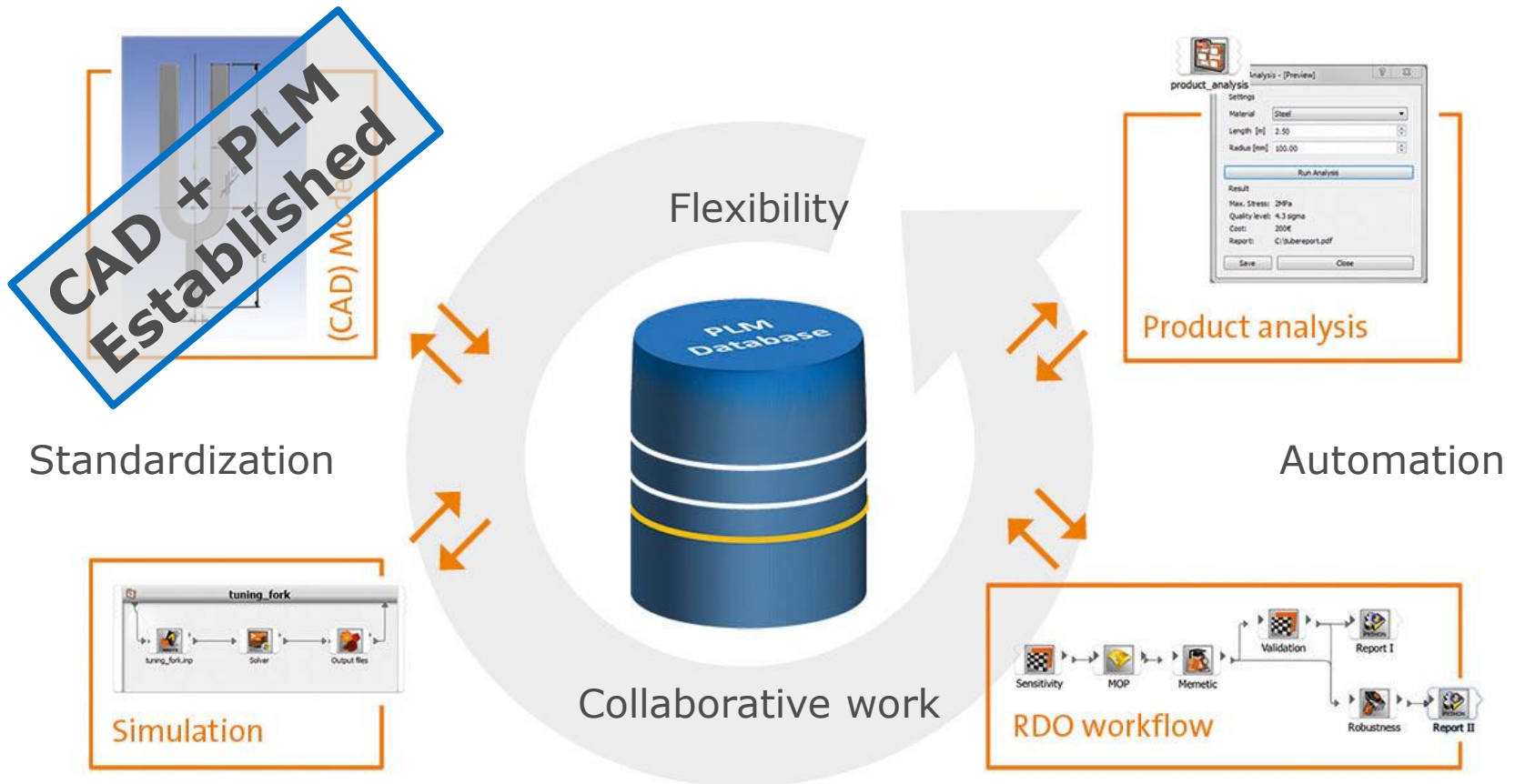
Consider 4 user groups

- Designer
 - Knowledge in CAD, construction rules, PLM, ...
- Simulation Engineer
 - Knowledge about model, meshing, ...
- System Engineer
 - Knowledge about coupling, "Multiphysics", ...
- Product Engineer
 - Knowledge about constraints, requirements, ...



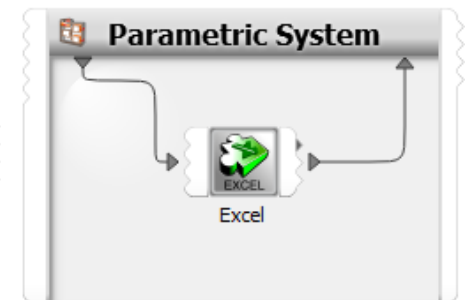
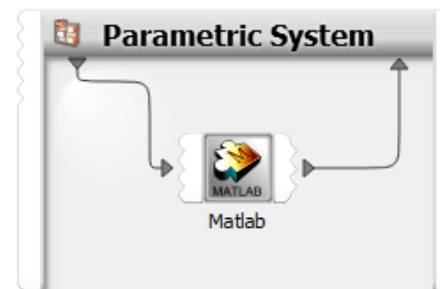
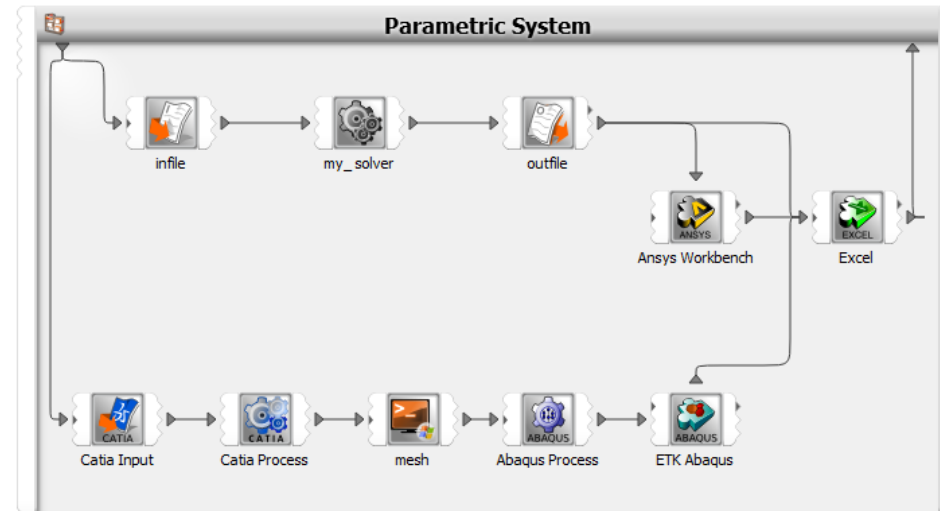
**Collaborative
Work required**

PLM/SDM system is central component
optiSLang for SPM, standardization and automation



Build CAE flow - Use optiSLang process integration

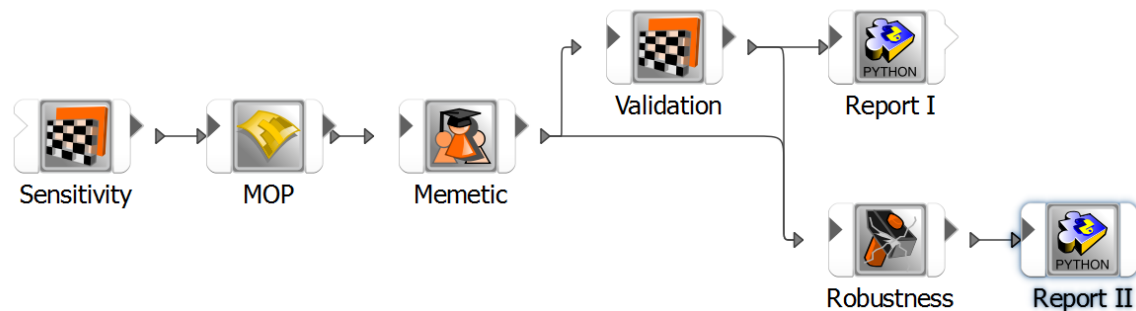
- Graphical setup
 - “Like a flowchart”
 - Wizard supported
- Combine
 - CAD Tools: CATIA, Creo, ...
 - Solver: ANSYS, Abaqus, LSDyna ...
 - Other: Excel, Matlab, ...
 - Internal solver
 - ...



optiSLang – general purpose software for parametric studies

- DOE & performance maps
- Sensitivity studies
- Optimization algorithms
- Robustness/Reliability analysis
- ➔ Robust Design Optimization

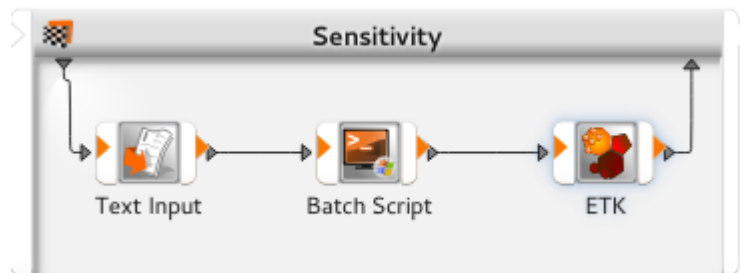
Easily couple algorithms and parametric model in a workflow



Simulation + Workflow engineer

Task: Create a project template

- Has access to all properties of an (existing) optiSLang project
- Selects which properties can be changed



Name	Placeholder
<ul style="list-style-type: none"> <ul style="list-style-type: none"> Parametric import file 	parametric_file
<ul style="list-style-type: none"> <ul style="list-style-type: none"> <ul style="list-style-type: none"> Environment Maximum degree of parallelism Maximum runtime Script path 	max_parallelism
<ul style="list-style-type: none"> <ul style="list-style-type: none"> Text Input 	input_file
<ul style="list-style-type: none"> <ul style="list-style-type: none"> <ul style="list-style-type: none"> Maximum degree of parallelism Path of displacement Path of displacement_ref Expression of difference 	max_parallelism output_file reference_file difference_exp

Simulation + Workflow engineer

Task: Define project parameter and rights to see & change

- Defines “project parameter”
 - Use Expressions and free variables to avoid redundancy
 - See e.g. variable “model”
- Defines what others see and can change
 - See column “minimum user level”

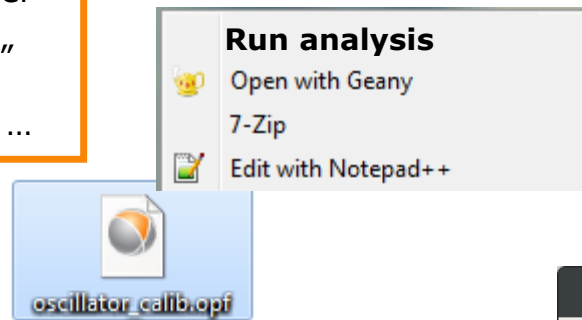
	Id	Minimum user level	Data type	Value
1	reference_files_dir	Computation Engineer	Path	\${project_dir}/files
2	model	Computation Engineer	String	oscillator
3	parametric_filename	Flow Engineer	Path	\${model}_parametric.csv
4	parametric_file	Flow Engineer	Path	\${reference_files_dir}/\${parametric_filename}
5	input_filename	Flow Engineer	Path	\${model}.s
6	input_file	Flow Engineer	Path	\${reference_files_dir}/\${input_filename}
7	max_parallelism	Computation Engineer	Integer	100

Product engineer

Task: Run the analysis (use the project template)

Standardized process e.g.

- Start Windows Explorer
 - Choose "Run analysis"
- Or web-interface, Excel ...



- Only needs to specify project parameters
- No need to care about optiSLang, CAD Model, ...

➔ Results can be automatically stored in PLM/PDM

Placeholder mapping		
	Id	
1	model	oscillator
2	max_parallelism	100

1. Post processing

2. optiSLang for ANSYS

6. Algorithms

3. Customization

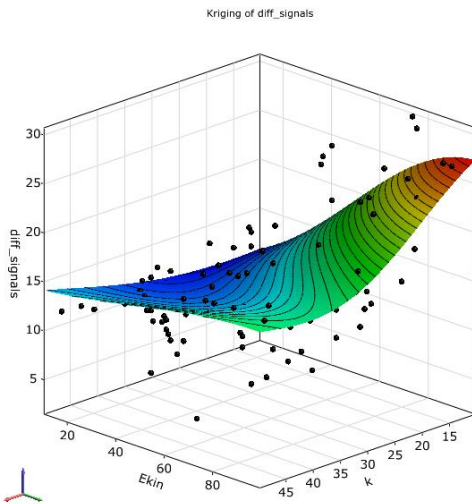


5. SPDM

4. Workflows

MOP – the next generation

- PMOP is now first choice
 - *Parallel search*
 - *Kriging, external models*
- Reduce settings on 1st page
- Use new postprocessing
 - *E.g. Plot cross validation values*



optiSLang optiSLang optiSLang optiSLang optiSLang

Metamodel of Optimal Prognosis (MOP)

Database file: OSL4Projects/cf.opd/Reevaluate/Reevalu

Automatic
 MOP
 PMOP

Settings: MOP advanced | PMOP advanced | Messa

Model complexity: complex

Variable reduction: filter unimportant

Postprocessing: do not show

Summary

Calculating models

- Linear regression
- Moving least squares
- Isotropic kriging
- Anisotropic kriging

All parameters are selectable

Parameters

- area_01
- area_02
- area_03
- area_04
- area_05
- area_06
- area_07
- area_08
- area_09
- area_10

Responses

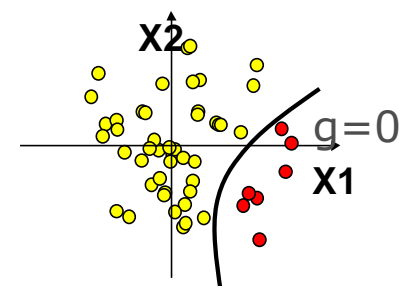
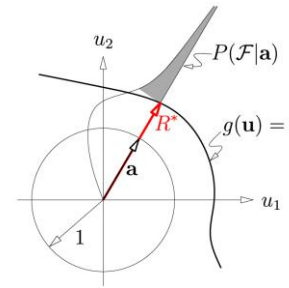
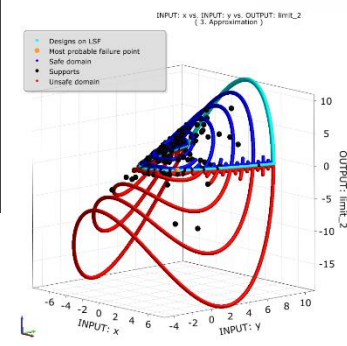
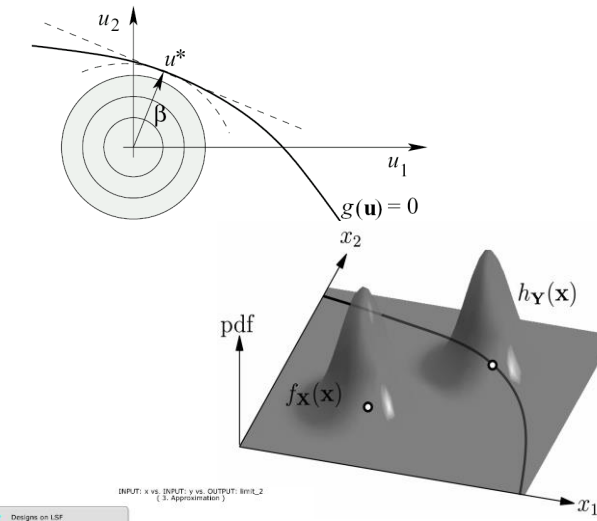
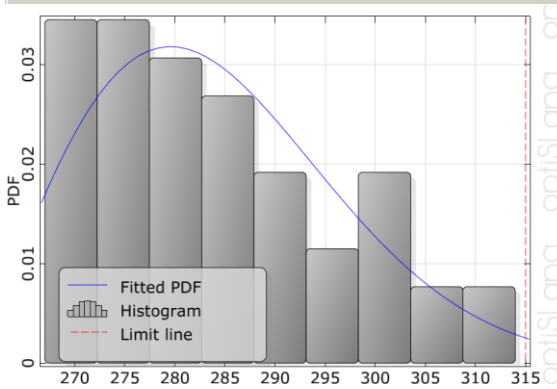
- Mass
- Stress_element

Show additional options

Approximation type	Smoothing
Max. responses in parallel	1
Use incomplete designs	<input type="checkbox"/> False
Adapt bounds	
CoP tolerance	
Transformation	
Models	
Polynomials	
Use	<input checked="" type="checkbox"/> True
Order	2
Coefficient factor	2.00
Moving Least Squares	
Use	<input checked="" type="checkbox"/> True
Order	2
Coefficient factor	8.00
Kriging	
Use	<input checked="" type="checkbox"/> True
Anisotropic	<input checked="" type="checkbox"/> True
Coefficient factor	8.00
External	
ASCMO	<input checked="" type="checkbox"/> True
my_custom	<input checked="" type="checkbox"/> True
my_other_custom	<input checked="" type="checkbox"/> True
Filter	
Postprocessing	
Algorithm messages	
Crossvalidation values	

Robustness Wizard

- Introduce reliability algorithms
- Best guidance through decision tree



dynardo

dynamic software & engineering

optiSLang 5

David Schneider

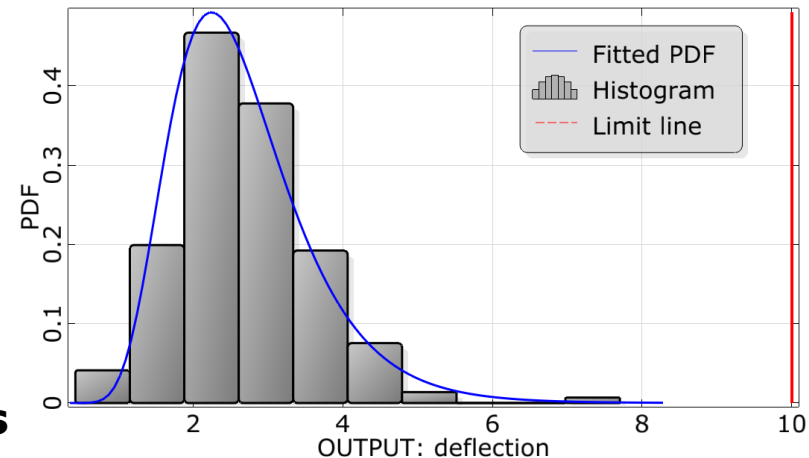
Product manager

Robustness/Reliability Analysis



Variance based Robustness Analysis

- Sufficient estimates of mean and variance with 50 to 100 samples
- Distribution fit and extrapolation of **small event probabilities** may be very inaccurate
- More precise **reliability methods** should be applied to verify small probabilities

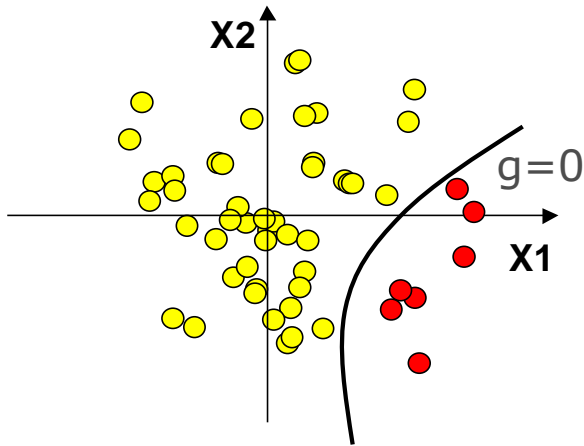


Fitted PDF: Normal	
Mean:	2.67
Sigma:	0.9357
Limit x = 10	
P_rel:	1
1 - P_rel:	0
P_fit:	1
1 - P_fit:	2.33147e-015
Sigma-Level:	7.83397

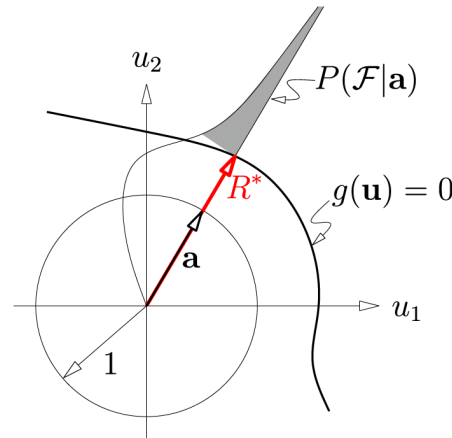
Statistic data	
Min:	0.4254
Max:	7.704
Mean:	2.67
Sigma:	0.9357
CV:	0.3505
Skewness:	1.017
Kurtosis:	6.465
Fitted PDF: Log-Normal	
Mean:	2.67
Sigma:	0.9357
Limit x = 10	
P_rel:	1
1 - P_rel:	0
P_fit:	0.999974
1 - P_fit:	2.56303e-005
Sigma-Level:	7.83397

optiSLang Methods for Reliability Analysis

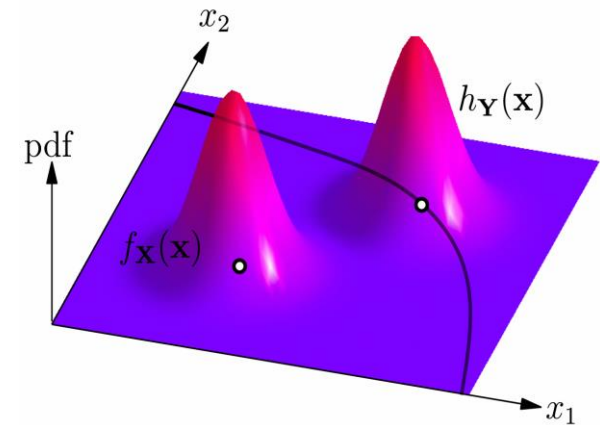
Monte Carlo Sim.



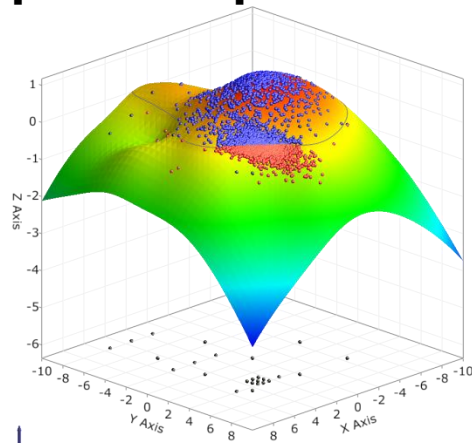
Directional Sampling



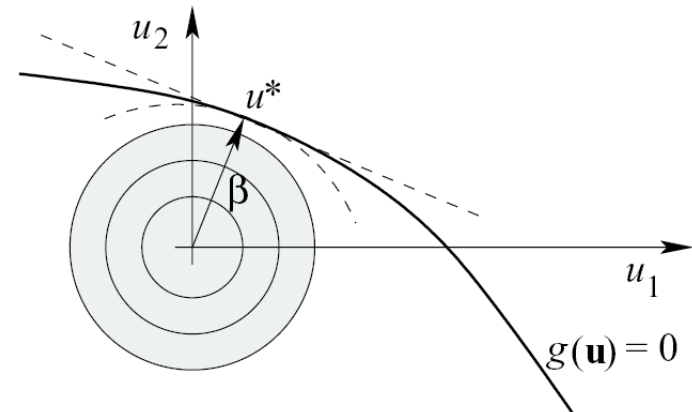
Adaptive Sampling



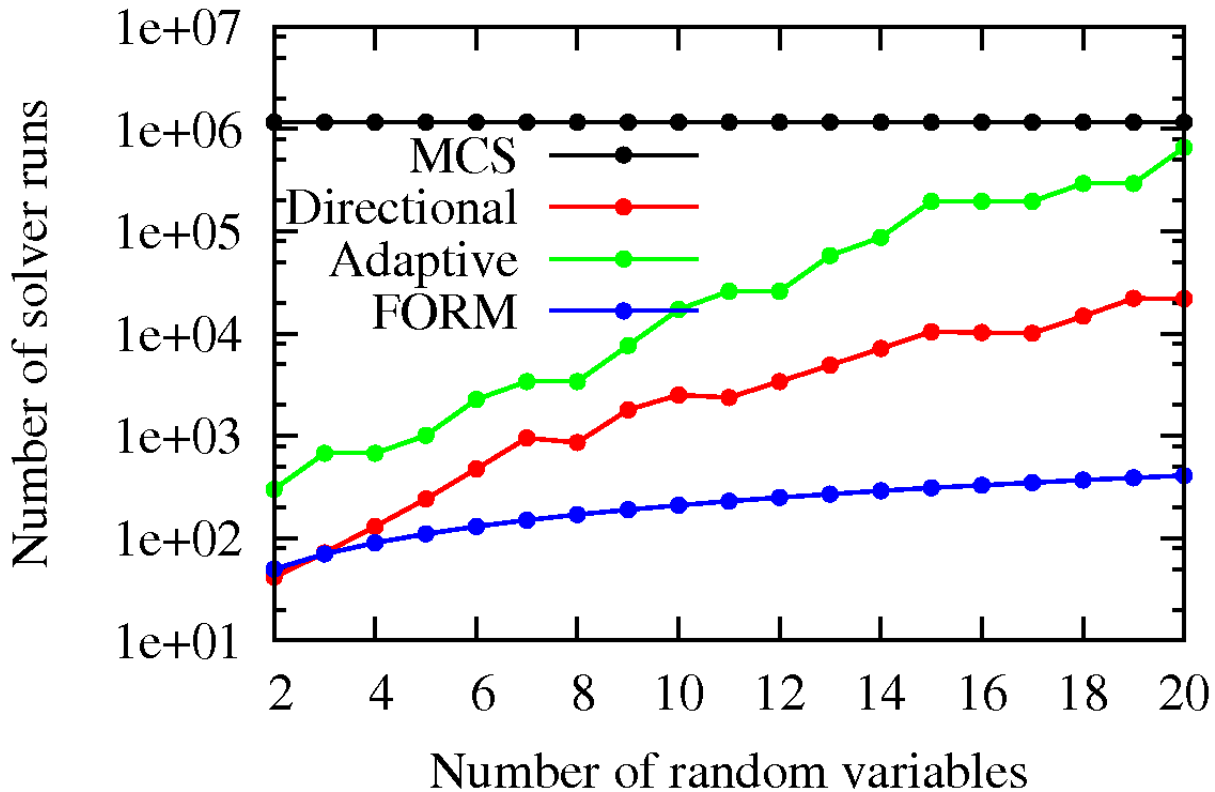
Adaptive Response Surfaces



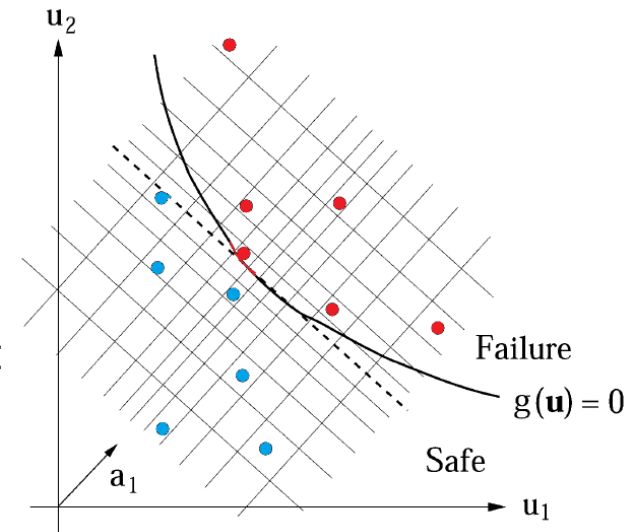
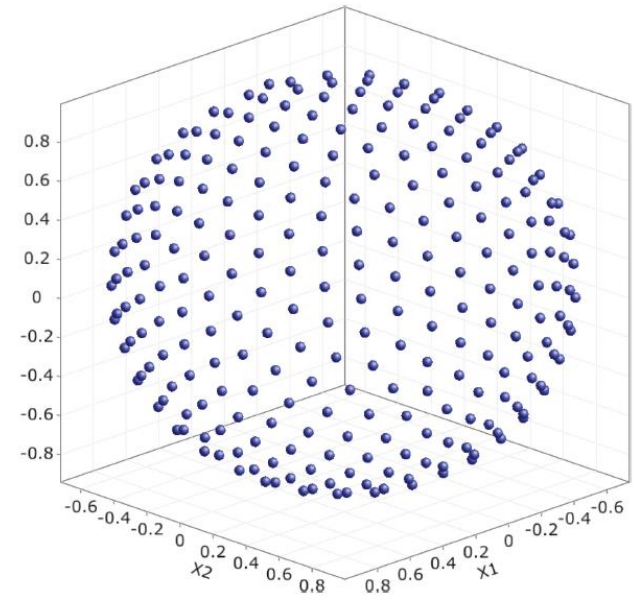
First Order Reliability Method



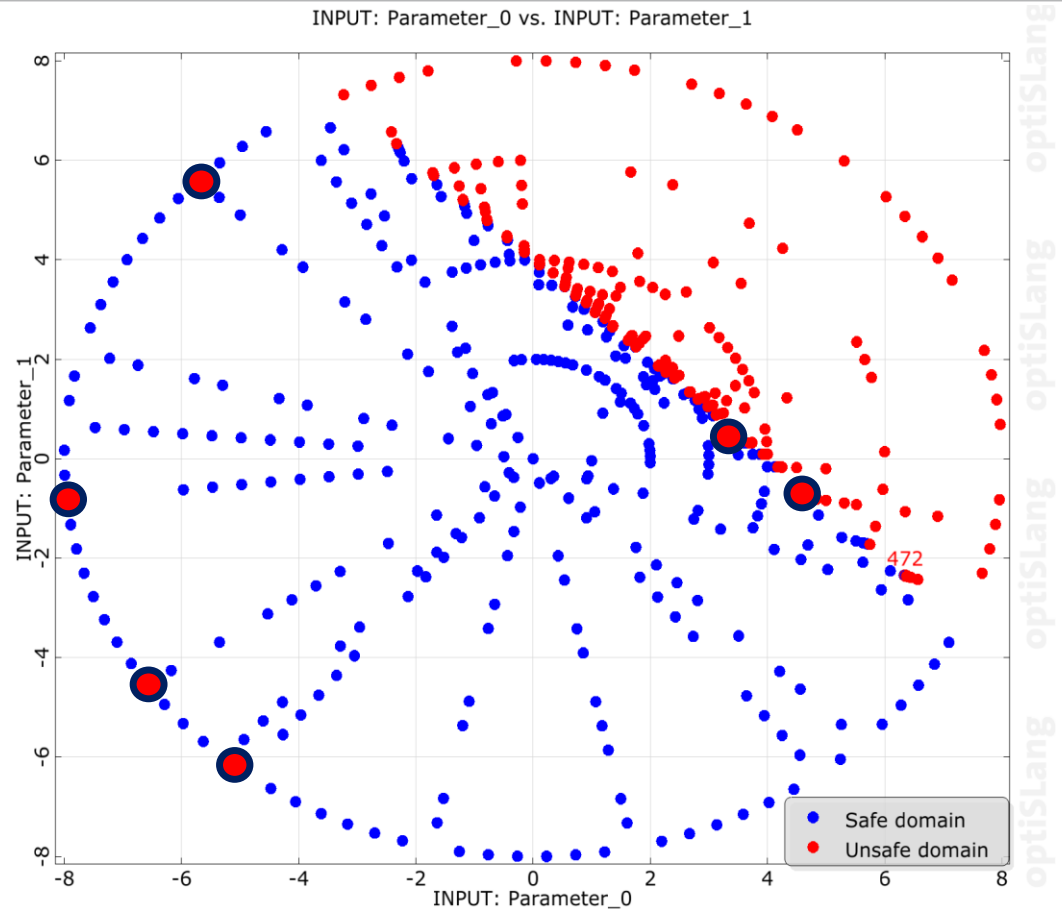
Performance



- Approximation methods are much more efficient
- Improvement of sampling scheme in Directional and Importance sampling



Failed Designs



- Failed designs are interpreted as random failure (e.g. license problems) and are not considered in the statistical evaluation
- Line search of Directional sampling and gradient estimation of FORM are extended for failed designs

Robustness/Reliability Wizzard

Uncertainty knowledge: Not set

Failed designs: Not set

Solver noise: Not set

Desired sigma level: 2 σ 3 σ 4,5 σ 6 σ

► Show additional settings

Robustness / Reliability method

Variance based

Robustness sampling

Probability based

Adaptive Response Surface Method (ARSM-DS)

Adaptive Sampling (AS)

Directional Sampling (DS)

First Order Reliability Method (FORM)

Importance Sampling using Design Point (ISPUD)

Monte Carlo Simulation (MCS)

- If **no limit state** is defined or
- If the uncertainty knowledge is **not qualified**
- Robustness sampling is recommended
- However, an extrapolation for more than 3 sigma is difficult

Robustness/Reliability Wizzard

Uncertainty knowledge: Qualified

Failed designs: Seldom

Solver noise: Some

Desired sigma level: 2σ 3σ $4,5\sigma$ 6σ

► Show additional settings

Robustness / Reliability method

Variance based

Robustness sampling

Probability based

Adaptive Response Surface Method (ARSM-DS)

Adaptive Sampling (AS)

Directional Sampling (DS)

First Order Reliability Method (FORM)

Importance Sampling using Design Point (ISPUD)

Monte Carlo Simulation (MCS)

- **Seldom failed designs** and **some solver noise** can be handled by all reliability methods
- For up to 15 variables, the ARSM-DS is the best compromise between accuracy and efficiency

Robustness/Reliability Wizzard

Uncertainty knowledge: Qualified

Failed designs: Frequently

Solver noise: Strong

Desired sigma level: 2σ 3σ $4,5\sigma$ 6σ

► Show additional settings

Robustness / Reliability method

Variance based

Robustness sampling

Probability based

Adaptive Response Surface Method (ARSM-DS)

Adaptive Sampling (AS)

Directional Sampling (DS)

First Order Reliability Method (FORM)

Importance Sampling using Design Point (ISPUD)

Monte Carlo Simulation (MCS)

- In case of **frequently failed designs** and **strong solver noise** ARSM-DS, FORM and directional sampling may be not robust enough
- Adaptive sampling is the best compromise between accuracy and efficiency up to 15 variables