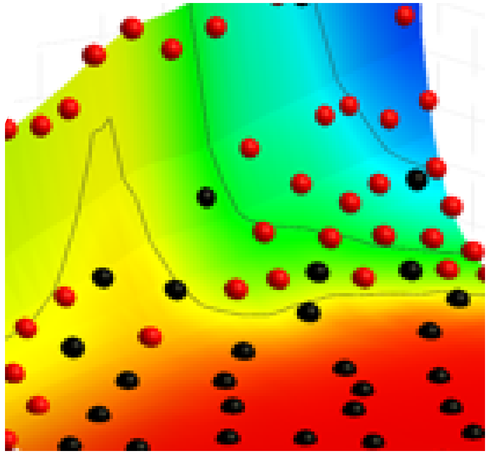
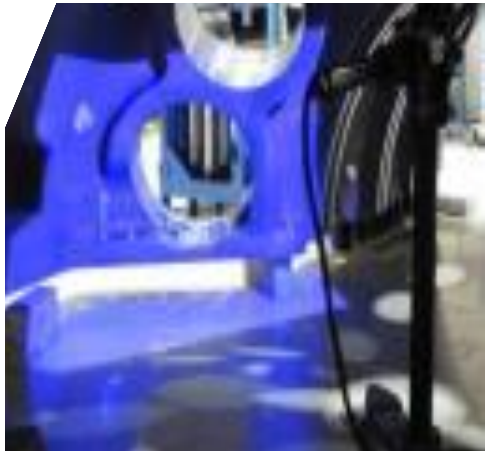
presented at the 17th Weimar Optimization and Stochastic Days 2020 | Source: www.dynardo.de/en/library

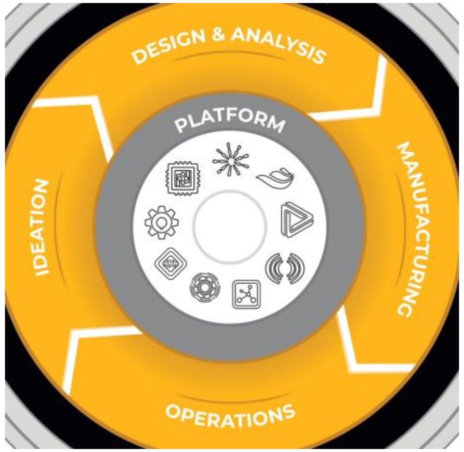
Recent developments and applications of Field Meta Modelling

Sebastian Wolff, WOST 17 (June 2020)

Agenda







Field meta models and add-on SoS

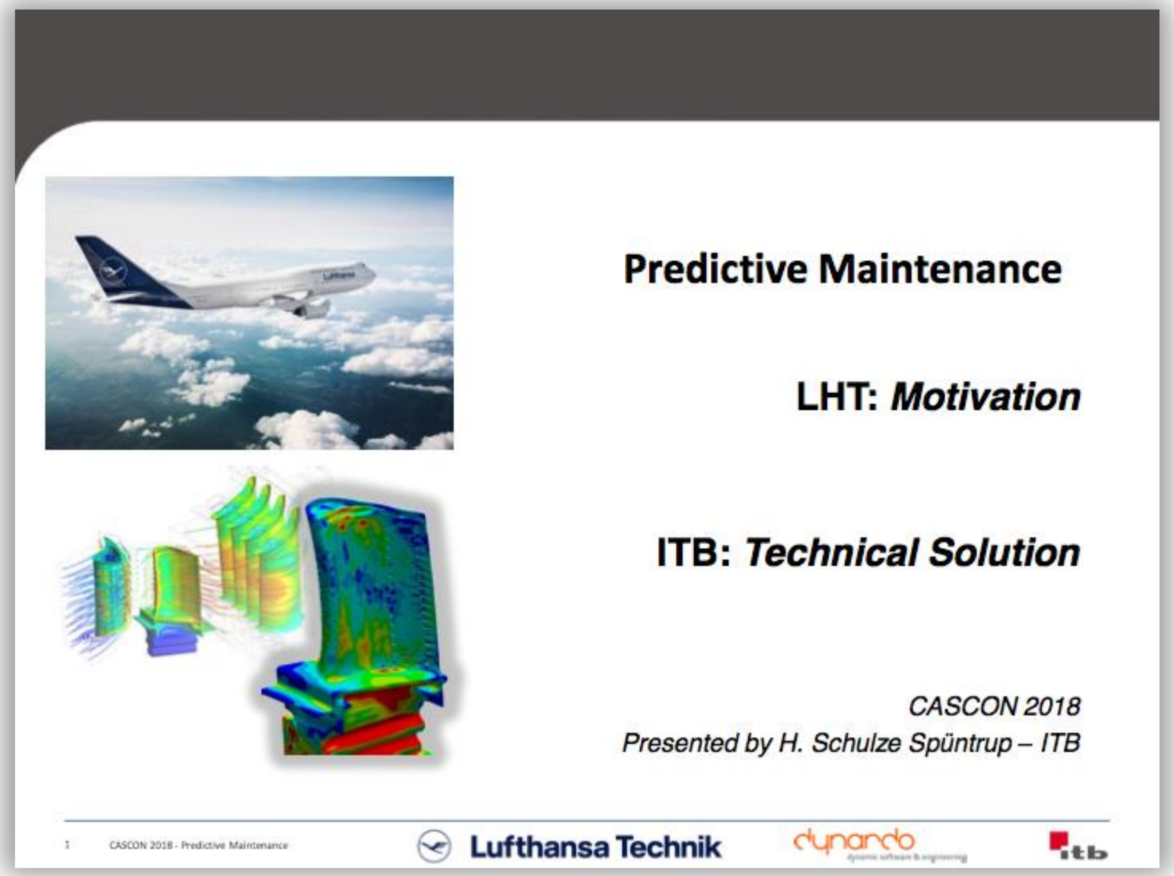
Dealing with Measurements

Data I/O and process integration

Field meta models

Customer example: ROM for Predictive Maintenance Digital Twin

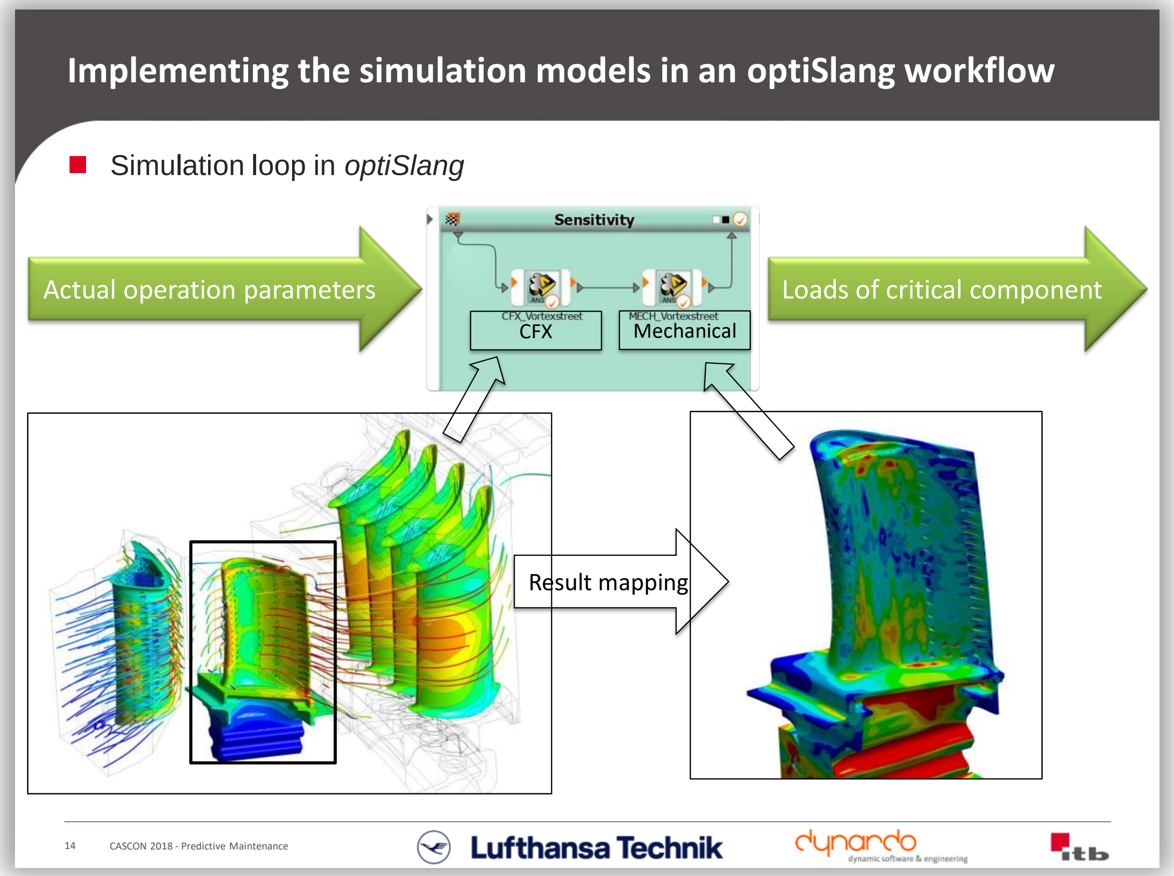
- Use flight data to increase maintenance intervals of airplane turbines and reduce cost
- Big data: Not enough data to predict failure events reliable
- Alternative: Combine simulation based Digital Twin with online data



[1] M.Eng. Holger Schulze Spüntrup (ITB Ingenieurgesellschaft für technische Berechnungen mbH), **Real-time processing with 3D meta models for predictive maintenance of aircraft engines,** *CASCON 2018*

Customer example: ROM for Predictive Maintenance Digital Twin

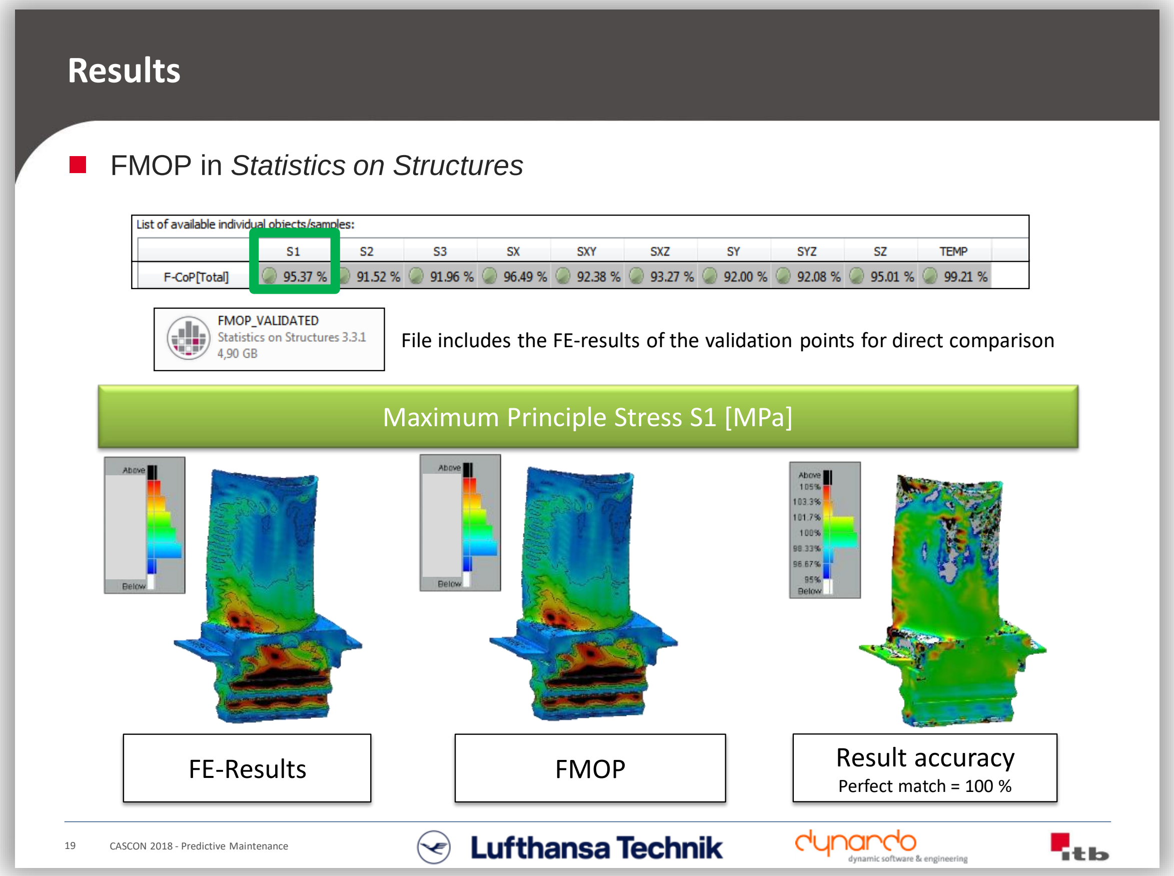
- Steps:
  - Simulation model with ANSYS CFX and Mechanical
  - optiSLang workflow for variation analysis of virtual sensors (Design of Experiments)
- Goal:
  - Search for a nonlinear 3D ROM
  - Objective: Approximate mechanical stress gradients based on transient flight data



[1] M.Eng. Holger Schulze Spüntrup (ITB Ingenieurgesellschaft für technische Berechnungen mbH), **Real-time processing with 3D meta models for predictive maintenance of aircraft engines,** *CASCON 2018*

Customer example: ROM for Predictive Maintenance Digital Twin

- Create Field-MOP for 3D stress field and temperature
- Predict prognosis quality by F-CoP
- Validate F-CoP by additional designs
- Consume Field MOP using DLL in Matlab



[1] M.Eng. Holger Schulze Spüntrup (ITB Ingenieurgesellschaft für technische Berechnungen mbH), **Real-time processing with 3D meta models for predictive maintenance of aircraft engines,** *CASCON 2018*

What is happening here ?

- Use virtual sensors as parameters to a nonlinear multi-physical simulation model
- Design of Experiments varying the parameter values systematically
- Process chain in optiSLang producing the result data
- Export data from ANSYS Mechanical for Field MOP generation
- Model understanding and validation by
  - Statistical analysis,
  - Sensitivity analysis,
  - Variation pattern analysis and
  - Estimation of prognosis quality

Prognosis quality: Field Coefficient of Prognosis (F-CoP)

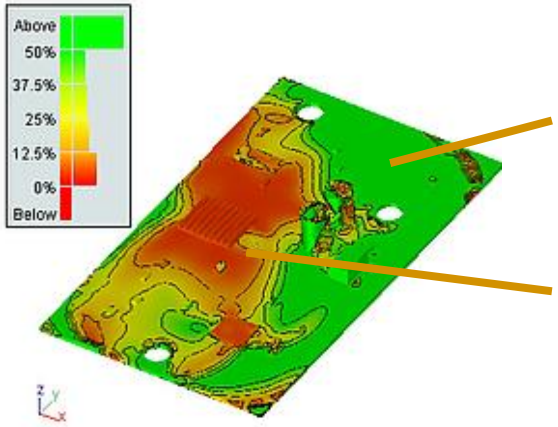
- Check prognosis quality in Field CoP matrix:
  - Single value (“easy to use”)
  - Indicates high or low model accuracy at a glance
  - Is an average value of the CoP in space

	LOGSEQV	TEMP	UX	UY	UZ
F-CoP[CTE_CPU]	7.75 %	2.29 %	1.75 %	1.67 %	1.95 %
F-CoP[CTE_PCB_xy]	18.73 %		15.92 %	14.99 %	4.01 %
F-CoP[TEMP_CPU]	31.42 %	31.63 %	53.71 %	43.62 %	87.40 %
F-CoP[TEMP_MCU]	12.91 %	5.52 %	4.18 %	2.84 %	1.59 %
F-CoP[TEMP_env]	55.24 %	61.74 %	28.39 %	40.77 %	5.09 %
F-CoP[Total]	84.42 %	94.52 %	94.86 %	94.84 %	94.99 %

Sensitivity to inputs

Prognosis quality

- Check prognosis quality in Field-CoP 3D plot:
  - Plots prognosis quality for each position
  - Compare with standard deviation as an indicator of the magnitude of variation
- Accept / Repair model / Add designs ?



High prognosis quality

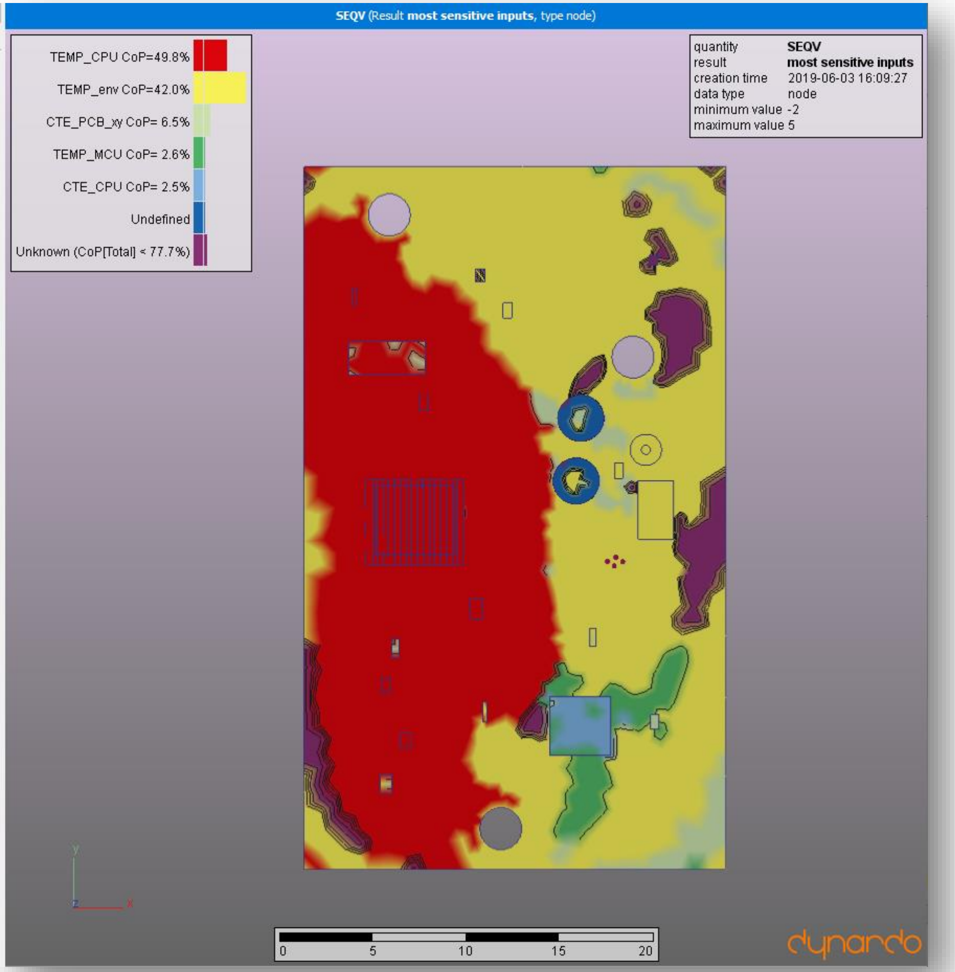
Low prognosis quality

Ansys

Sensitivity analysis: See all in a single plot

- Which parameter has the largest influence at what location ?
- Where has the ROM a too low accuracy ?

- Further post processing:
  - Use statistical measures to understand the statistics of variations at each position (e.g. mean value, standard deviation, quantiles….)
  - Plot variation patterns to identify correlations in space

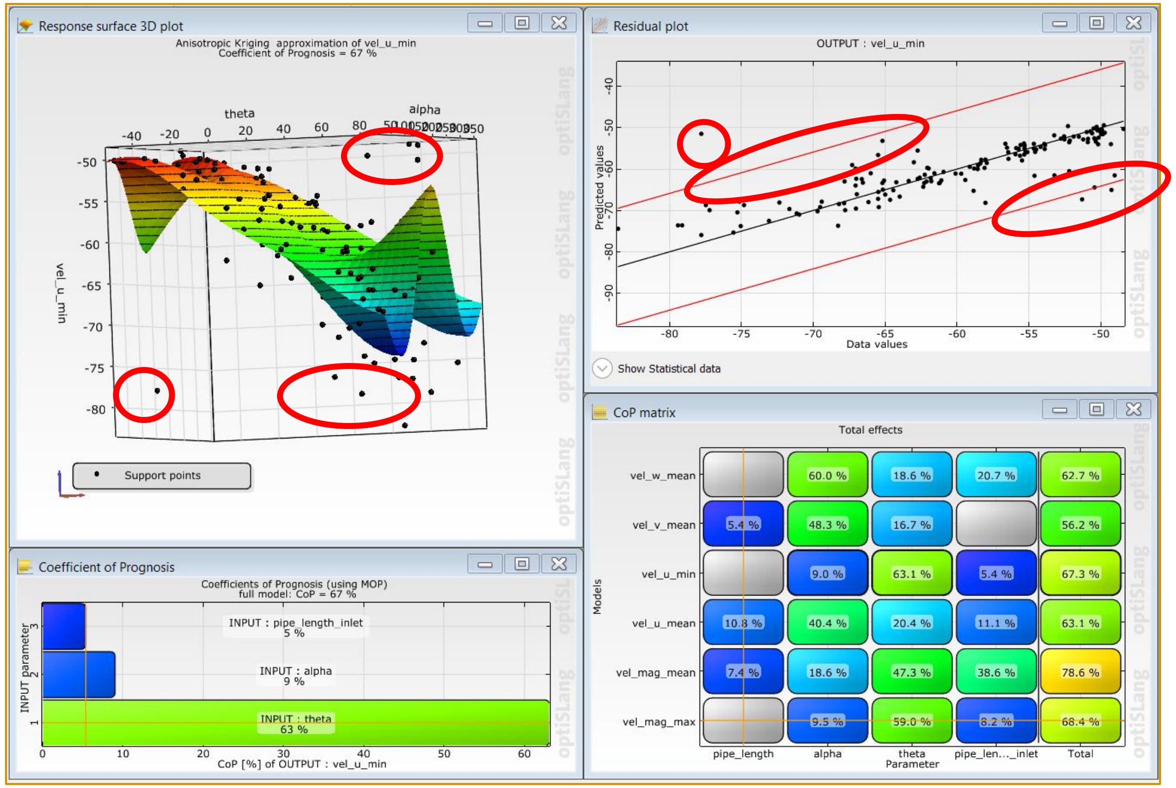


Innovations in optiSLang SoS 8 (2020R2): Detailed postprocessing

Open nonlinear interpolation functions used in Field MOP directly in optiSLang post processing

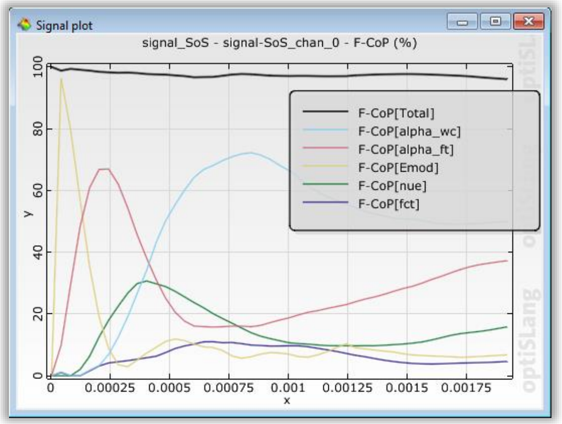
- Plot interpolation functions
- Easily identify statistical outliers in CV plot (highlighted in red)

Compute and visualize absolute differences between original and approximated field designs

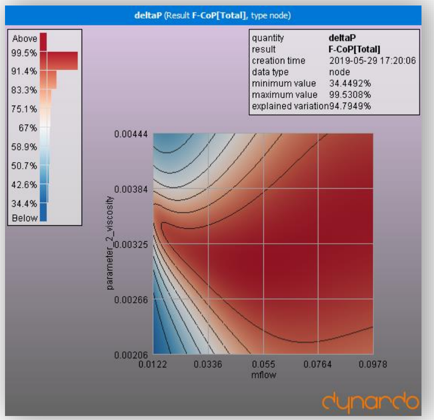


Types of Field Meta Models

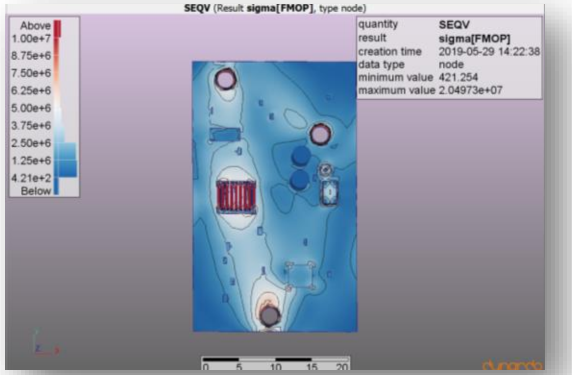
- 1D (Signal MOP / curves)



- 2D (matrices / performance maps)



- 3D (FEM meshes, CFD grids)



How to deal with measurements ?

How to deal with measurements in FEM, e.g. geometries ?

Validation

Apply measurement

- Create CAD0 geometry and mesh
- Determine geometric deviations to measurement (STL)
- Morph mesh and compute structural performance

Uncertainty quantification

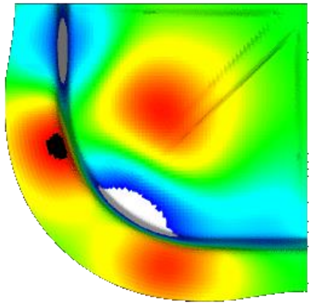
Robustness and Reliability Analysis

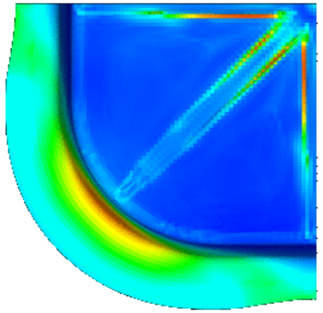
- Create CAD0 geometry and mesh
- Determine geometric deviations to several measurements (STL)
- Create a statistical shape model
- Generate artifical geometries for DOE and use morphed FEM meshes in CAE

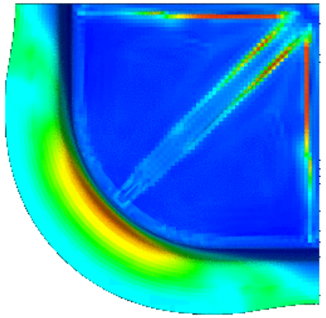
CAD0 → Measurements (STL) → Morph CAD0 mesh → Run FEM

The key: Random fields depending on available data

1. **No/single measurement**:
   assumptions
   (synthetic random field model)
2. **3-5 measurements**:
   empirical mean+stddev
   assumed correlation
   (synthetic random field model)
3. **Many measurements**:
   Empirical random field model
   Anisotropic, inhomogeneous,
   Non-Gaussian



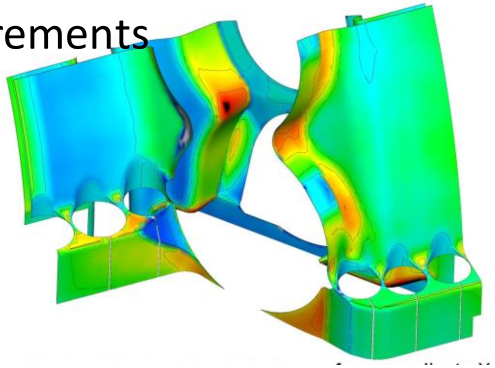


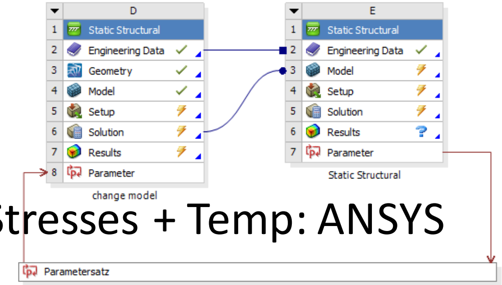


Customer story: Few geometric measurements
How do manufacturing tolerances affect low-cycle fatigue ?

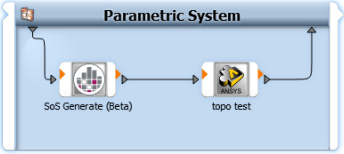
- Casting process (here: gas turbine housings)
- Question: How do geometric imperfections in production influence stress / fatigue behavior ?

Mean imperfection from measurements



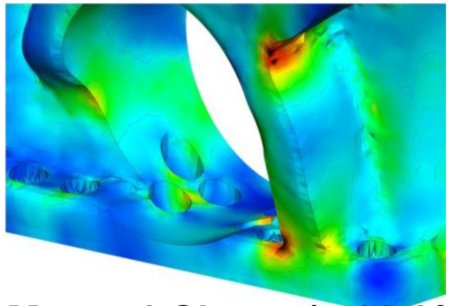


Stresses + Temp: ANSYS

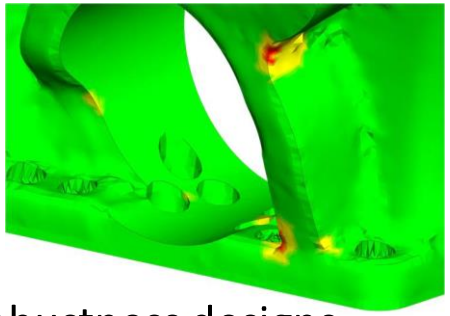


optiSLang: 100 robustness designs

Mean stress



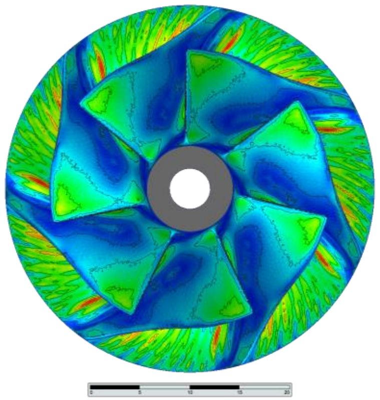
Mean+1·Sigma (~ 68,3%)



With courtesy of SIEMENS (source: Lohse et al, DVM Probabilistic Workshop 2016)

Customer story: Many measurements available
Which production process is more reliable ?

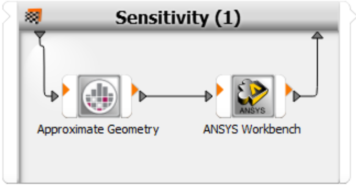
- GOM Measurements produced compressor wheels of different production processes
- Quality tolerances are all met, but different behavior in fatigue failure

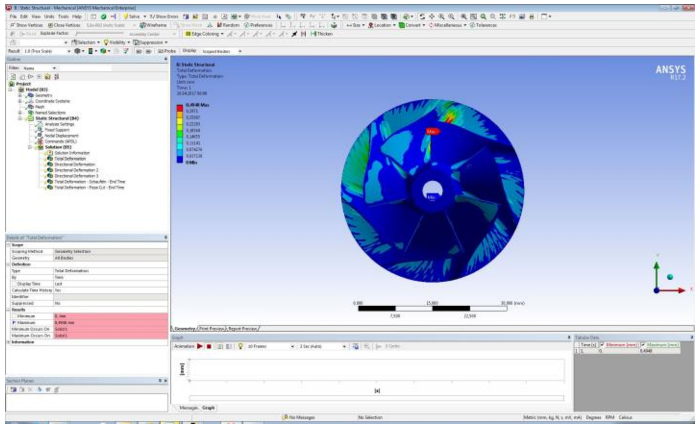


Mean of geometry

How to approach ?

- Import data and analyse statistical properties of produced geometries
- Setup ANSYS Mechanical model
- Analyse influence of geometry imperfections onto lifetime using an automated robustness analysis

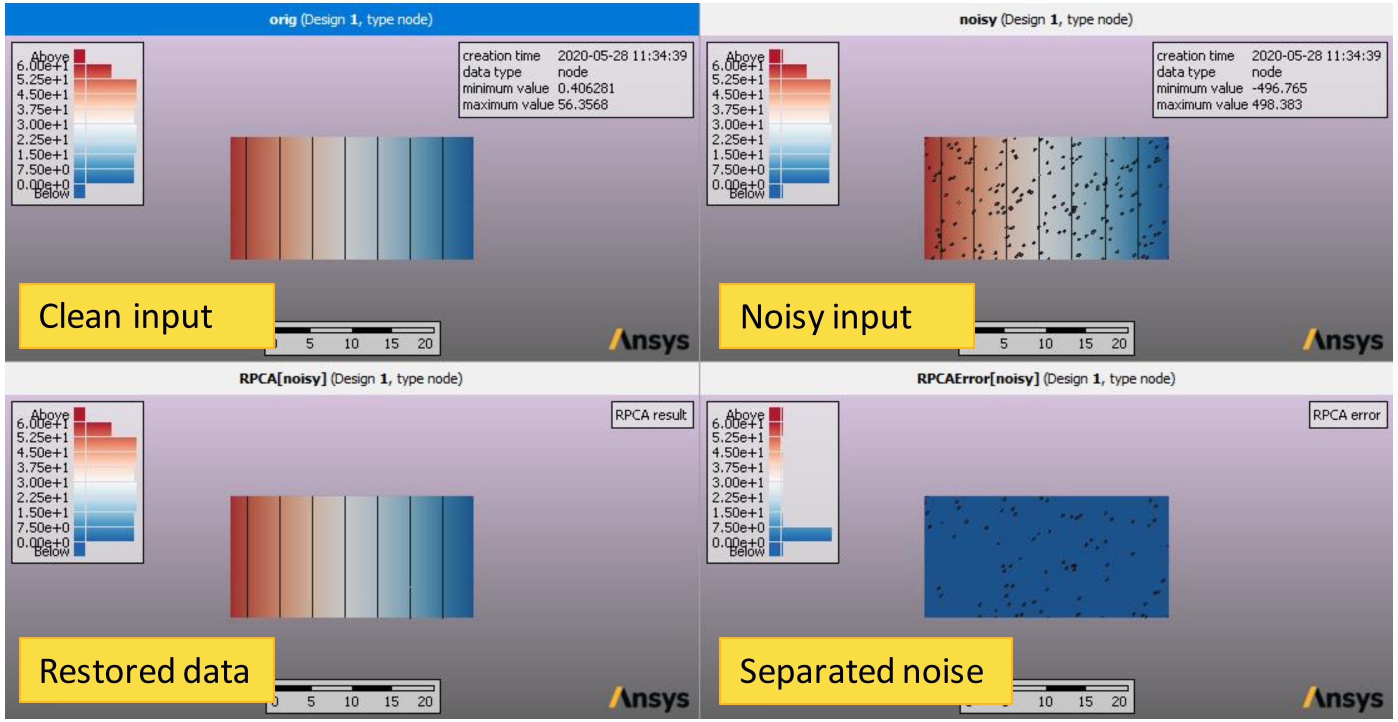




New in optiSLang SoS 8 (2020R2): Filtering noisy training data

- Filter noise from measurements based on statistical filtering
- Restore original data by comparing typical correlations among all other measurements

- To be applied to:
  - Geometric measurements of production tolerances (laser scans)
  - Noisy signal data



New in SoS 8 (2020R2): Improved analysis of measurements: Easier identification of potentially “false” measurements

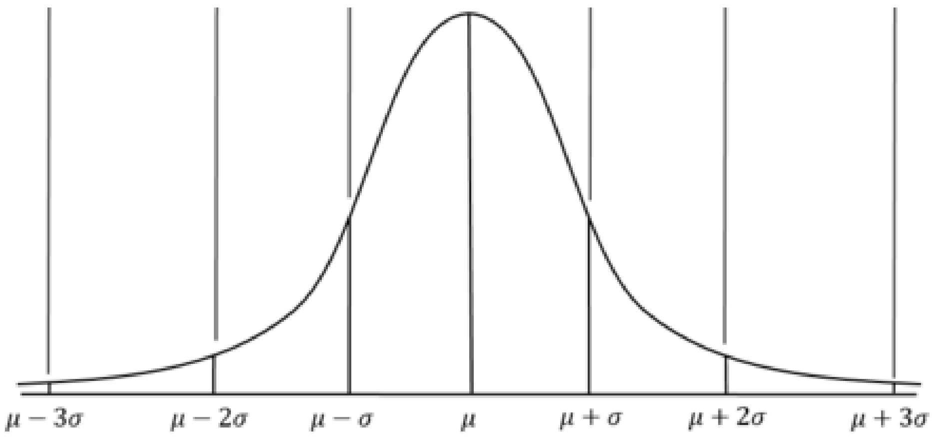
Check if the spatial pattern is broken :

- Compute and compare error norms for each measurement when using a random field, identify potential outliers
- Plot the error when representing a measurement by a random field

	err_max[pstrain]	err_min[pstrain]	err_rms[pstrain]
35	0.0232362	-0.0290202	0.0033847
75	0.0314974	-0.0343026	0.00259706
26	0.0273092	-0.0312857	0.00247323
61	0.0254184	-0.0195148	0.00244799
8	0.0233878	-0.0251521	0.00243145

Easily identify histogram tails and measurement outliers by :

- Translating field data into scalars
- Measure the distance from mean
- Easily find measurements being too far from mean



Import data to Field MOP

Import data from text files

- Field MOP training needs:
  1. Reference mesh for connectivity and visualization
  2. Training data (node or element data saved on the mesh)

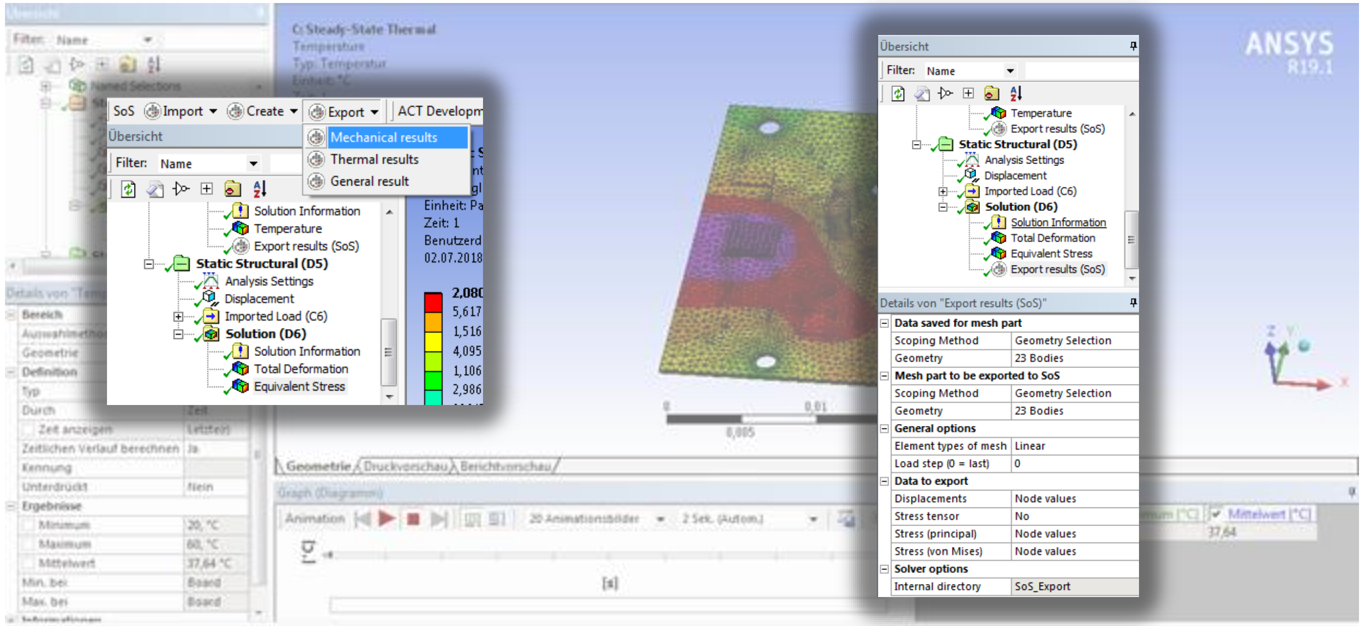
- Import data
  - From mesh ASCII files: ANSYS CDB, LS-DYNA Dynain, Abaqus INP, Nastran BDF, STL, VTK…
  - Element/Node data: CSV, LS-PrePost, …
  - Converter scripts for unsupported file formats

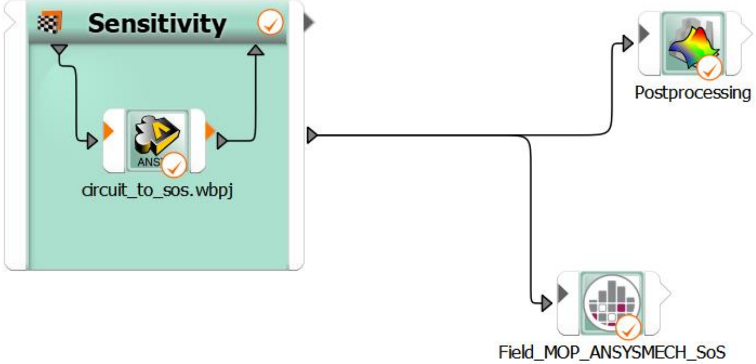
- Mesh mapping
  - Determine coordinate deviations between CAD0 and final geometries
  - Map data between incompatible meshes

Ansys

ANSYS Mechanical Plugin

- Exports result data and mesh from ANSYS Mechanical to SoS
- Apply geometric imperfections (Uncertainty Quantification and Reverse Engineering) inside Mechanical
- optiSLang integration node for full-automated analysis





Ansys

Export data and process automation

1. “Brute force”: Direct export and consumption of data

- Export any data from Field MOP database as CSV to Excel, optiSLang etc.
- Connect Field MOP consumpton with 3rd party software through shared libraries:
  - Solve Field MOP and retrieve complete data vectors (3D fields, signals, etc.)
  - Access mesh connectivity
  - Use embedded scripting for full SoS capability including Field Mop creation and I/O
  - ANSI C API and examples for Matlab, C++, Python …

```c

DYNARDO_FMOP_API fmop_error_t FMOP_getModelIdents
     ( const fmop_db_handle_t database, fmop_dataobject_types data_ty
DYNARDO_FMOP_API fmop_error_t FMOP_getModelParamIdents
     ( const fmop_handle_t fmop, char *** const param_idents, size_t
DYNARDO_FMOP_API fmop_error_t FMOP_getParamLowerBounds ( const fmop_
DYNARDO_FMOP_API fmop_error_t FMOP_getParamUpperBounds ( const fmop_
DYNARDO_FMOP_API fmop_error_t FMOP_getModelTotalAvgFCoP ( const fmop
DYNARDO_FMOP_API fmop_error_t FMOP_getModelAvgFCoP ( const fmop_hand
DYNARDO_FMOP_API fmop_error_t FMOP_getModelDim ( const fmop_handle_t
DYNARDO_FMOP_API fmop_error_t FMOP_getDataPointIndizes ( fmop_handle
DYNARDO_FMOP_API fmop_error_t FMOP_getDataPointCoors (fmop_handle_t


/*****************************************/

DYNARDO_FMOP_API fmop_error_t FMOP_approxField
     ( const fmop_handle_t fmop, const double * param_values, double

```

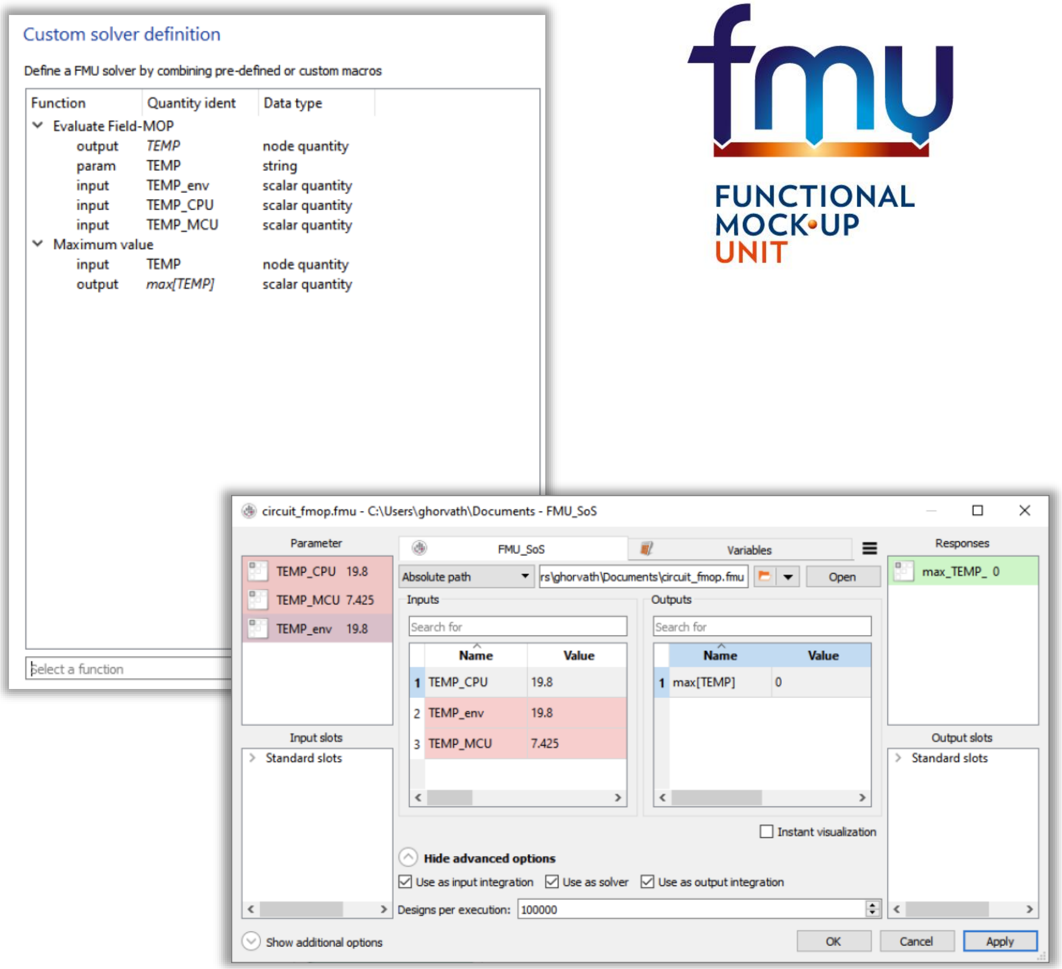
ANSI C/C++

```python
param_idents = ctypes.POINTER( ctypes.c_char_p )()
num_idents = ctypes.c_ulonglong(0)
sos.FMOP_getModelParamIdents ( fmop, ctypes.byref( param_idents ), ctypes.

num_mesh_items = ctypes.c_ulonglong(0)
sos.FMOP_getModelDim ( fmop, ctypes.byref ( num_mesh_items ) )

param_values = ( ctypes.c_double * num_idents ) ( 1., 2., 3., 4., 5., 6. )
approx_field = ( ctypes.c_double * num_mesh_items.value ) ()
sos.FMOP_approxField ( fmop, param_values, ctypes.byref( approx_field ) )
```

Python example

2. Innovation: Export FMU 2.0 (Functional Mockup Unit)

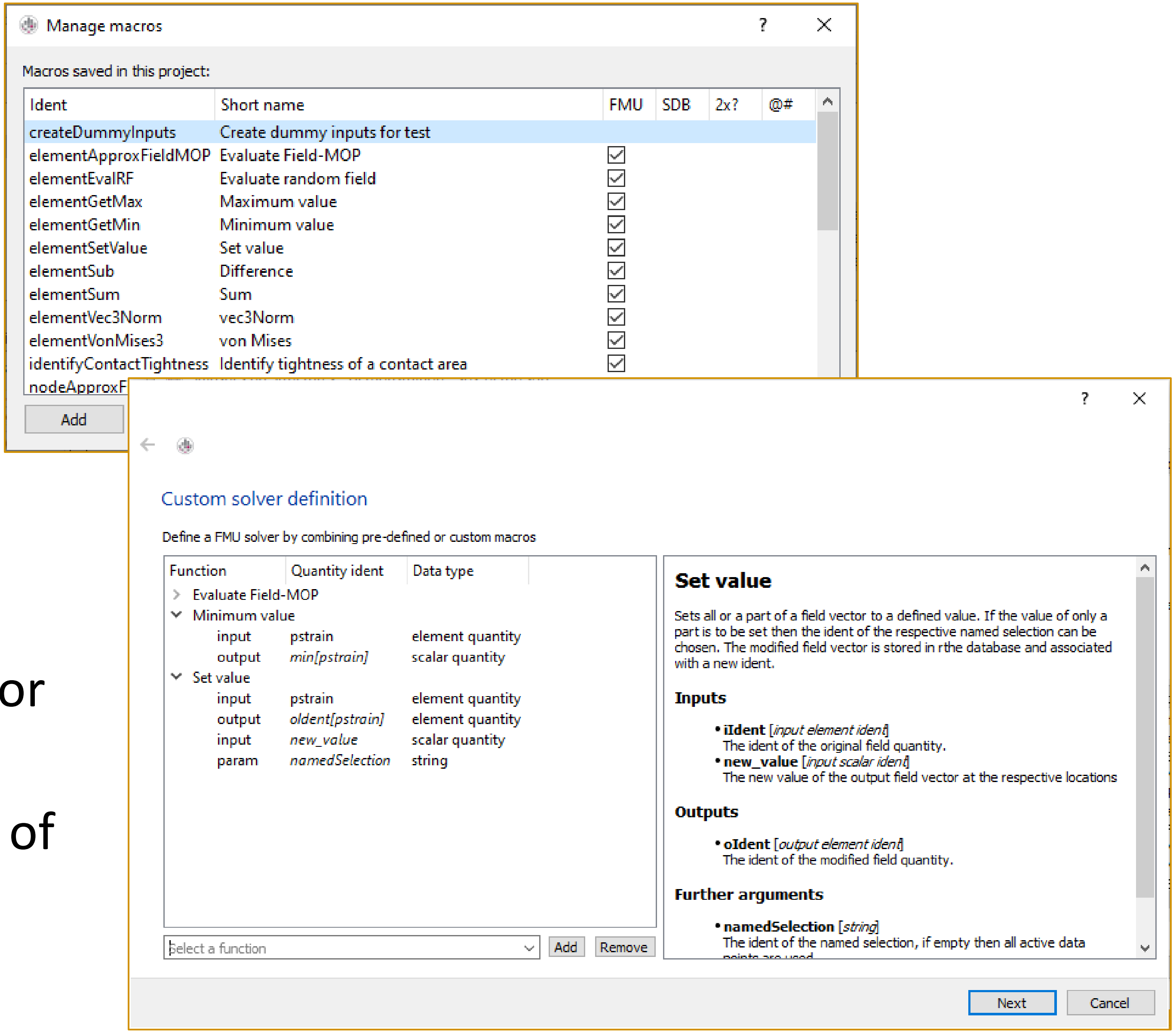
- User can write his own analysis macros
- Combine macros into a single automated analysis
- Export workflows to FMU 2.0 (model exchange)
- Consume FMU in optiSLang or TB
- Visualize all 3D fields afterwards in SoS post prociessing



2. Innovation: User macros

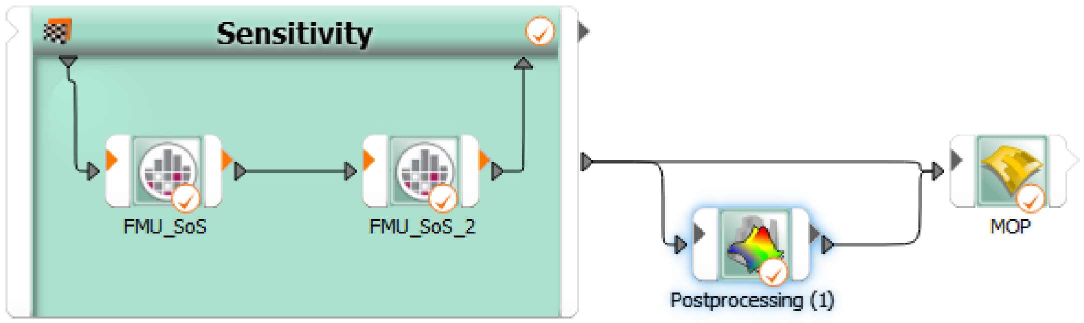
Macros may include

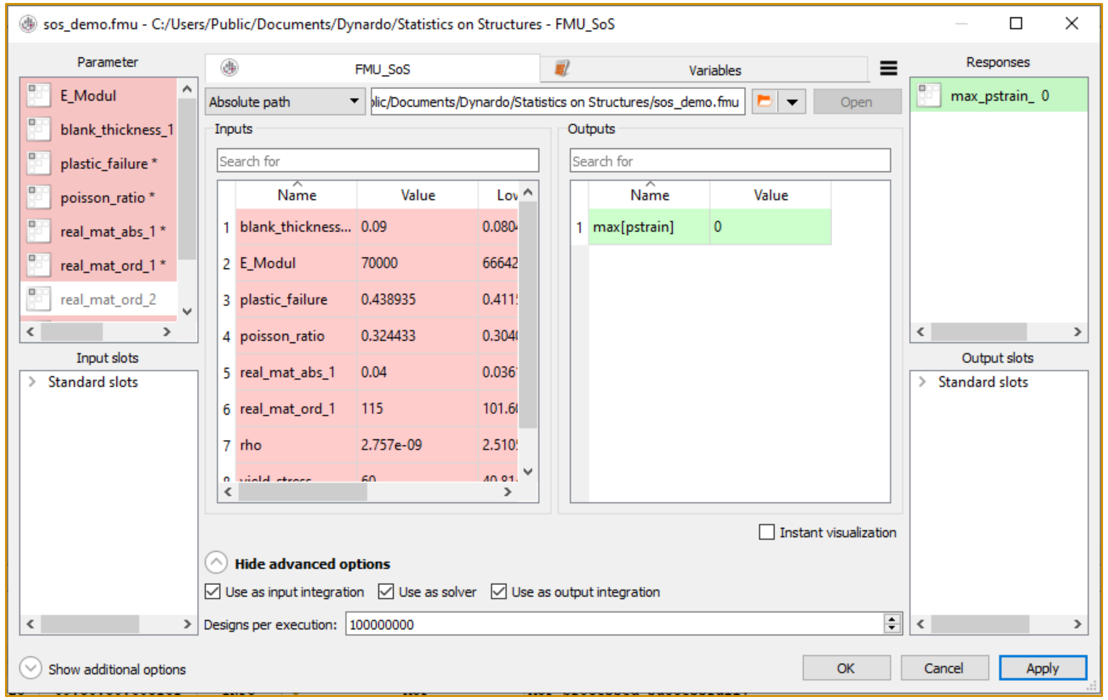
- Simple analysis macros (e.g. extract maximum along an edge)
- Post processors (e.g. Log, Exp, von Mises stress from tensor, vector norms)
- Statistical analysis over all designs (for Robustness, Reliability or Fatigue)
- Complex analysis (e.g. identification of tightness of contact areas in high pressure valves; see presentation of Tamasi et al)



2. Innovation: Consumption of FMUs in optiSLang workflows

- Use Field MOP FMU for simulation in optiSLang
- FMU solver node (Beta option)
  - Autoregister inputs and responses
  - Runs in optimized mode
  - Visualize all 3D fields afterwards in SoS post prociessing
  - Entirely implemented using optiSLang’s powerful customization features (Python 3)





Ansys

3. Process automation for field data in optiSLang Innovation: Improved process nodes (**output** fields)

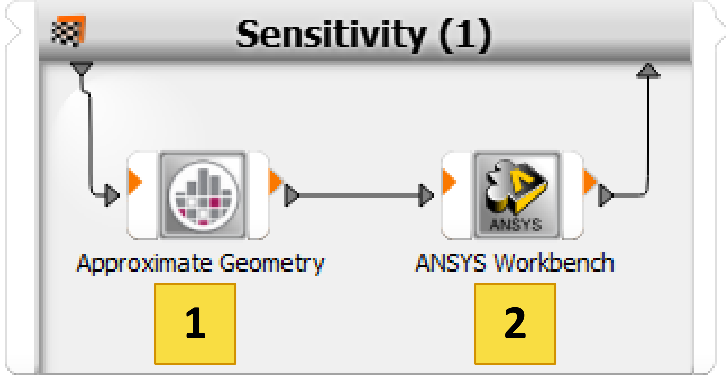
How does it work ?

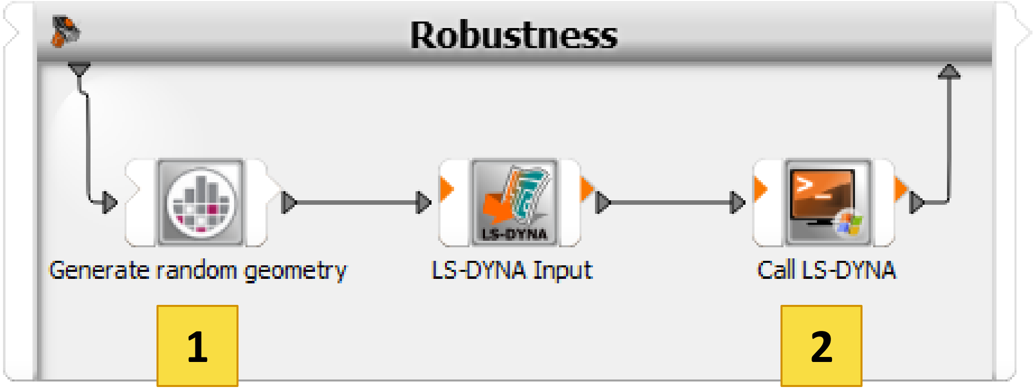
- User prepares CAE solver
- User prepares SoS model for export (to CSV ? To mesh file ? To ANSYS Mechanical, LS-DYNA, Abaqus, Nastran?)

For each design:

1 optiSLang calls SoS to modify the CAE input deck based on scalar parameters

2 optiSLang calls CAE to run with modified mesh





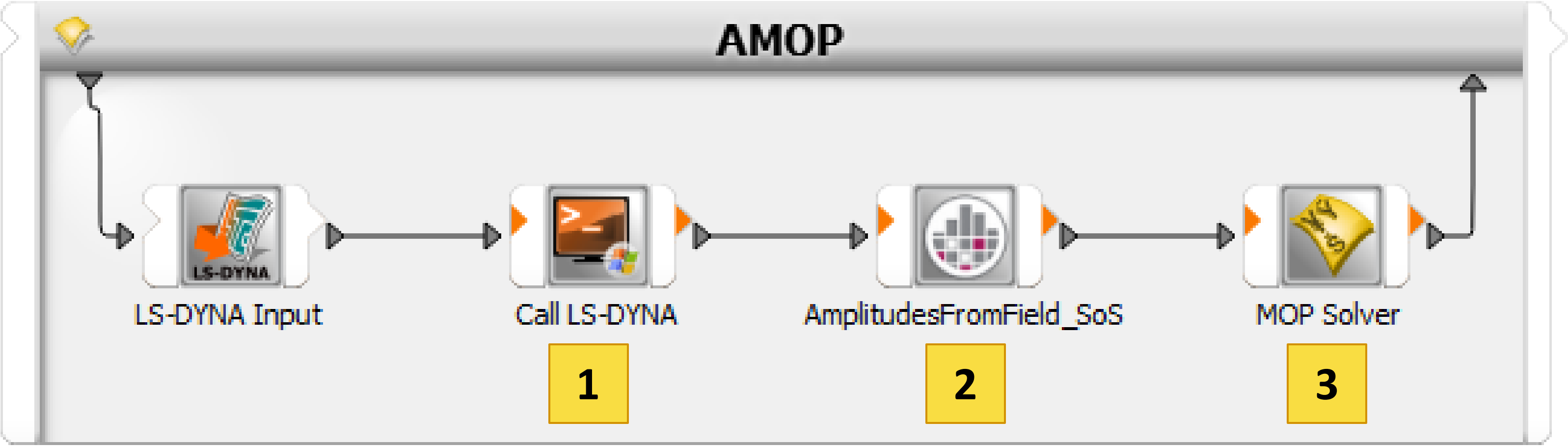
3. Process automation for field data in optiSLang
Innovation: Improved process nodes (**input** fields)

How does it work ?

- User prepares CAE solver or measurement that produces field output (e.g. a modified FEM mesh, a STL 3D measurement, a signal)
- User prepares SoS model that imports the file and projects the field data into scalar "parameters"

For each design:

1. optiSLang calls the CAE solver
2. optiSLang calls SoS to read CAE result and gets the scalar parameters
3. optiSLang uses the scalars, e.g. in (Field)MOP, as inputs to CAE solvers or in optimization goals



Summary: optiSLang SoS 8 (2020R2)

Powerful analysis tools for model understanding and approximation

Improves and simplifies data analysis for beginners and experts

Improves workflow automation by field-in and field-out nodes and FMU export

Ansys