

Technology update

optiSLang v4.1

Robust Design Optimization

Johannes Will
Dynardo GmbH

optiSLang v4

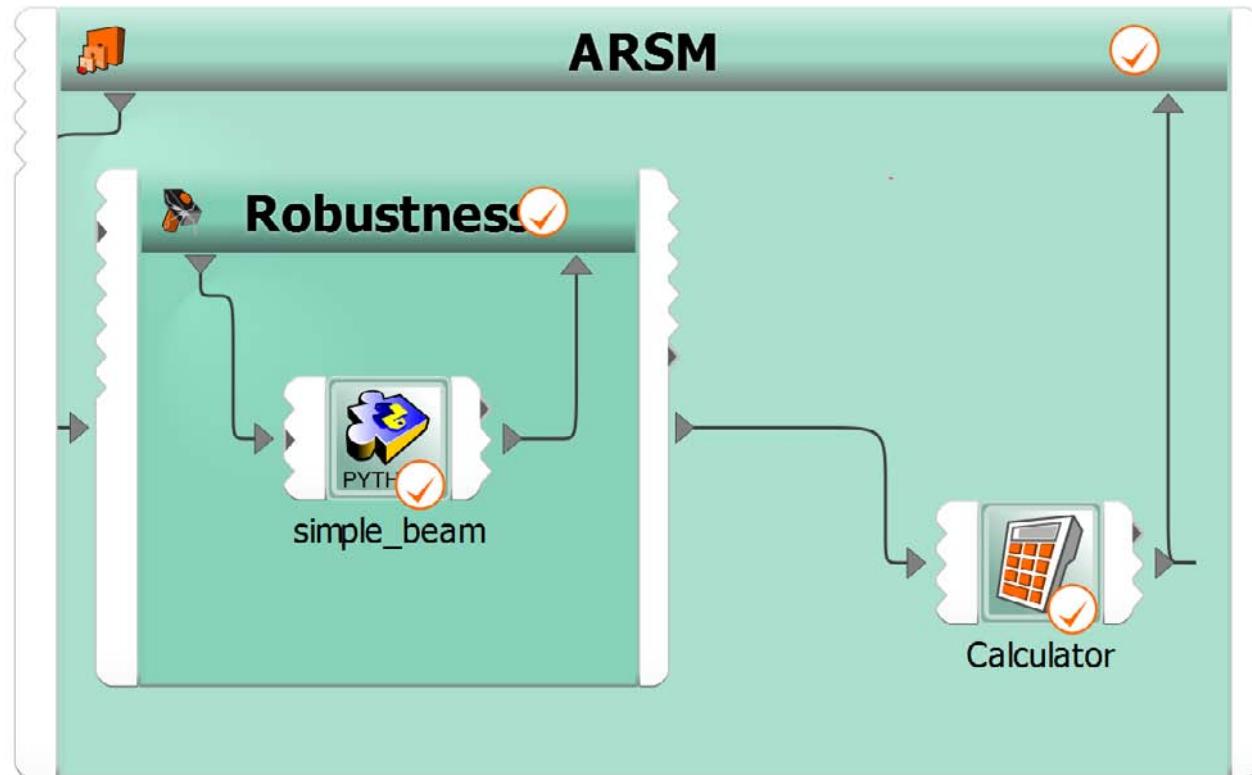
- Comprehensive systems “easy and safe to use ”
 - Easy modeling of the process chain
- “easy and safe to use methodology”
 - Minimal effort to set up analysis
 - Algorithms with proven default settings
- Clear and intuitive graphical representation
 - To see and understand the workflow



Simultaneous RDO Nested Algorithms (Variance based RDO)

Build loop in loop systems

e.g. variance based Robust Design Optimization

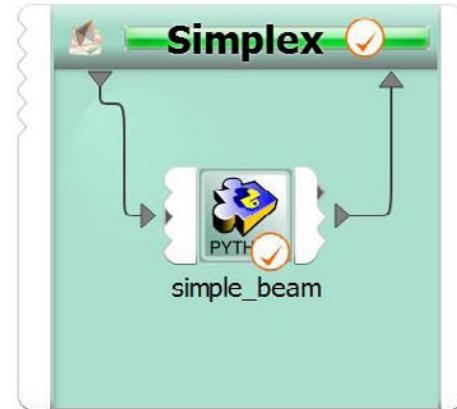




New & updated Algorithms

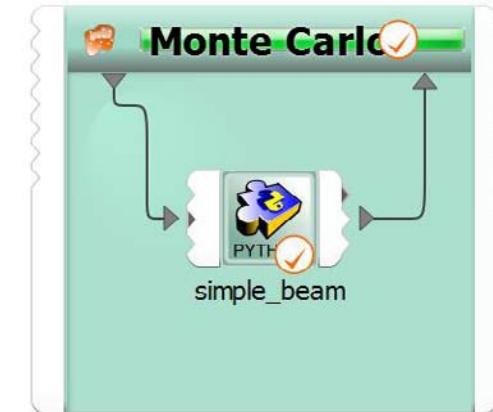
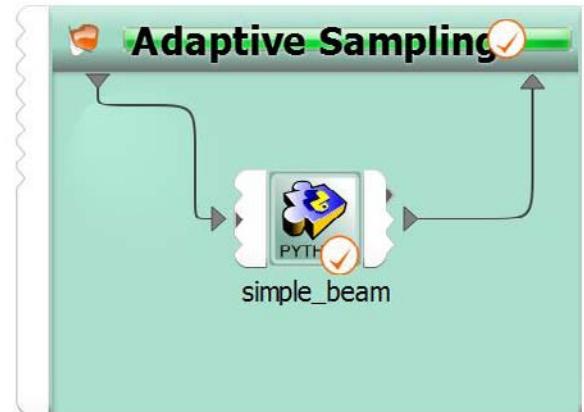
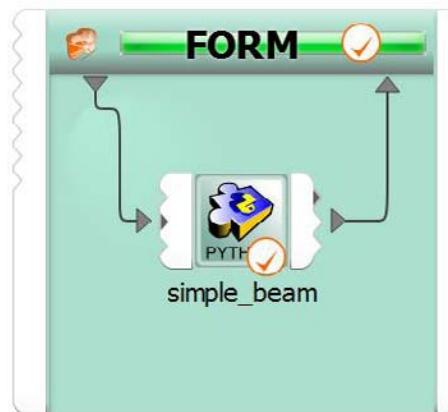
Optimization

- Simplex



Reliability

- Monte Carlo
- FORM
- Adaptive Sampling

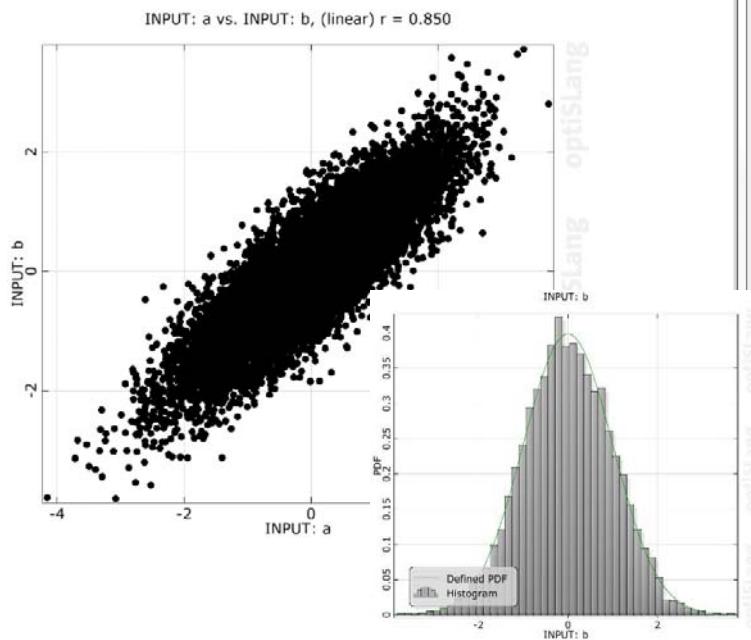




Input Correlations now available at v4

V4 Parameter manager supports correlations

- stochastic and mixed parameters
- for all parameters or a selection
- check validity “on click”



ten_bar_truss - Parametric System

| Parameter | Criteria | Other | Result designs | | | | |
|------------|----------------|-----------------|-------------------------------------|-----|--------|------|-----------|
| Name | Parameter type | Reference value | Constant | PDF | Type | Mean | Std. Dev. |
| 1 area_01 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 2 area_02 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 3 area_03 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 4 area_04 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 5 area_05 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 6 area_06 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 7 area_07 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 8 area_08 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 9 area_09 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |
| 10 area_10 | Stochastic | 10 | <input checked="" type="checkbox"/> | | NORMAL | 0 | 1 |

Define parameter correlations

| area_02 | area_03 | area_04 | area_07 | area_08 |
|---------|---------|---------|---------|---------|
| area_08 | 0.1 | | | 1 |
| area_07 | | 0.01 | | 1 |
| area_04 | | | 1 | |
| area_03 | | | | 0.01 |
| area_02 | 1 | | | 0.1 |

Buttons: Restore Defaults, OK, Cancel, Apply

Cluster Identification at Box (Parallel Coordinate) Plots

- Cluster analysis

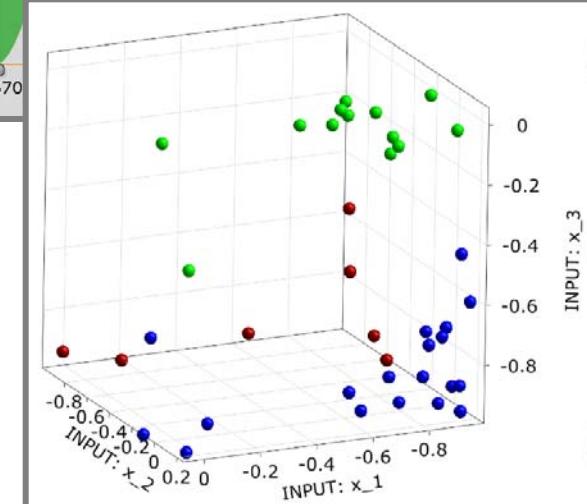
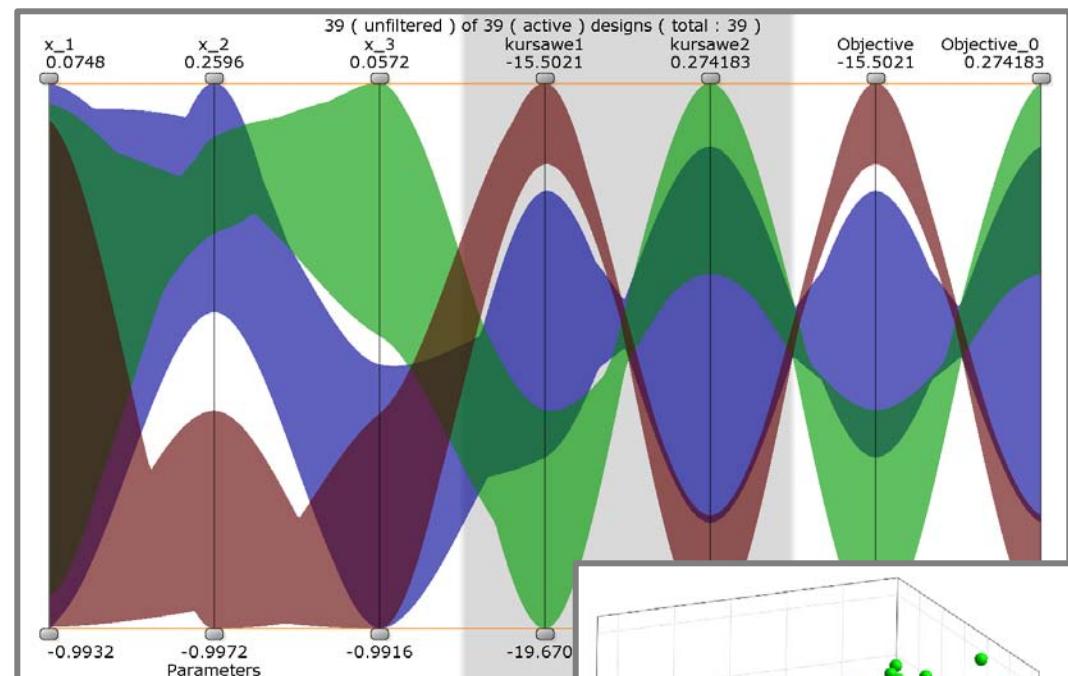
Use cluster algorithm to automatically find concentrations in input, response or criteria space

- Parallel coordinates plot (PCP)

Use PCP to apply cluster analysis on selected data and get visual feedback

- Transfer information

Use the same coloration in anthill plots via "Design colors"-Button

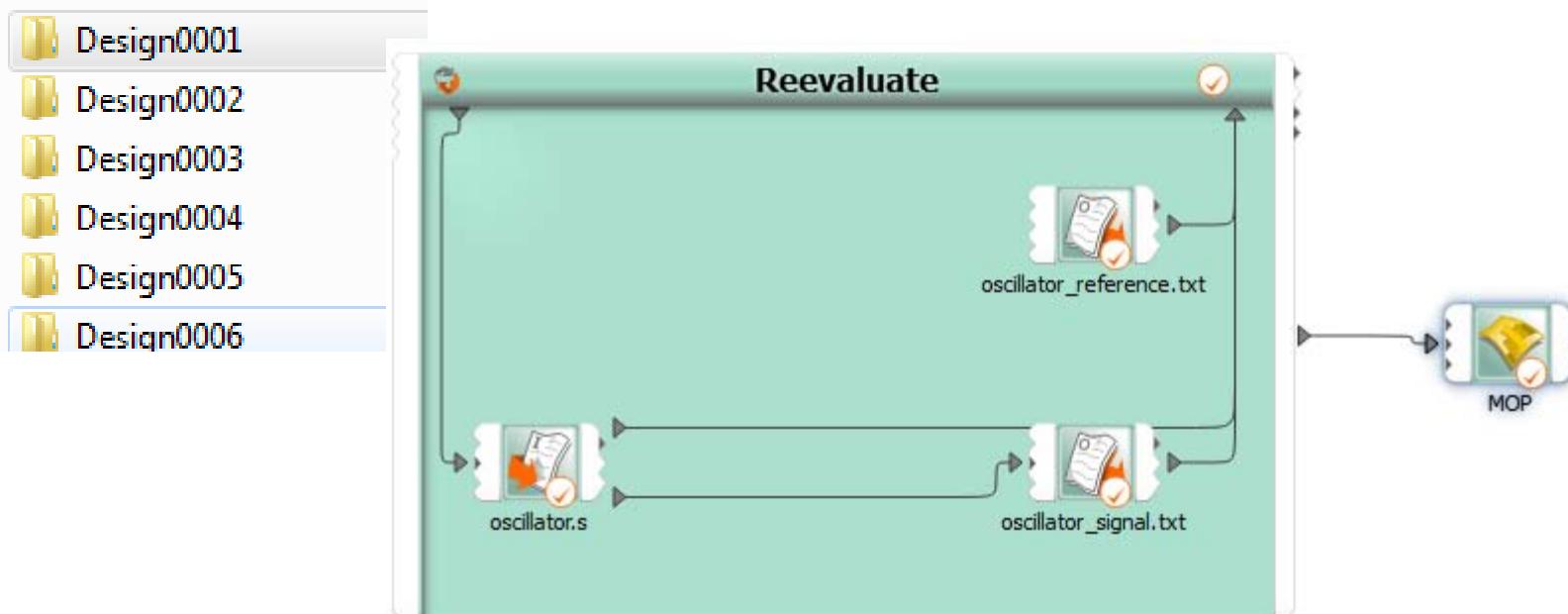




Re-evaluate in v4

Read designs from directories

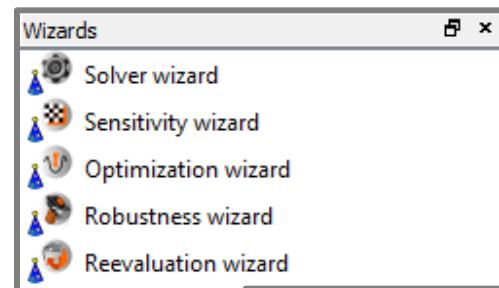
- Inputs and Outputs
- Add Postprocessing
- Merge with previous analysis results



Find the Modules

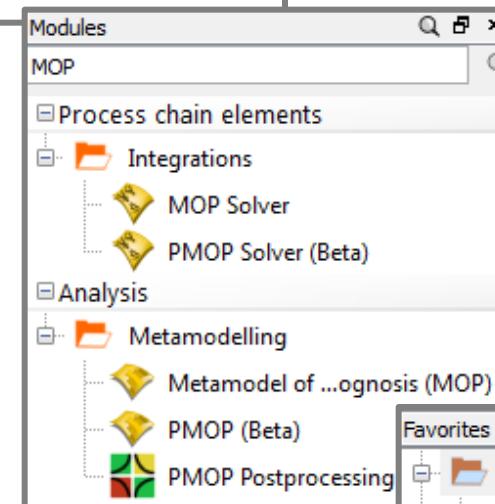
- Wizards

Separated to extra box



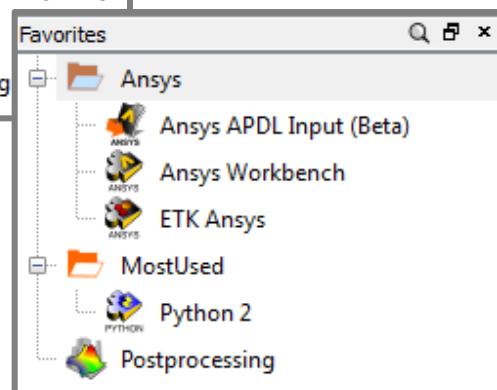
- Search functionality

Guarantees quick access to specific solution and good overview of all modules

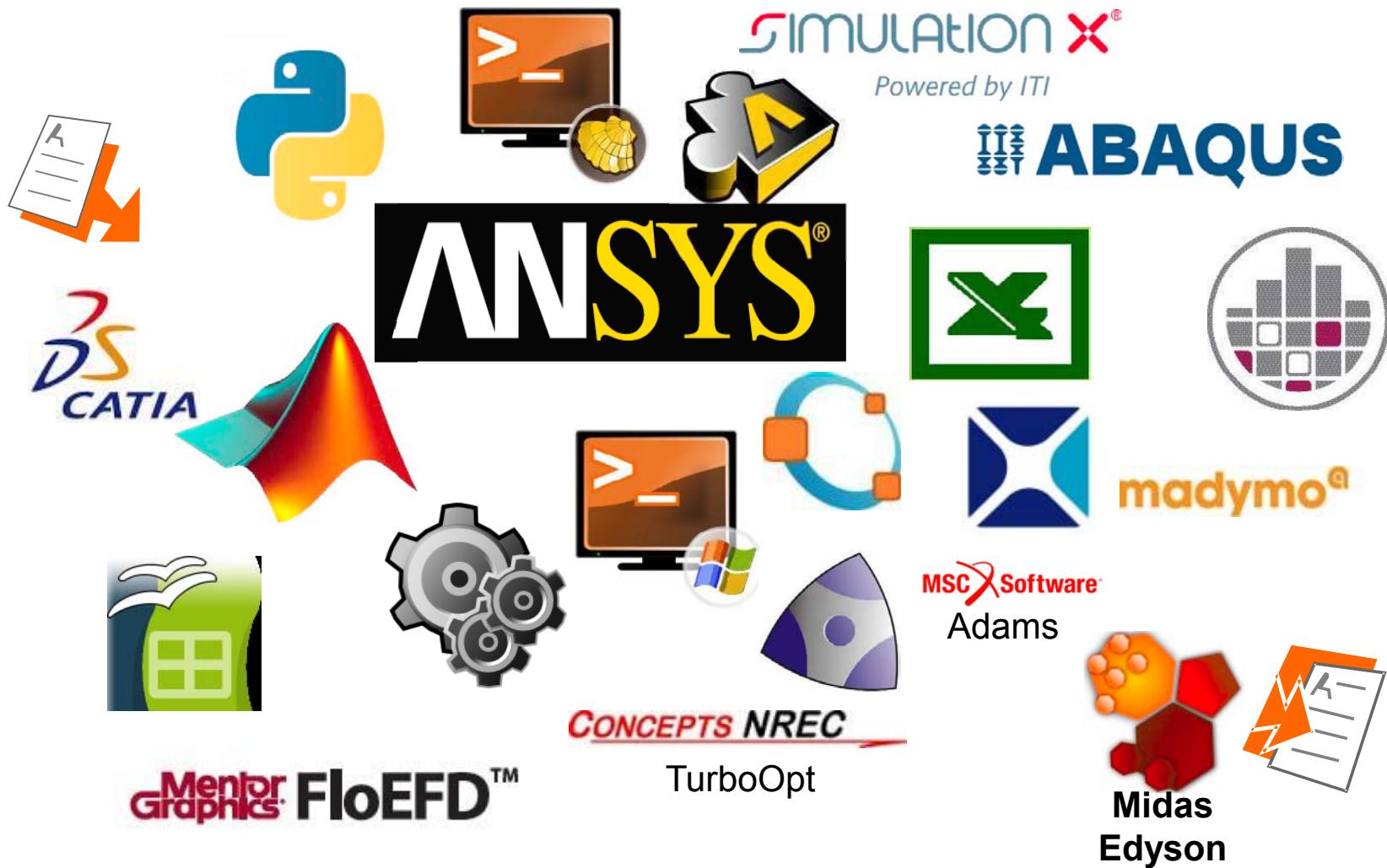


- New favorites box

Drag & Drop often used modules and use folders to organize your favorites



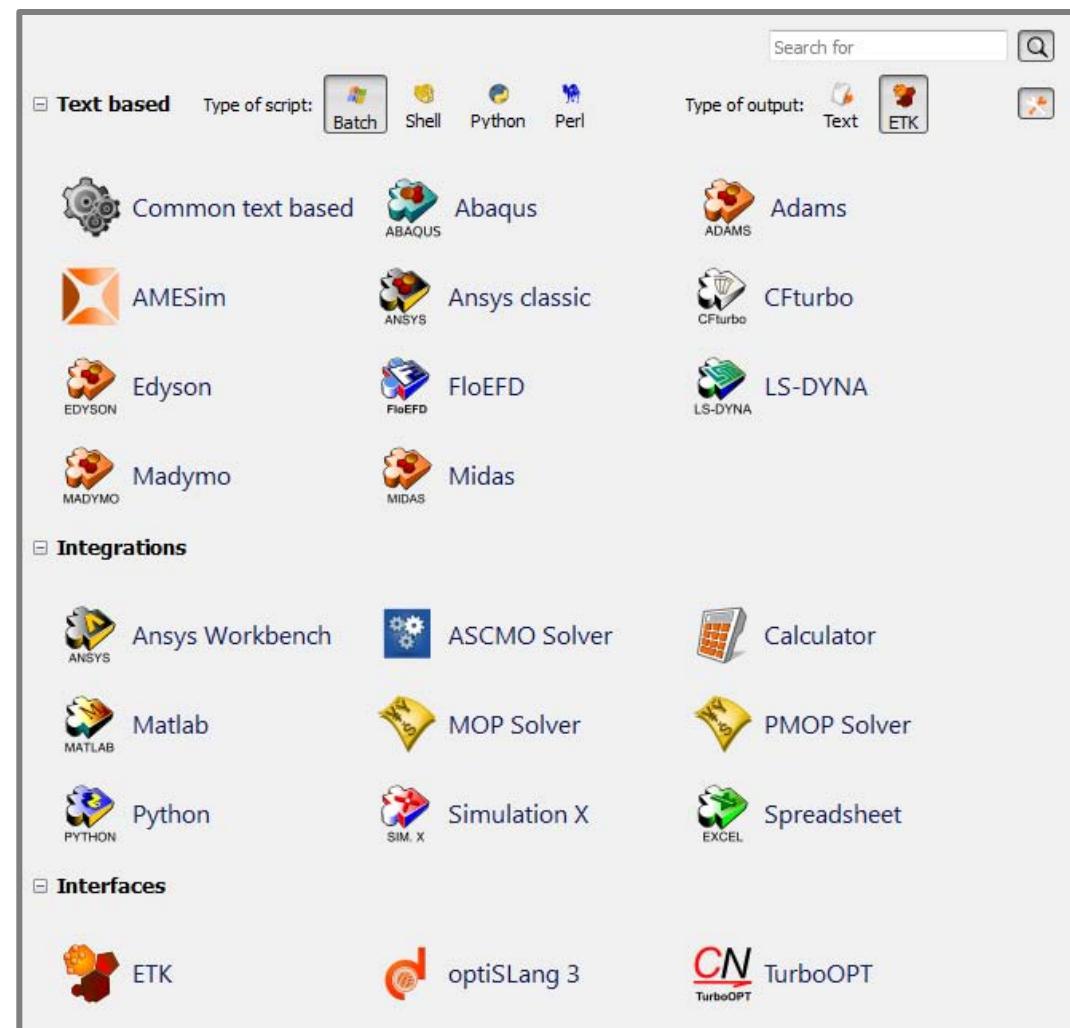
Integrations optiSLang v4.1



Integration Wizard

- Search functionality
Guarantees quick access to specific solution and good overview of all provided wizards
- Customize process chain
*Select script type
Choose extraction via Text output or ETK*

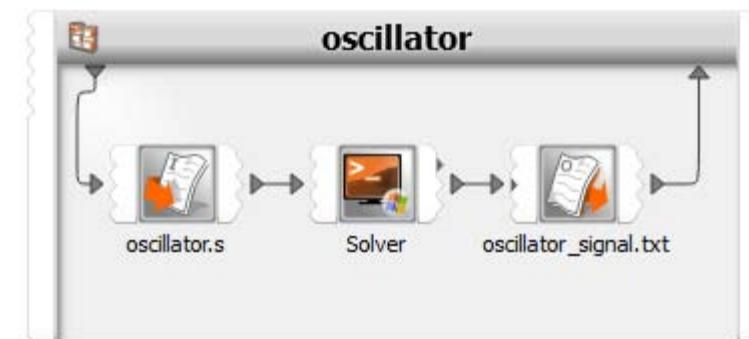
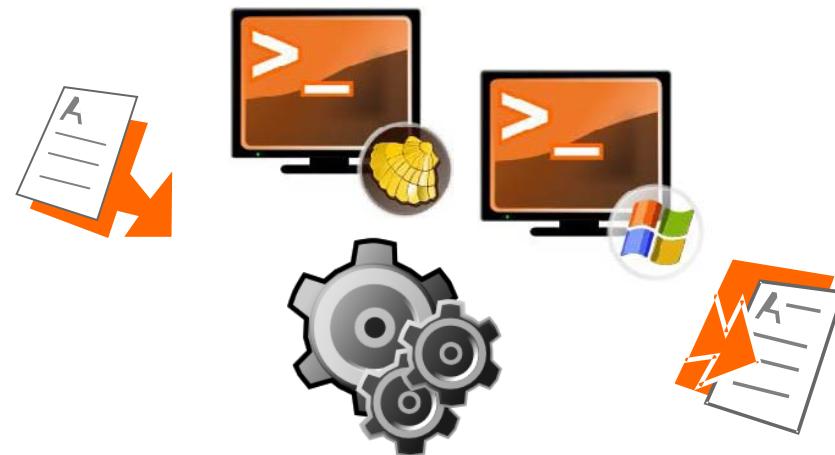
- Customized scripts
User can provide solver scripts which are automatically built in process chain



Standard text based

Text based

- Standard interface
- Define parameter/responses
- Set solver call
- Dynamically adaptable
- N parallel

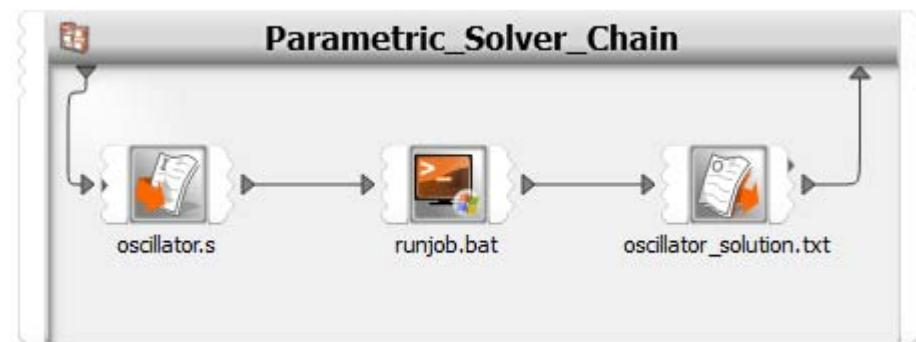


Import optiSLang v3 project



Problemfile (optiSLang <v4)

- Reads .pro file
- Defines process chain
 - Parameter
 - Responses
 - Criteria
- Result: Standard text based process chain

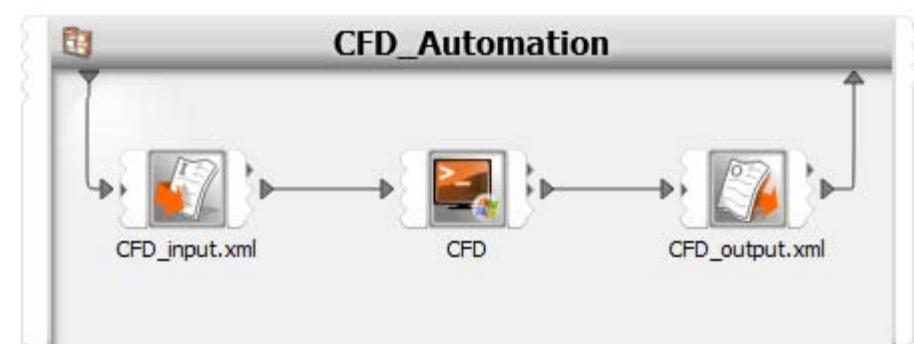


Import TurboOpt project

CONCEPTS NREC
TurboOpt

TurboOPT task file

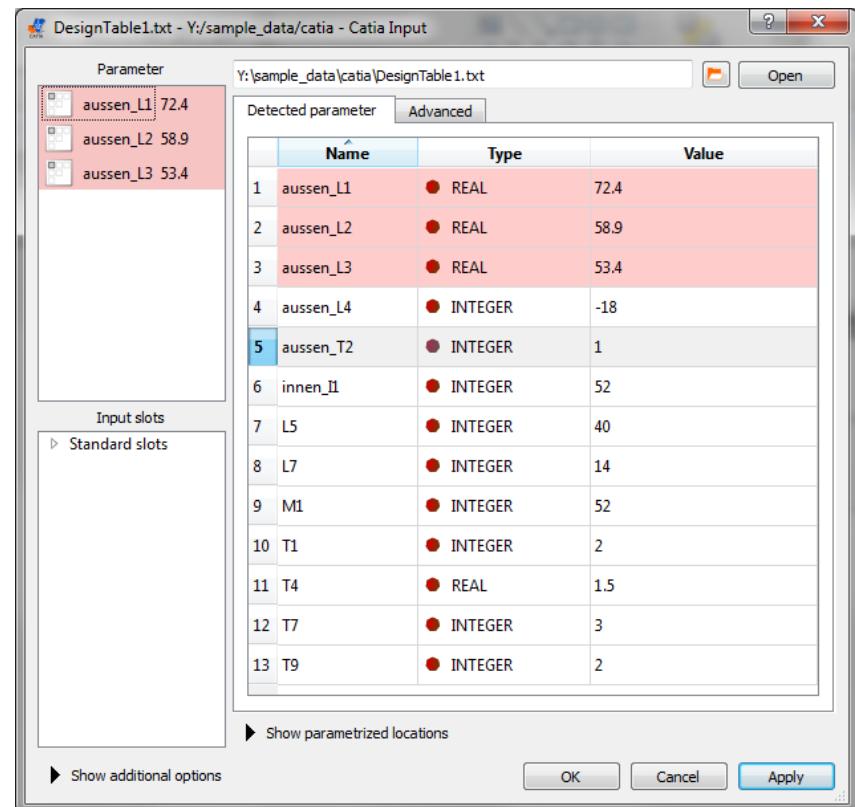
- Reads .opt file
- Defines process chain
 - Parameter
 - Responses
- Result: Standard text based process chain



Input parametrization

Catia - Text Based (v4.1.0)

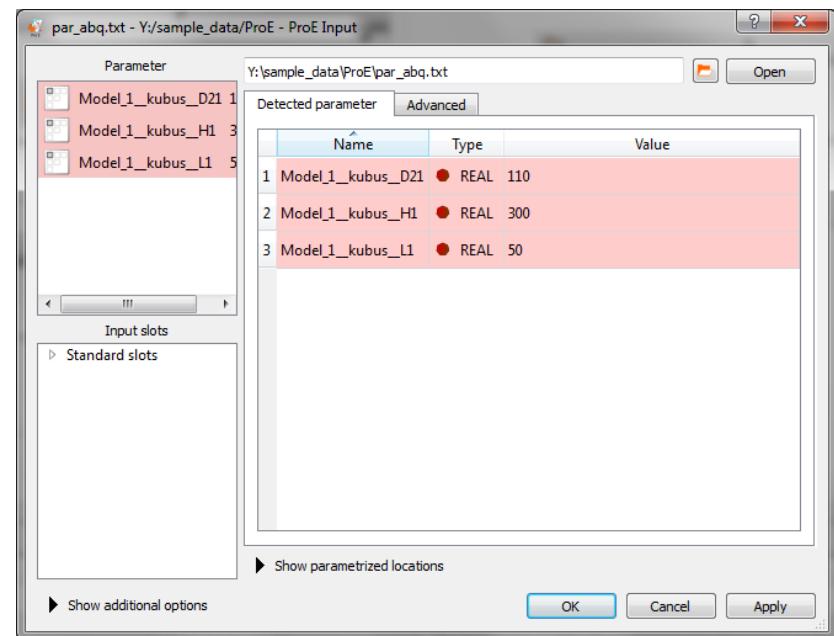
- Autoparse Designtable
- Get/set parameter
- Catia Input
- Update Catpart (batch call)
- Meshing
- Solving
- Output extraction



Input parametrization

PRO E- Text Based (v4.1.2)

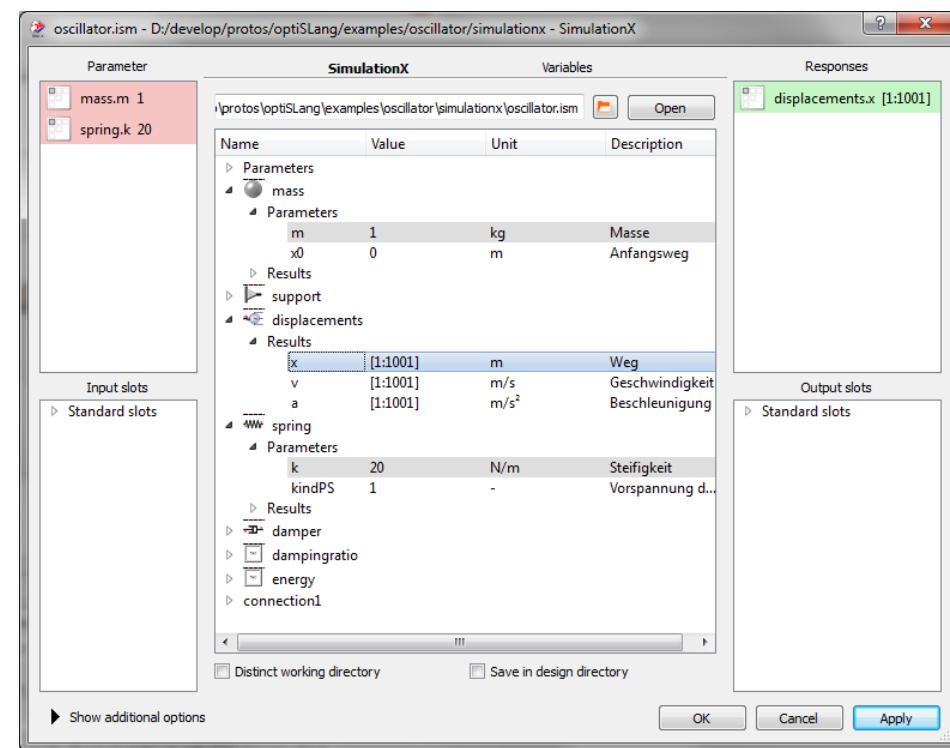
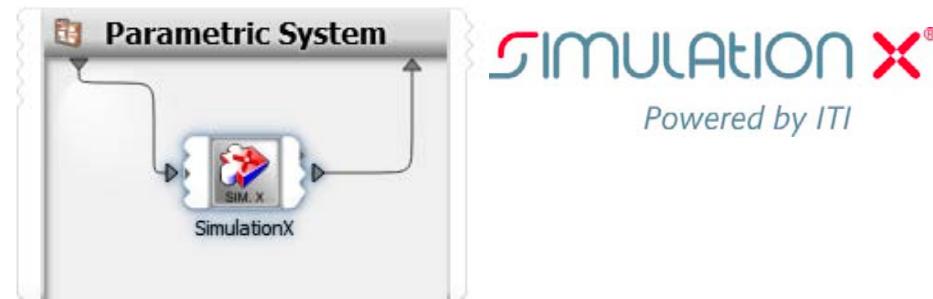
- Autoparse parameter file
- Get/set parameter
- PROE Input
- Update Geometry (batch call)
- Meshing
- Solving
- Output extraction



Interface

SimulationX - COM Interface

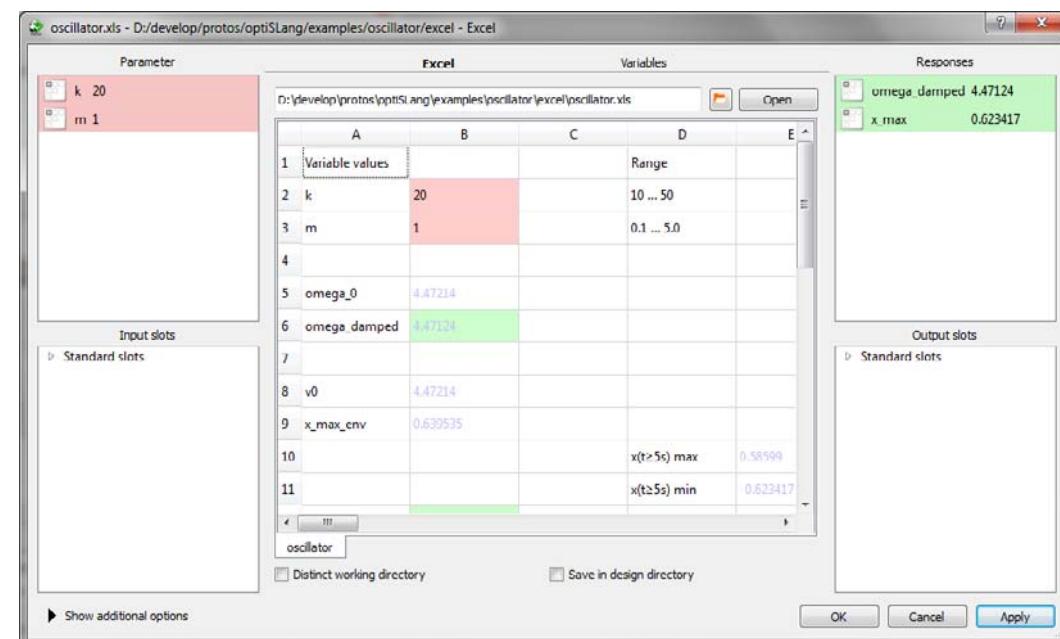
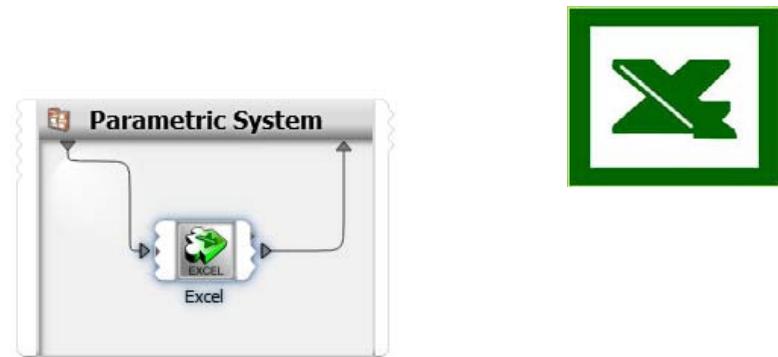
- Get/set parameter
- Get responses (signals)
- Windows only
- 4.1.2: support v3.6
- Don't stop @failed design
- 4.2: support 3.5+



Interface

Excel – COM Interface

- Get/set parameter
 - From/to cell
- Get responses
 - From cells
- (Test) run macros
- Windows only
- Serial



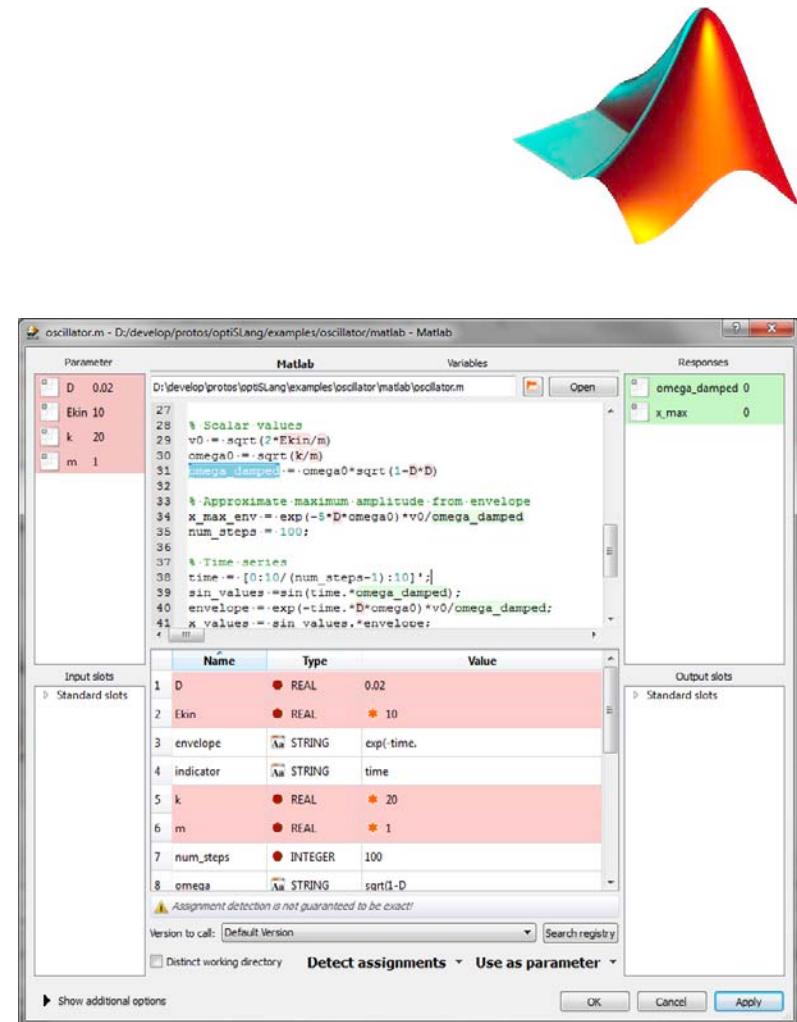
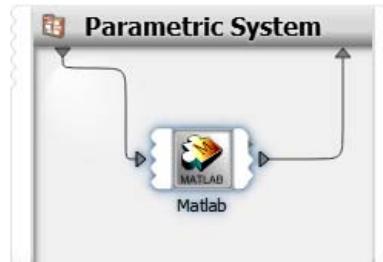
Interface

Matlab

COM Interface (Windows)

C++ API (Linux)

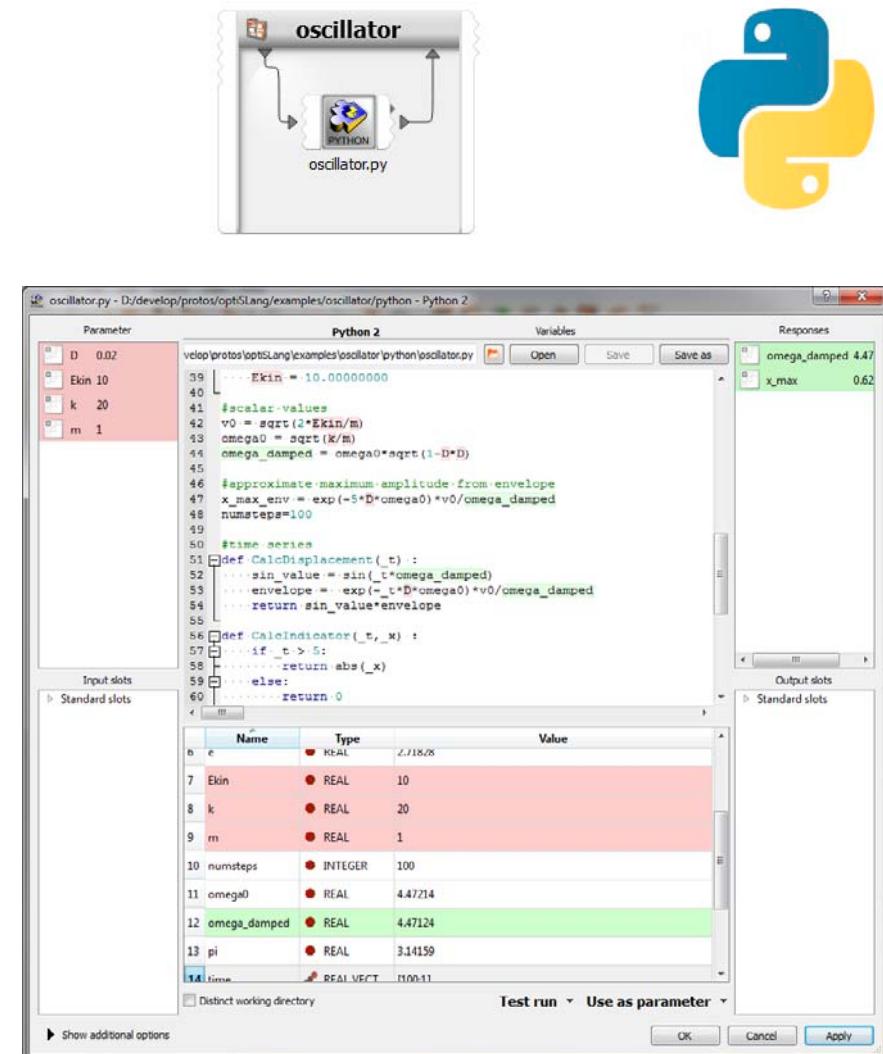
- Get/set parameter
- Get responses
- Test run
- N parallel (Windows) / serial (Linux)
- Specific version (Windows)



Interface

Python – Integrated

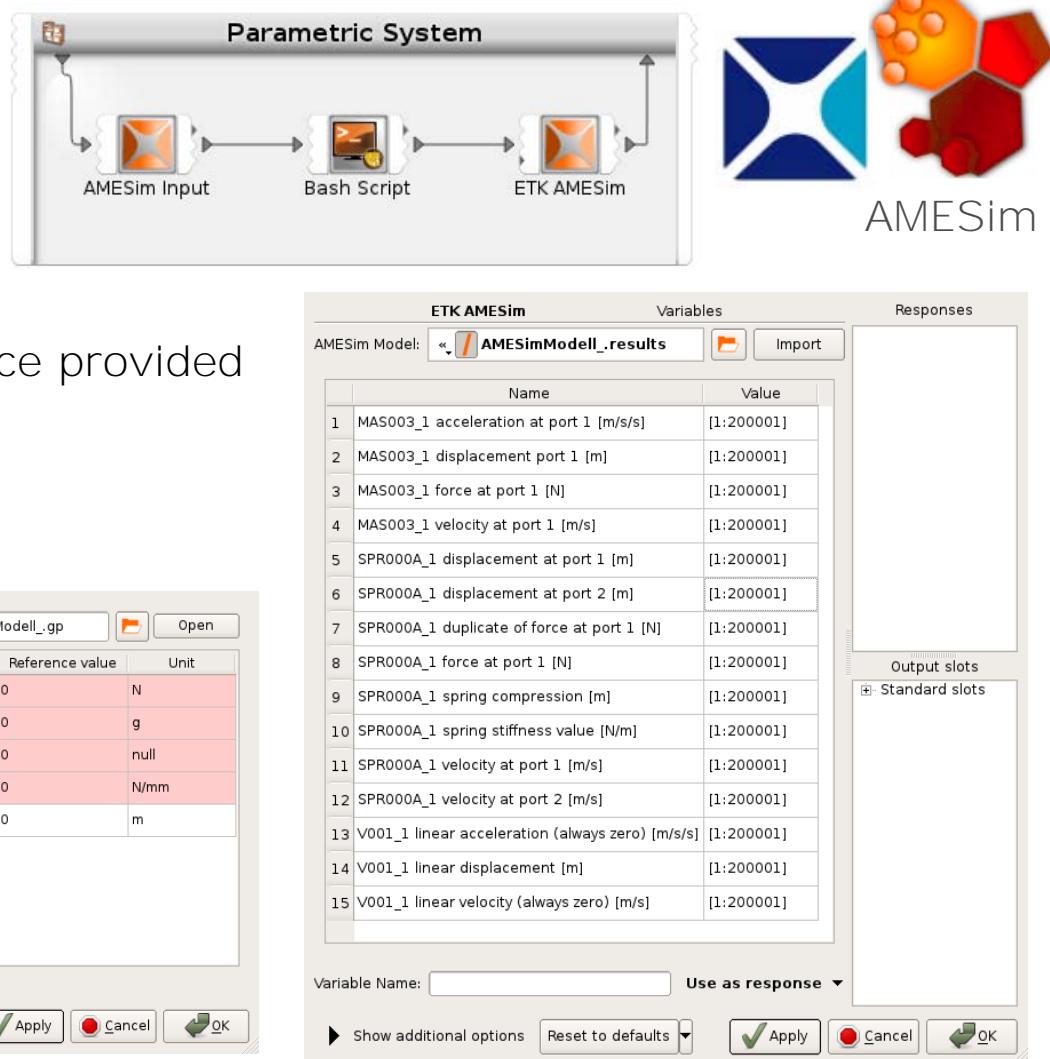
- Get/set parameter
- Get responses
- Test run
- Part of installation
- Dynardo Python modules



Integrations v4.1.2

“Built-In” Customization

- Get/set parameter
- Get responses (signal)
- Python-based scripting interface provided for AMESim



Parameter

| /home/kuehn/develop/sandbox/practice/amesim/data/00004/AMESimModell/AMESimModell_.gp | <input type="button" value="Open"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------------------------------|-------------------|-------------|---------------|-----------------|-------------|-----------------|------|-----------|-----------------|----|--------|-------|---|---|---------|------------------------------|----|--------|-------|---|---|---------|----------------------------|---|--------|-------|---|------|-----------|-------|--------|--------|-------|---|------|------|---------------------------------|-------------------|--------|-------|---|---|
| <table border="1"> <thead> <tr> <th>Name</th> <th>Title</th> <th>Nominal value</th> <th>Lower bound</th> <th>Upper bound</th> <th>Reference value</th> <th>Unit</th> </tr> </thead> <tbody> <tr><td>1 fSpring</td><td>Federvorspan...</td><td>10</td><td>-1e+06</td><td>1e+06</td><td>0</td><td>N</td></tr> <tr><td>2 mMass</td><td>Masse des Schlitten (desi...</td><td>10</td><td>-1e+06</td><td>1e+06</td><td>0</td><td>g</td></tr> <tr><td>3 frict</td><td>Friction (operation poi...</td><td>0</td><td>-1e+06</td><td>1e+06</td><td>0</td><td>null</td></tr> <tr><td>4 cSpring</td><td>title</td><td>100000</td><td>-1e+06</td><td>1e+06</td><td>0</td><td>N/mm</td></tr> <tr><td>5 x0</td><td>Auslenkung fuer Federvorspan...</td><td>fSpring/cSprin...</td><td>-1e+06</td><td>1e+06</td><td>0</td><td>m</td></tr> </tbody> </table> | | Name | Title | Nominal value | Lower bound | Upper bound | Reference value | Unit | 1 fSpring | Federvorspan... | 10 | -1e+06 | 1e+06 | 0 | N | 2 mMass | Masse des Schlitten (desi... | 10 | -1e+06 | 1e+06 | 0 | g | 3 frict | Friction (operation poi... | 0 | -1e+06 | 1e+06 | 0 | null | 4 cSpring | title | 100000 | -1e+06 | 1e+06 | 0 | N/mm | 5 x0 | Auslenkung fuer Federvorspan... | fSpring/cSprin... | -1e+06 | 1e+06 | 0 | m |
| Name | Title | Nominal value | Lower bound | Upper bound | Reference value | Unit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 fSpring | Federvorspan... | 10 | -1e+06 | 1e+06 | 0 | N | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 mMass | Masse des Schlitten (desi... | 10 | -1e+06 | 1e+06 | 0 | g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 frict | Friction (operation poi... | 0 | -1e+06 | 1e+06 | 0 | null | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 cSpring | title | 100000 | -1e+06 | 1e+06 | 0 | N/mm | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 x0 | Auslenkung fuer Federvorspan... | fSpring/cSprin... | -1e+06 | 1e+06 | 0 | m | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Input slots

Show additional options Reset to defaults

Interface



“Built-In” Customization

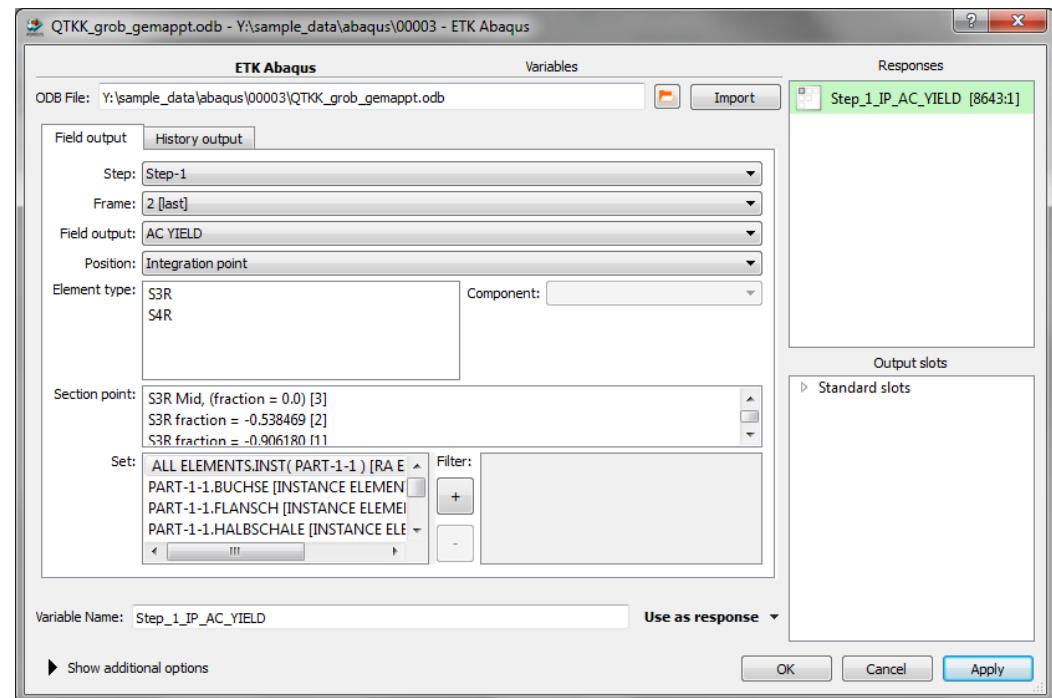
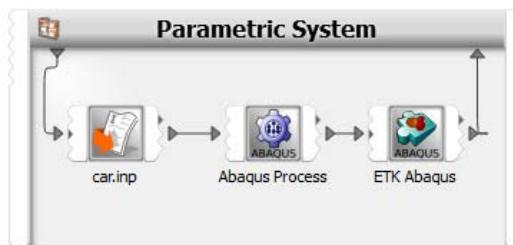
Inputnode + Outputnode

- Get/set parameter
- Get responses
- Reads/writes XML-files (Python)
- FloEFD output integrated in ETK

ETK Output extraction

Abaqus odb – ETK

- Use ETK node to extract results



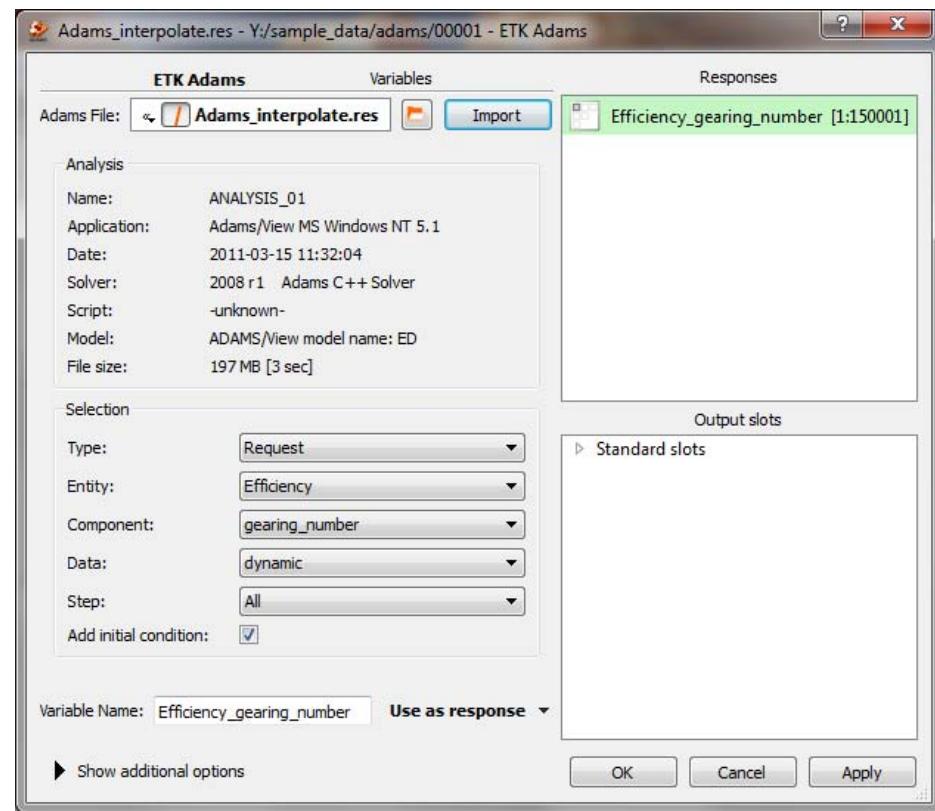
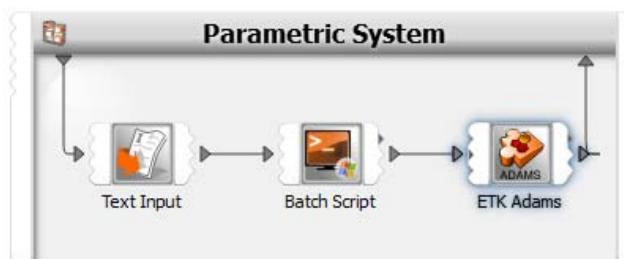
ETK Output extraction

Adams res – ETK

- Use ETK node to extract results



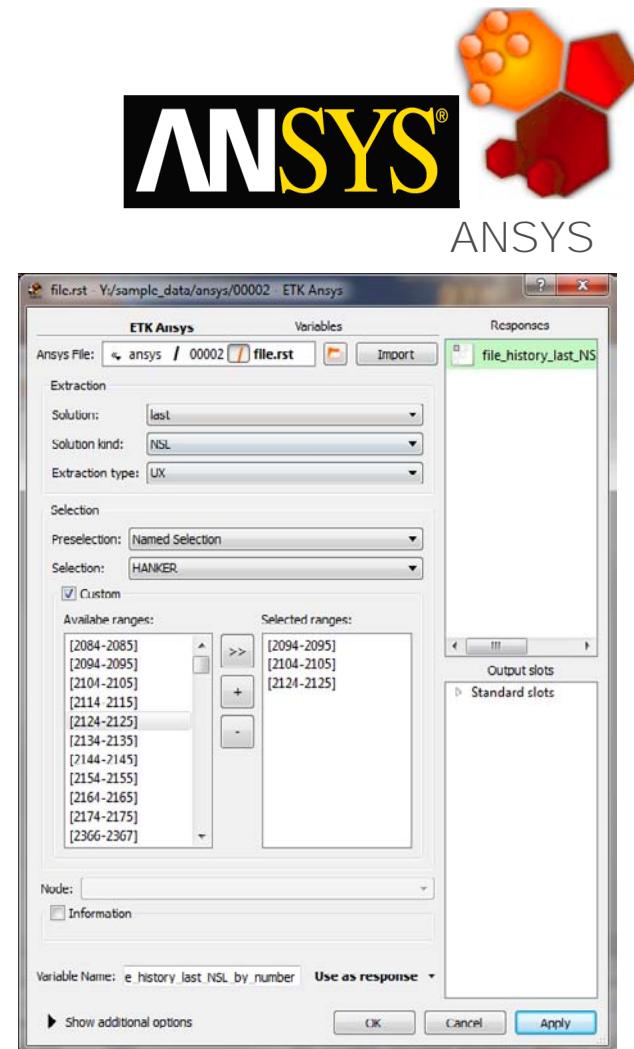
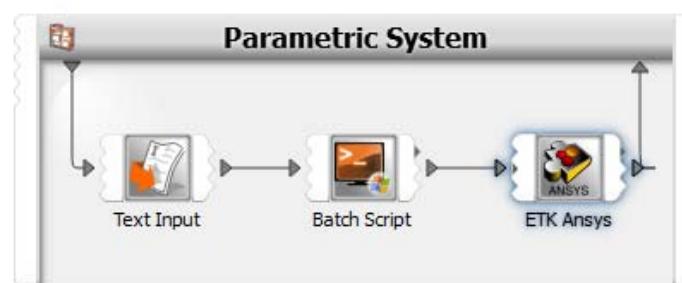
Adams



ETK Output extraction

ANSYS rst, rmg, .. binary – ETK

- Use ETK node to extract results



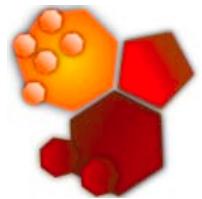
ETK Output extraction

Madymo jps – ETK

- Use ETK node to extract results



The screenshot shows the 'ETK Madymo' dialog box. In the 'ETK Madymo' tab, the 'Madymo File' is set to 'basis15.jps', 'Run ID' is 'BMW', and the signal is '/101/4051 (/Hybrid_III_50th/potentiometer_ang) - joint /101/4C'. The 'Responses' section shows a green entry for 'basis15_sig0_chn0 [1:10001]'. Below the dialog is a 'Parametric System' diagram with nodes: 'Text Input' → 'Batch Script' → 'ETK Madymo'. The 'ETK Madymo' node has an output arrow pointing to the 'Responses' section of the dialog.



ETK Output extraction

Midas XML – ETK (v4.1.2)

- Use ETK node to extract results

EDYSON – ETK (v4.1.2)

- Use ETK node to extract results

Text based parametrization

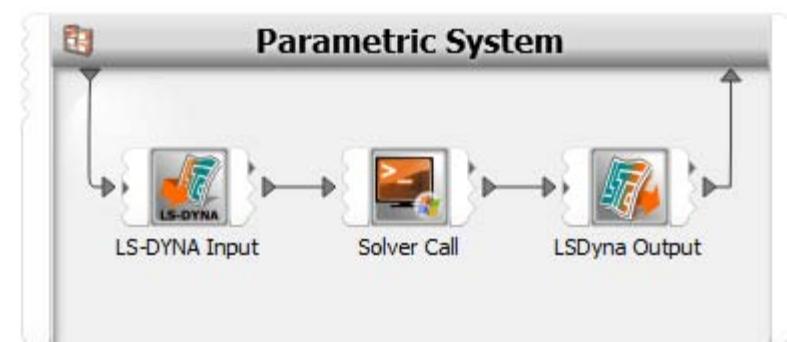
LSDyna - Text Based (v4.1.2-beta)

- Autoparse *Parameter
- Get/set parameter
- Solving



“Built-In” Customization

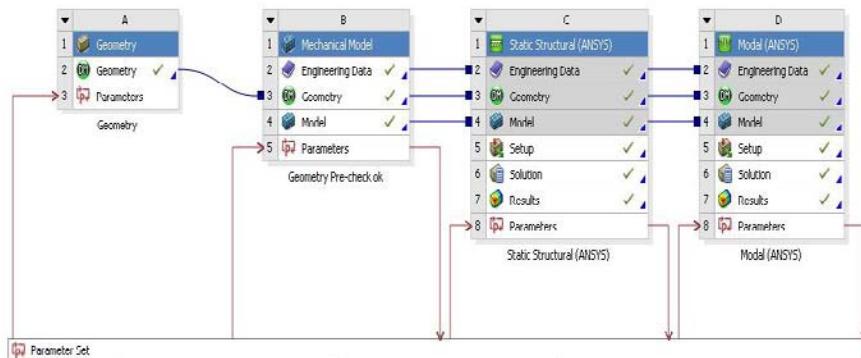
- Get responses
- Reads LS-DYNA ASCII result files (Python)
- integrated in ETK



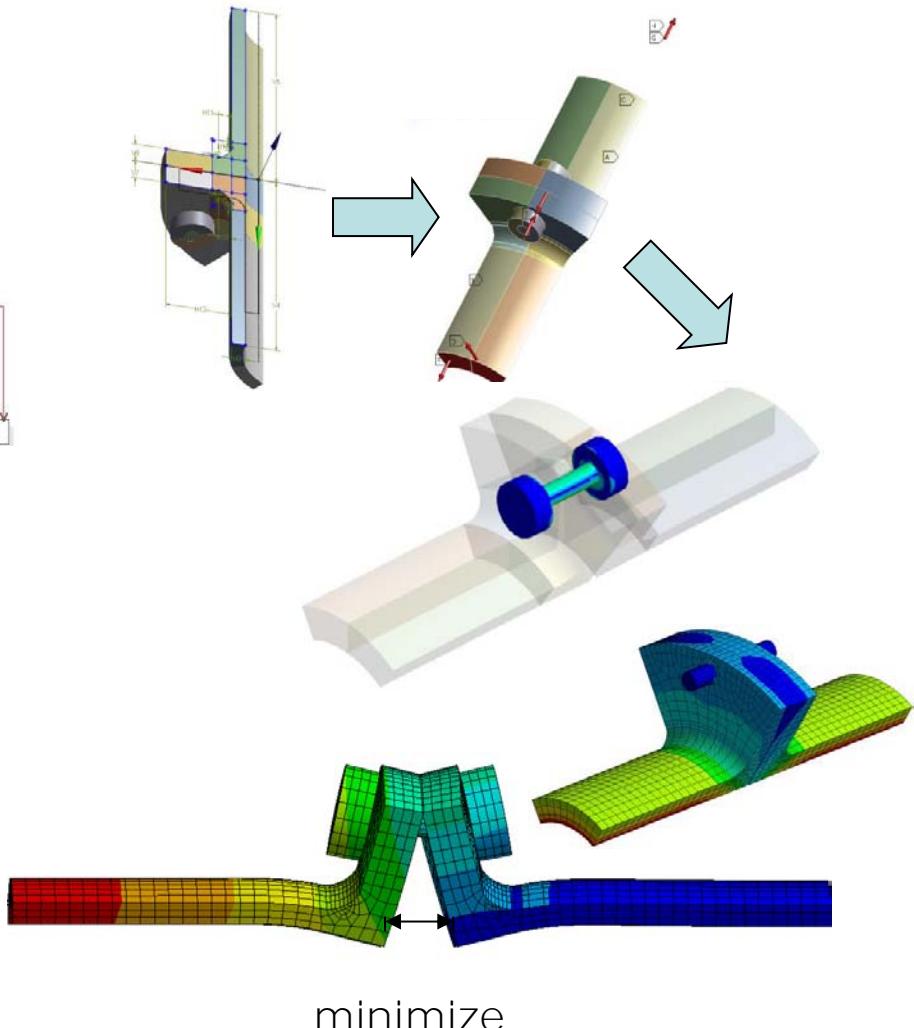
optiSLang inside ANSYS Workbench



ANSYS is a very powerful parametric modeling environment

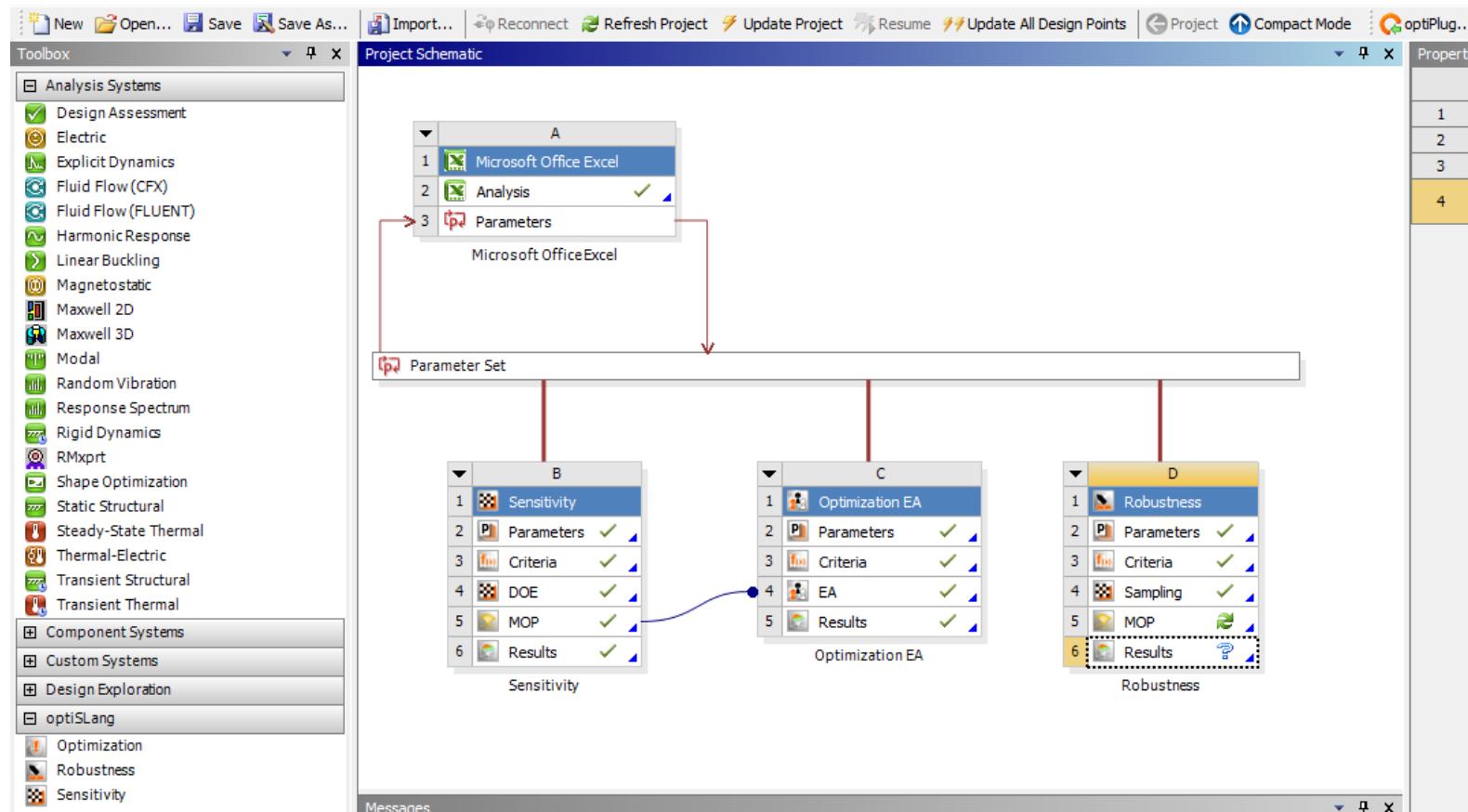


- including process automation, third party CAE integration, bidirectional CAD interfaces, parallel computing
- easy access to parametric via parameter manager
- with these important technology ANSYS Workbench is ready to address RDO task's



optiSLang inside ANSYS Workbench

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

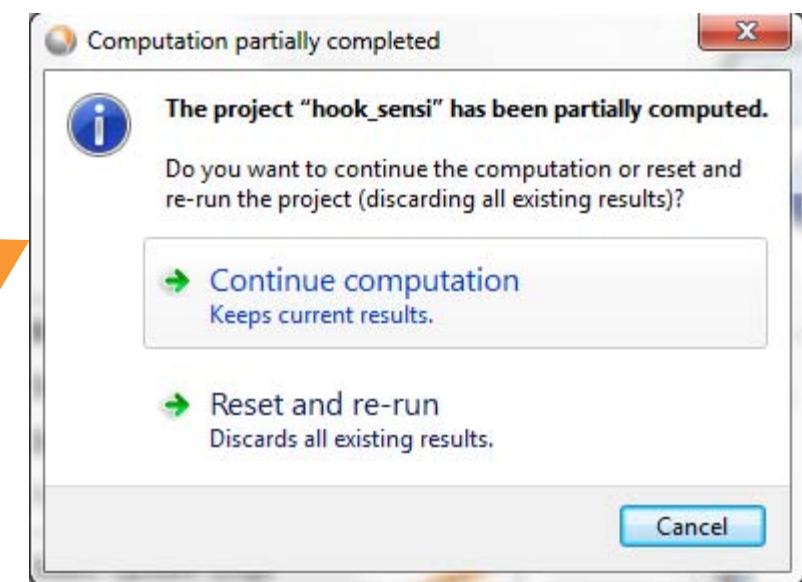
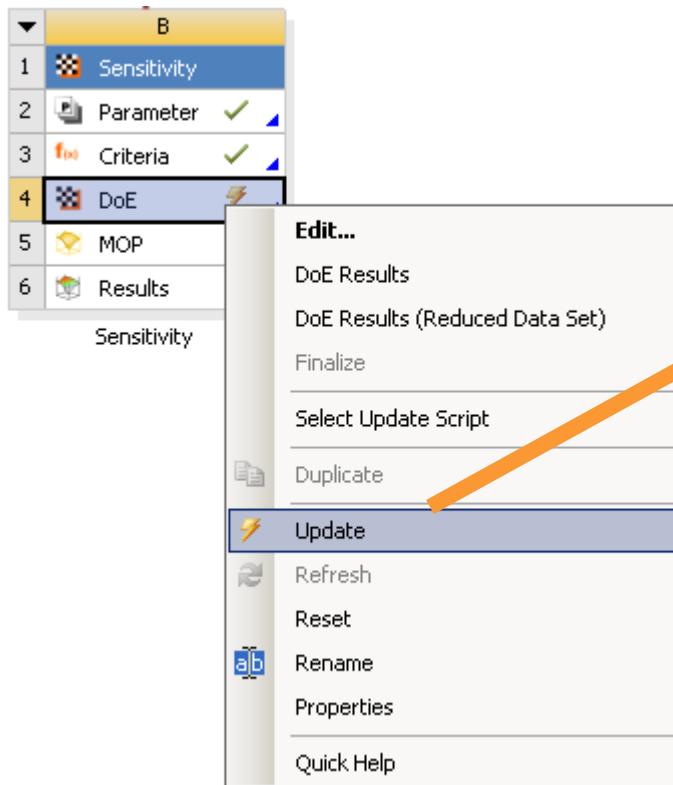


Important Features optiSLang for ANSYS v4.1



Continue Crashed Session

- using continue optiSLang only send unsolved designs
- Available in optiSLang and optiSLang inside WB (using update at optiSLang container - continue or reset can be chosen)



Recalculate Failed Designs

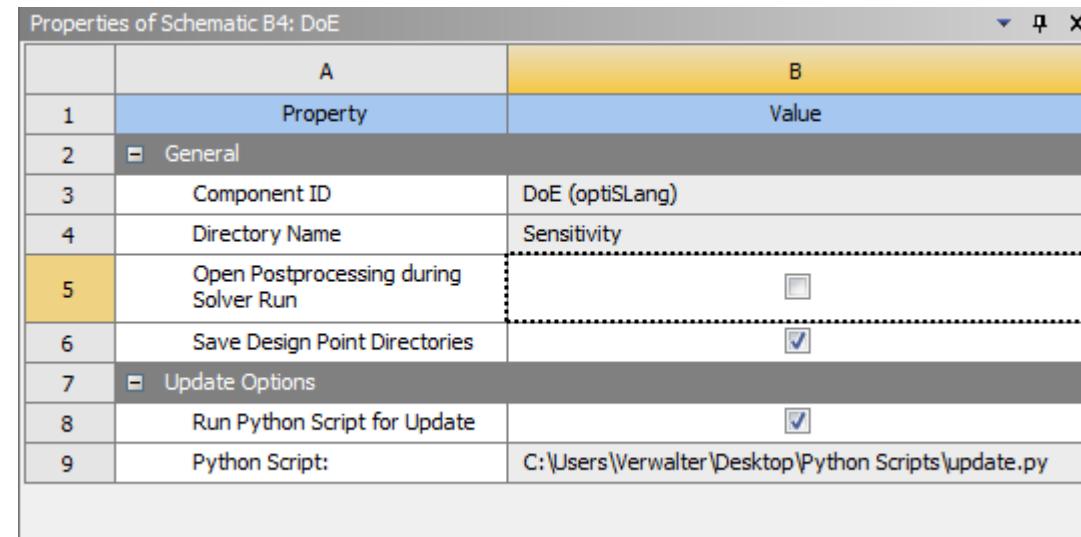
- Due to different reasons design evaluations may fail
- With “Recalculate Failed Design Points” you can start them again

The screenshot shows the "Sensitivity" dialog box. On the left, there is a table titled "Result designs" with columns: Id, Activation, Violated, Duplicates, Status, AREA01, and AREA. The table contains 10 rows of data. Row 2 is highlighted with a blue background. The "Status" column uses color coding: green for "Succeeded" and red for "Not succeeded". Rows 2, 4, 6, 7, 8, and 9 have "Not succeeded" status. On the right, a context menu is open, listing several options: Edit..., DoE Results, Show Reduced Data Set, Remove Reduced Data Set, Finalize, Recalculate Failed Design Points (which is highlighted with a blue background), Select Update Script, Duplicate, and Update.

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

Update via Python scripting

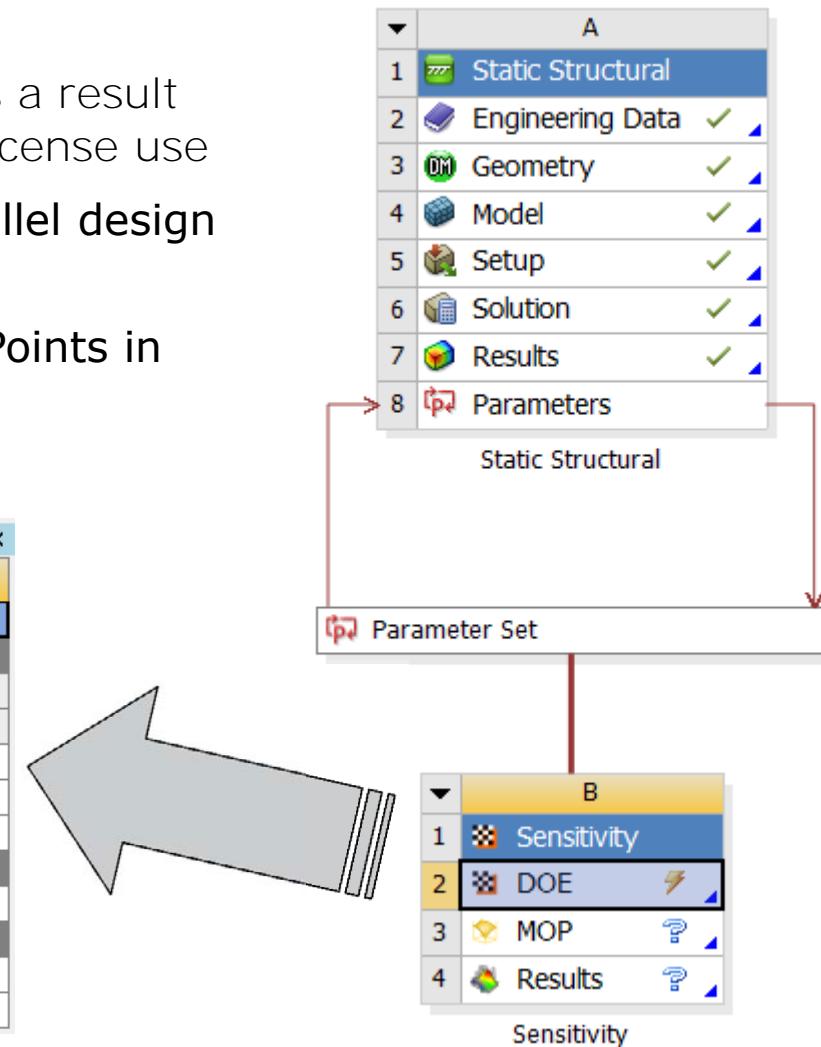
- ANSYS initialize per default an update mechanism, which updates a complete ANSYS Workbench project
- Mechanism can be overridden via python file
- optiSLang provides this feature for optiSLang design evaluations
- User has full access to his ANSYS model update



Support ANSYS HPC Parametric Pack

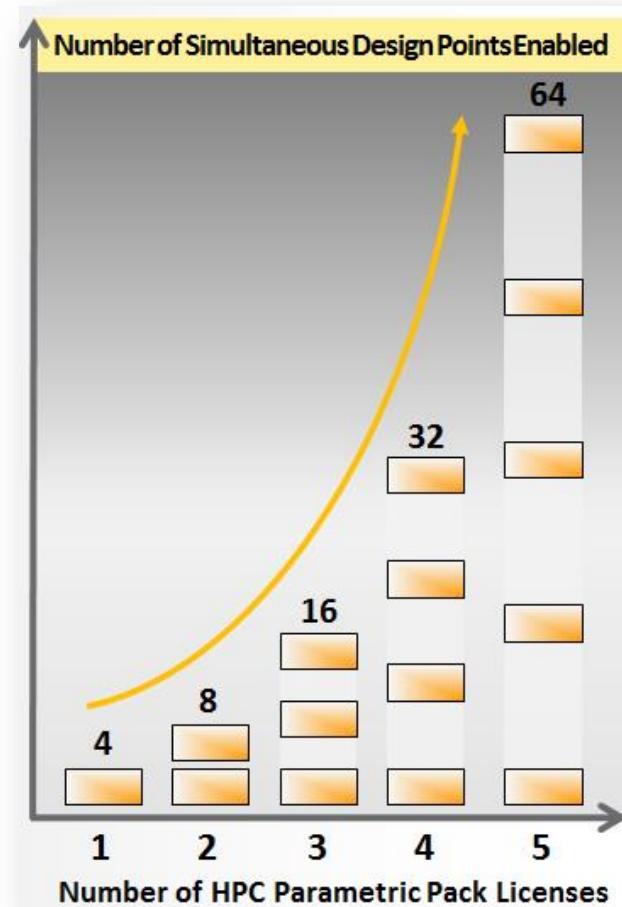
- Submit designs to WB design table, as a result ANSYS organize job distribution and license use
- Select “Use RSM Mode” to enable parallel design point submission
- Set the “Preferred Number of Design Points in Parallel” to the intended RSM job size

| 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 <input type="checkbox"/> |
| 6 | Open Postprocessing after Calculation <input type="checkbox"/> |
| 7 | Save Design Point Directories <input checked="" type="checkbox"/> |
| 8 | Notes |
| 9 | Notes |
| 10 | Update Options |
| 11 | Use RSM Mode <input checked="" type="checkbox"/> |
| 12 | Preferred Number of Design Points in Parallel 4 |



Get ready for RDO using ANSYS HPC Parametric Pack

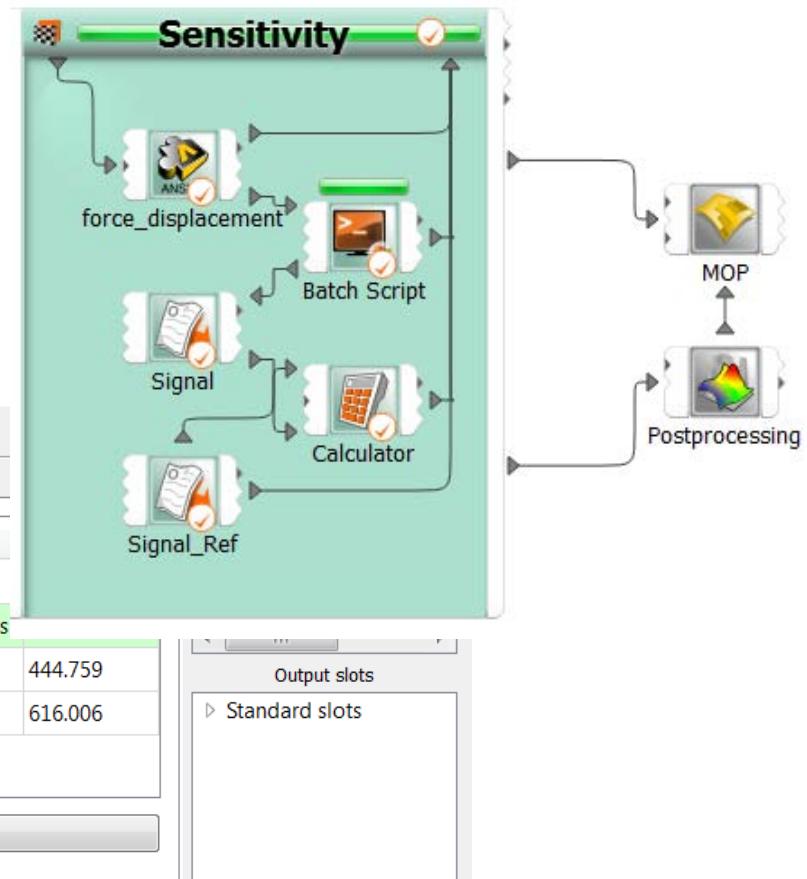
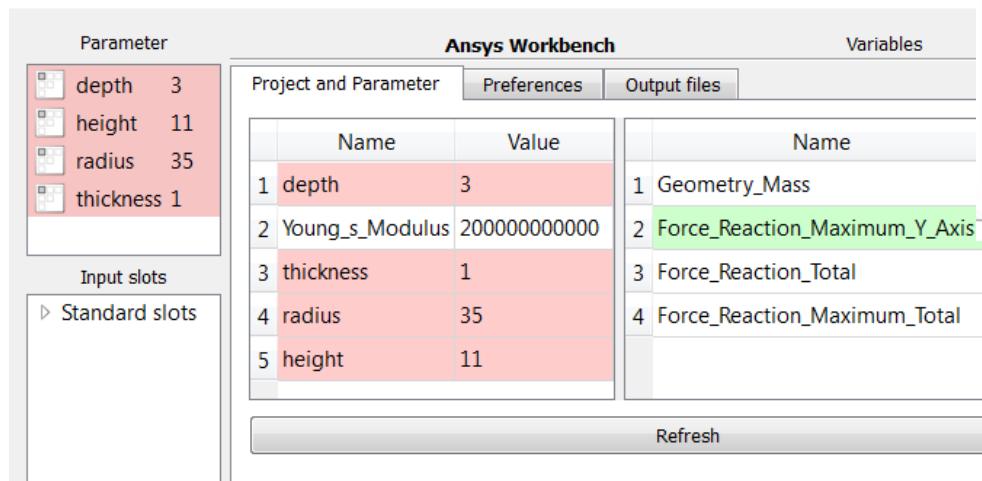
- HPC Parametric pack is available in ANSYS workbench only
- All parametric models from ANSYS classic needs to be integrated in ANSYS workbench
- optiSLang v4 supports integration node for ANSYS workbench including parameterization of additional responses (signals,..)
- Dynardo Consulting switched to HPC Parametric Pack for all ANSYS RDO tasks Q2/2013



The WB Node (ANSYS inside optiSLang)

Use ANSYS WB in your optiSLang flow provides flexibility to extend the process chain

- More flexibility (compared to “inside”)
 - e.g. use signals for calibration
- Batch call using HPC Parametric Packs



ETK 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

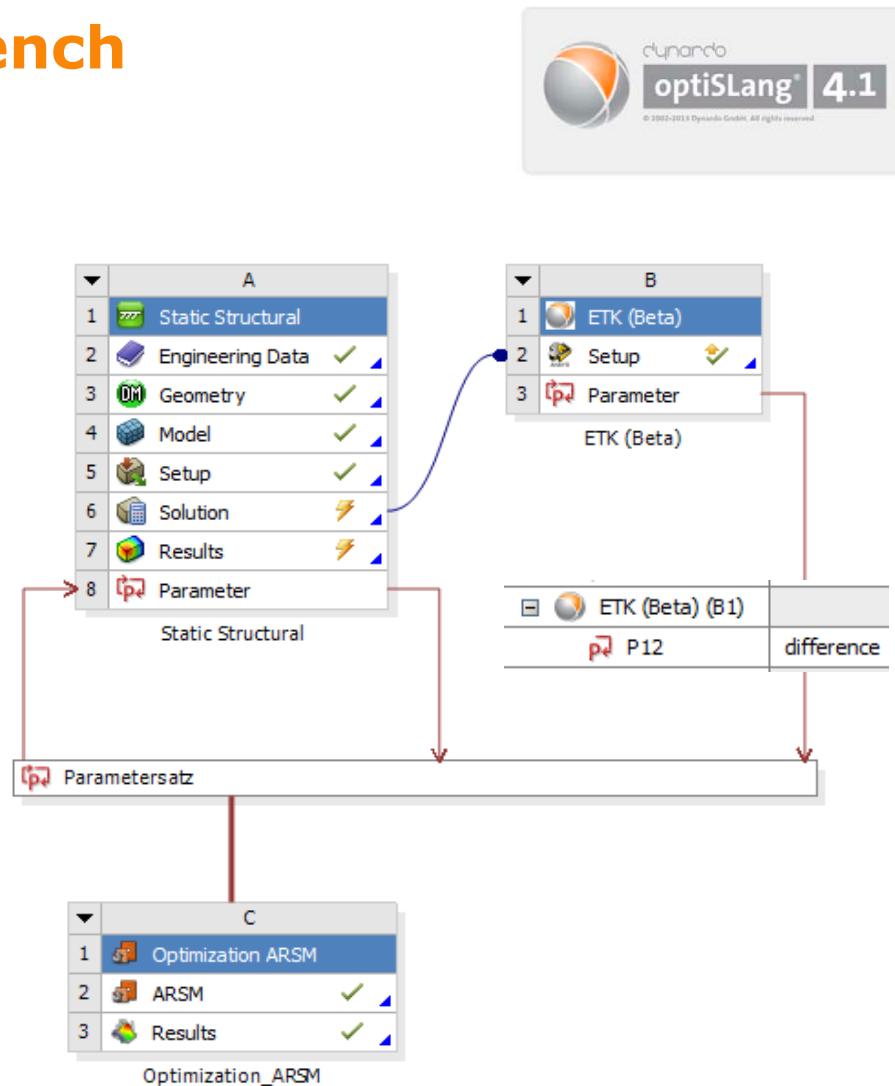
Use functionality of Extraction nodes inside ANSYS Workbench to work with data which is not supported via ANSYS, i.e. vectors, signals, matrices

- Register response values

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

- Parameter identification

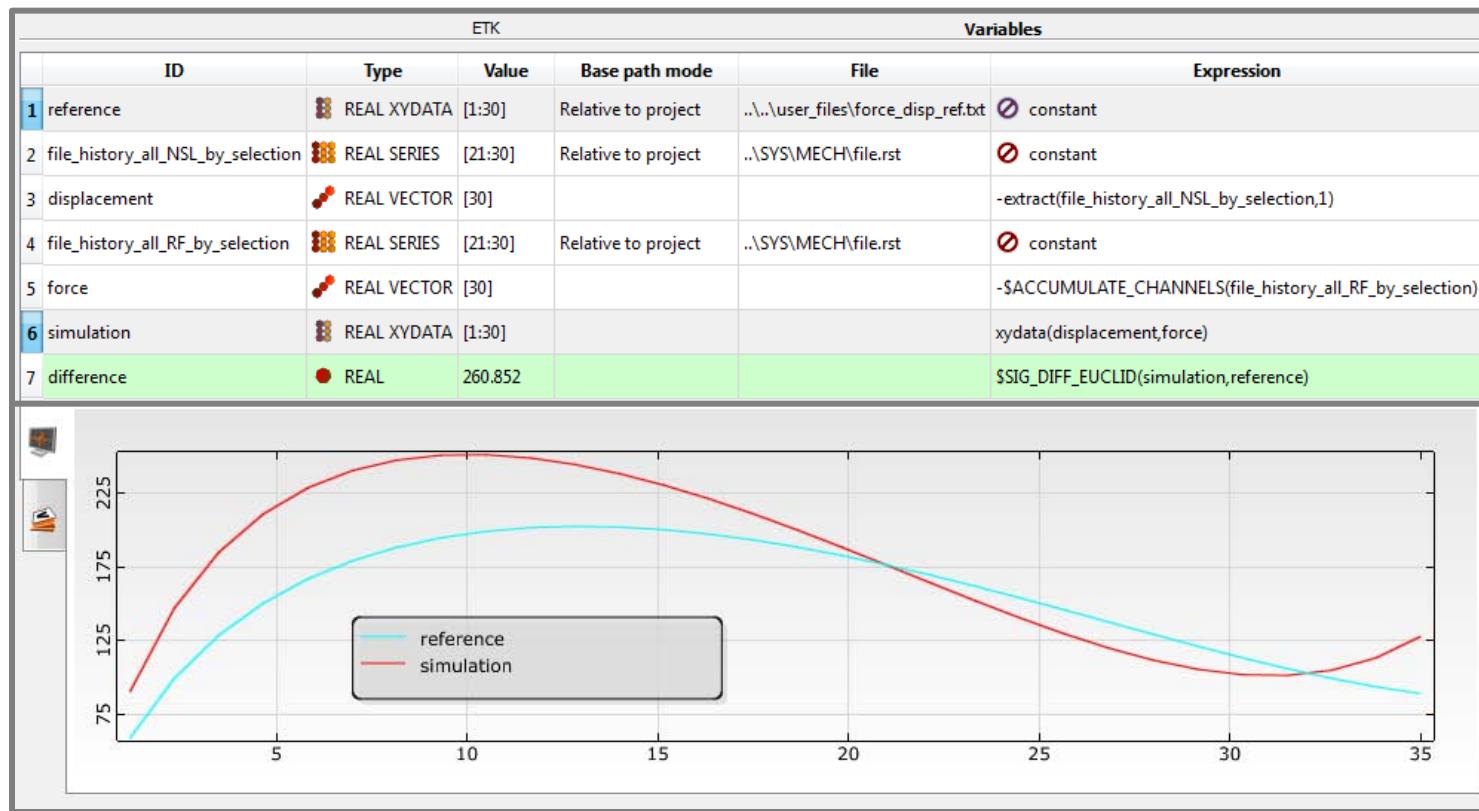
Use parameterized values to set up calibration task through optiSLang inside ANSYS Workbench



ETK inside ANSYS Workbench



- Signal visualization, extraction and analysis inside ANSYS Workbench



Licensing news

optiSLang for ANSYS bundle

Worldwide licensenses = 150% of limited area licenses

MOP solver license Pack (10 Licenses)

Backup



Product bundle: optiSLang for ANSYS

- Product bundle containing
 - optiSLang inside ANSYS Workbench
 - optiSLang 4 including WB node
 - optiPlug
- Use same dialogs, same algorithms, same post processing
 - Wizard driven set up of analysis
 - Algorithms with proven default settings
 - Minimization of necessary user input to design/robustness space, constraints and objectives
 - Maximize the outcome/benefits of parametric studies with the help of Wizards, best practice algorithms and the Metamodel of Optimal Prognosis (MOP)



Variables Tab

Post process Outputs

Calculator in each integration node

- Combine Signals, Outputs, ...
- Forward your results

oscillator_signal.txt - C:\Users\Public\Documents\Dynardo\optiSLang\4.1.0\examples\oscillator\ascii_parametrize - Text Output

| Text Output | | | Variables | |
|-------------|--------------|---------------------|-------------------------|---|
| ID | Type | Value | File | Expression |
| 1 | disp | REAL VECTOR [100:1] | "oscillator_signal.txt" | constant |
| 2 | time | REAL VECTOR [100:1] | "oscillator_signal.txt" | constant |
| 3 | disp_channel | REAL XYDATA [1:100] | | $\text{xydata}(\text{time}, \text{disp})$ |
| 4 | max0 | REAL | 0.95086 | $\text{SIG_MAX_Y}(\text{disp_channel})$ |
| 5 | max2 | REAL | 0.74887 | $\text{SIG_MAX_Y_SLOT}(\text{disp_channel}, 2, 0, 10, 0)$ |
| 6 | max4 | REAL | 0.66297 | $\text{SIG_MAX_Y_SLOT}(\text{disp_channel}, 4, 0, 10, 0)$ |
| 7 | max6 | REAL | 0.53689 | $\text{SIG_MAX_Y_SLOT}(\text{disp_channel}, 6, 0, 10, 0)$ |
| 8 | max8 | REAL | 0.45563 | $\text{SIG_MAX_Y_SLOT}(\text{disp_channel}, 8, 0, 10, 0)$ |

Add variable ▾

Show additional options

OK

oscillator_signal.txt - C:\Users\Public\Documents\Dynardo\optiSLang\4.1.0\examples\oscillator\ascii_parametrize - Text Output

| Text Output | | Variables | | Responses | |
|----------------------------------|------------------------|--------------|-----------|-----------|----------------------|
| Text File: oscillator_signal.txt | | Import | | | |
| 1 | SLtxt 5.1.2 ... | Object: | T_VALUES | Responses | disp_channel [1:100] |
| 2 | | Object.info: | 2 3 100 2 | | max0 0.95086 |
| 3 | | | | | max2 0.74887 |
| 4 | | | | | max4 0.66297 |
| 5 | | | | | max6 0.53689 |
| 6 | | | | | max8 0.45563 |
| 7 | .. 0.00000 .. 0.00000 | | | | |
| 8 | .. 0.10101 .. 0.43260 | | | | |
| 9 | .. 0.20202 .. 0.77145 | | | | |
| 10 | .. 0.30303 .. 0.95086 | | | | |
| 11 | .. 0.40404 .. 0.93801 | | | | |
| 12 | .. 0.50505 .. 0.73890 | | | | |
| 13 | .. 0.60606 .. 0.39646 | | | | |
| 14 | .. 0.70707 .. -0.01867 | | | | |
| 15 | .. 0.80808 .. -0.42266 | | | | |
| 16 | .. 0.90909 .. -0.73539 | | | | |
| 17 | .. 1.01010 .. -0.89630 | | | | |
| 18 | .. 1.11111 .. -0.87615 | | | | |
| 19 | .. 1.21212 .. -0.68216 | | | | |
| 20 | .. 1.31313 .. -0.35602 | | | | |
| 21 | .. 1.41414 .. 0.03505 | | | | |
| 22 | .. 1.51515 .. 0.41216 | | | | |
| 23 | .. 1.61616 .. 0.70057 | | | | |
| 24 | .. 1.71717 .. 0.84453 | | | | |
| 25 | .. 1.81818 .. 0.81801 | | | | |
| 26 | .. 1.91919 .. 0.62934 | | | | |
| 27 | .. 2.02020 .. 0.31892 | | | | |
| 28 | .. 2.12121 .. -0.04934 | | | | |
| 29 | .. 2.22222 .. -0.40120 | | | | |
| 30 | .. 2.32323 .. -0.66700 | | | | |

Output slots

Standard slots

Instant visualization in integration and output nodes

- Quick visual feedback

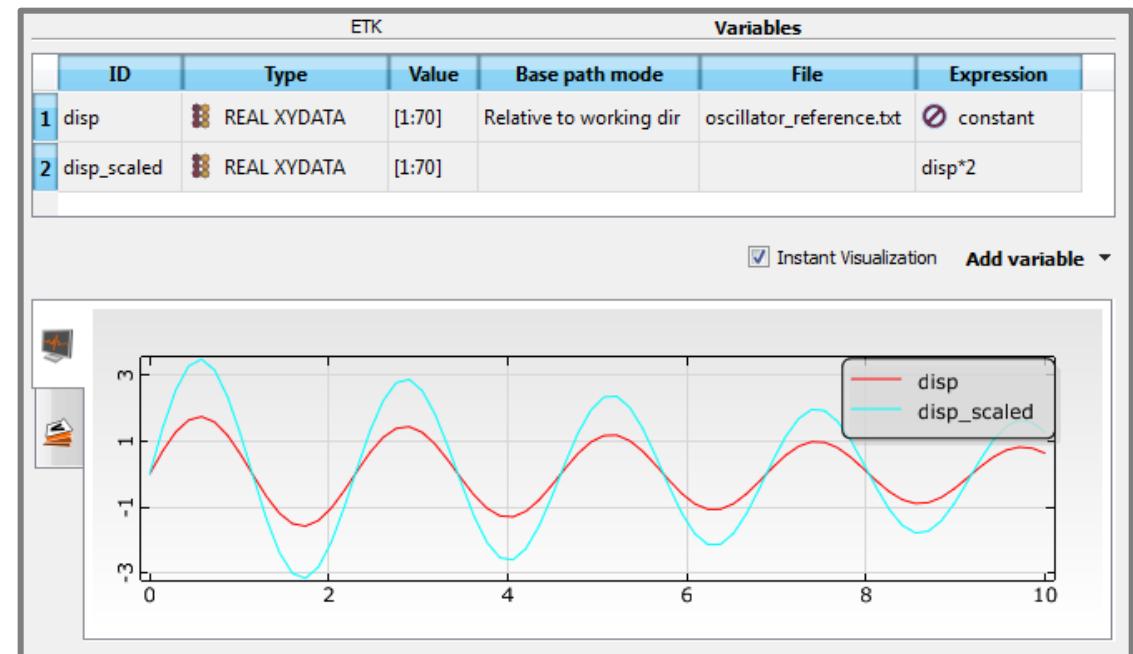
*Plot for Vectors, Signals, Matrices
Tables for Scalars, Vectors,
Signals and Matrices*

- Parametrized values

*Show reference value in Output,
Python, Matlab nodes*

- Multiple values

*Show and compare values of
registered outputs and derived
variables*



Search function in Text editors

- Search

Forward/backward

Case sensitive/insensitive

Regular expressions supported

- In any text editor

Bat/SH/Perl/Python Script node

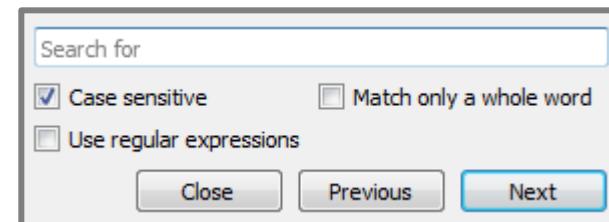
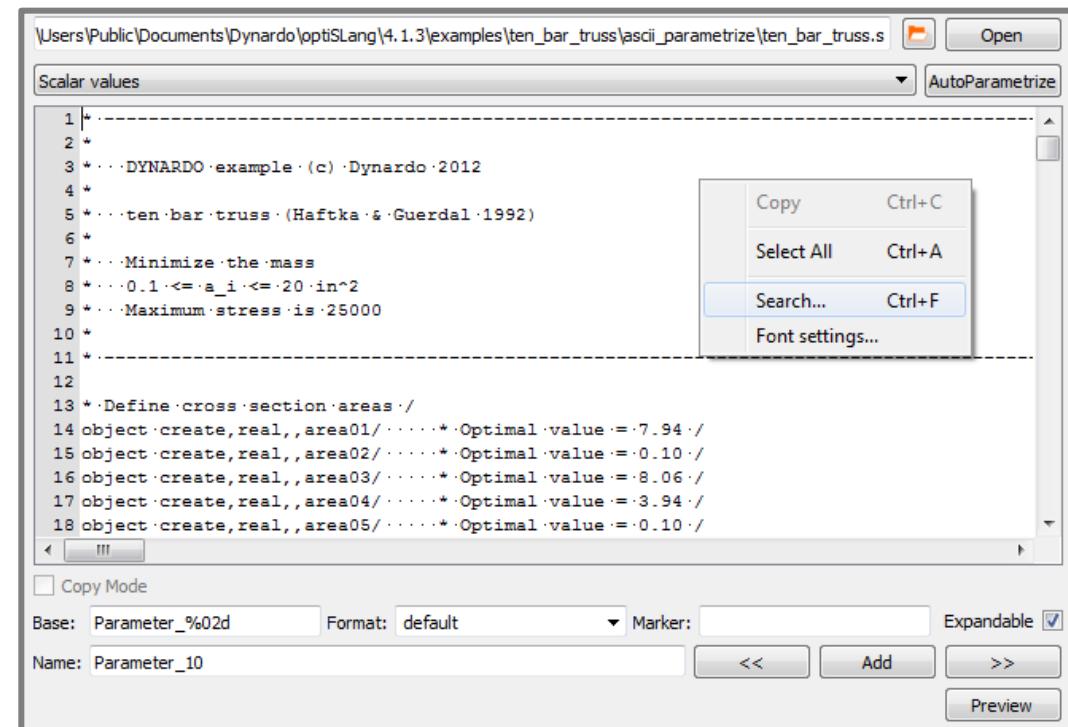
Matlab/Python/Octave node

Text output node

- Easy access

Via shortcut "CTRL+F"

Via Context menu



Update 4.1.2

Excel Interface

- Expandable Extraction

Extraction vectors up to the first not extractable cell

- Close after Run

Stabilize Excel

Allows to remove directory after run if file is saved in directory

- Read mode

Allows Reevaluation if files are saved in design directories

| | A | B | C |
|----|--------------------------|----------|------------|
| 12 | x_max | 0.623417 | |
| 13 | | | |
| 14 | | | |
| 15 | x(t) = e^-D\omega_0 t... | | |
| 16 | | | |
| 17 | Step | time | disp |
| 18 | 1 | 0 | 0 |
| 19 | 2 | 0.10101 | 0.432603 |
| 20 | 3 | 0.20202 | 0.771453 |
| 21 | 4 | 0.30303 | 0.95086 |
| 22 | 5 | 0.40404 | 0.938012 |
| 23 | 6 | 0.505051 | 0.738905 |
| 24 | 7 | 0.606061 | 0.39646 |
| 25 | 8 | 0.707071 | -0.0186747 |
| 26 | 9 | 0.808081 | -0.422662 |
| 27 | 10 | 0.909091 | -0.735385 |
| 28 | 11 | 1.0101 | |

oscillator

Show macro options

Distinct working directory Save in design directory Close after run

Hide additional options

Auto-save behavior: No auto-save

Read mode

The WB Node (ANSYS inside optiSLang)

- The workbench node directly connects to the workbench project and gets the inputs and outputs from the parameter set

