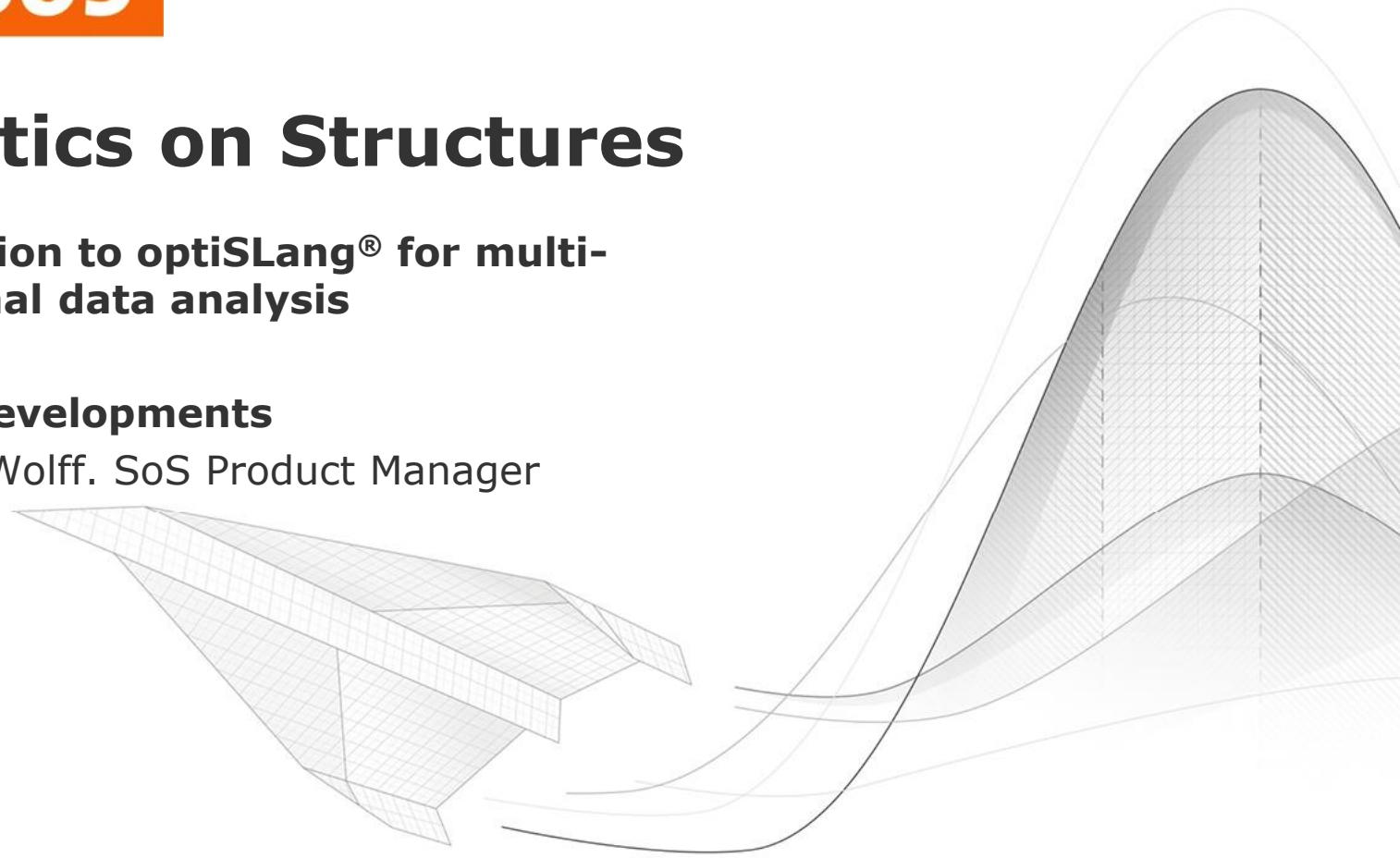
Statistics on Structures

An extension to optiSLang® for multi-dimensional data analysis

News & Developments

Sebastian Wolff. SoS Product Manager



Summary
Current version: SoS 7

- New features
  - ANSYS Mechanical SoS plugin
  - Interactive 3D/ROM Viewer
  - Free-form shape optimization
  - Binary interface, ANSI C (FMOPSolver.DLL)
- Improvements
  - New FMOP Backend, improved SignalMOP
  - Mesh smoothening, free-form models, named selections
  - File formats
- In Development
  - SoS Script DLL

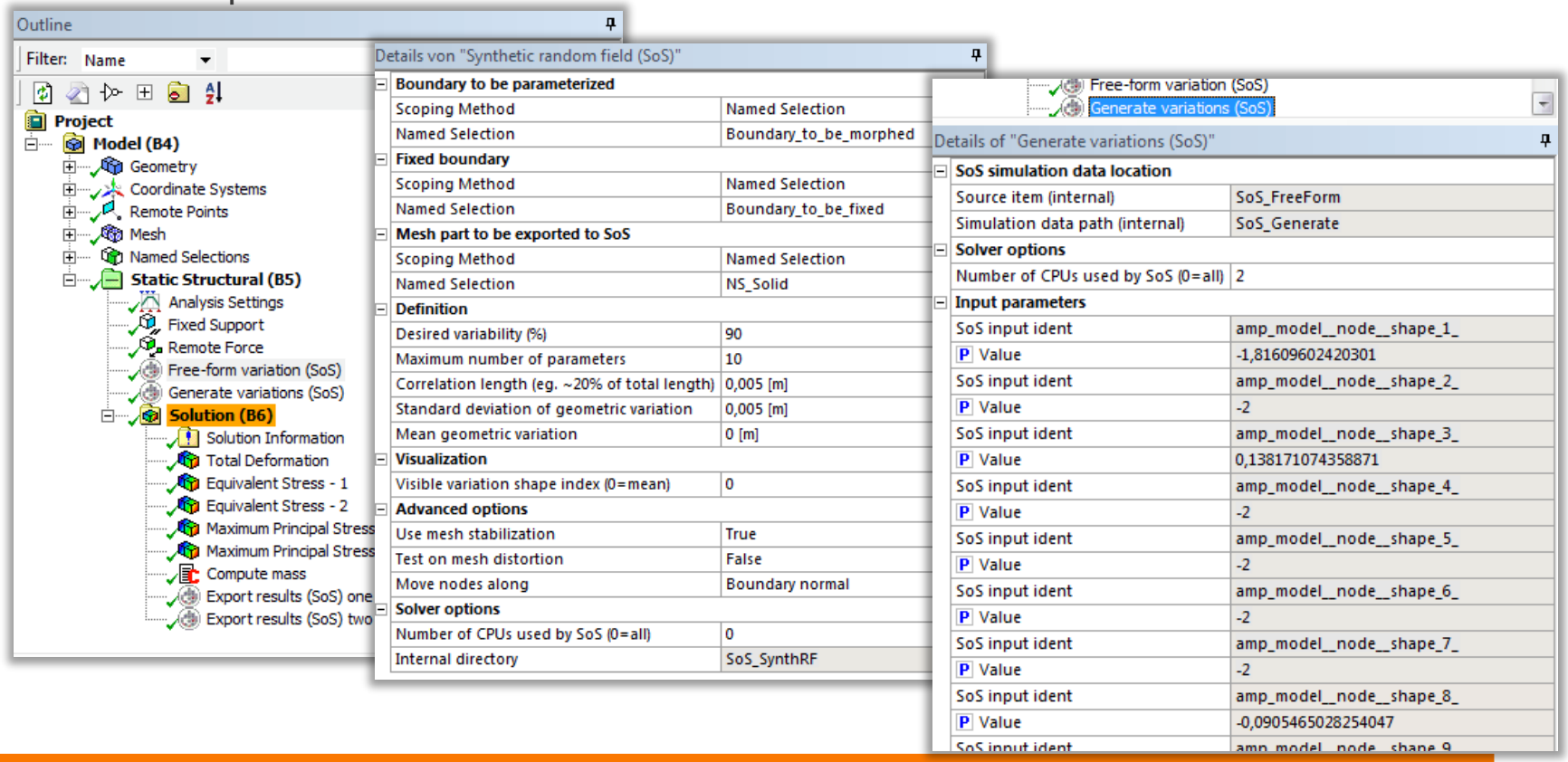
New features:
ANSYS Mechanical SoS Plugin

- Objectives:
  - Create random fields for geometric imperfections
  - Create free-form variation models for geometric imperfections
  - Import external random fields into ANSYS (e.g. obtained from measurements)
  - Expose random field parameters to ANSYS WB parameter set
  - Export result data to SoS

- Toolbar

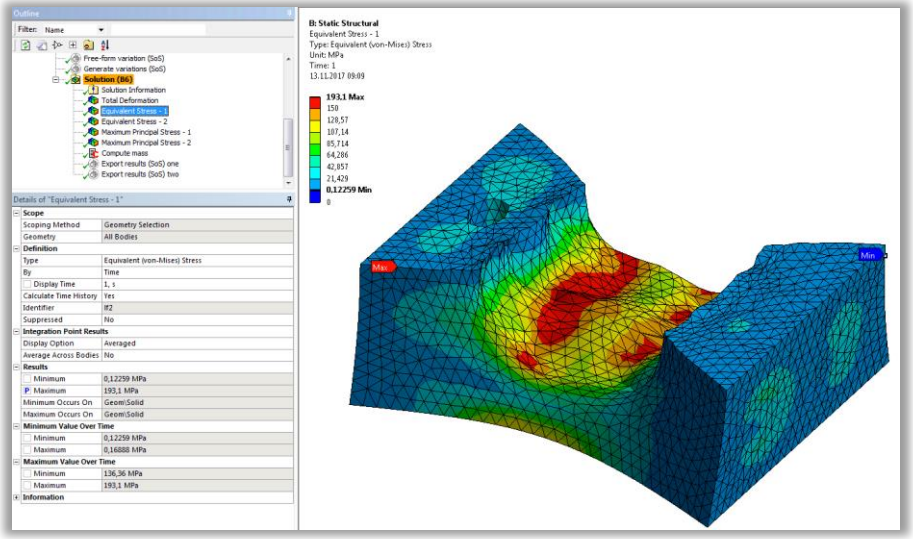
New features:
ANSYS Mechanical SoS Plugin

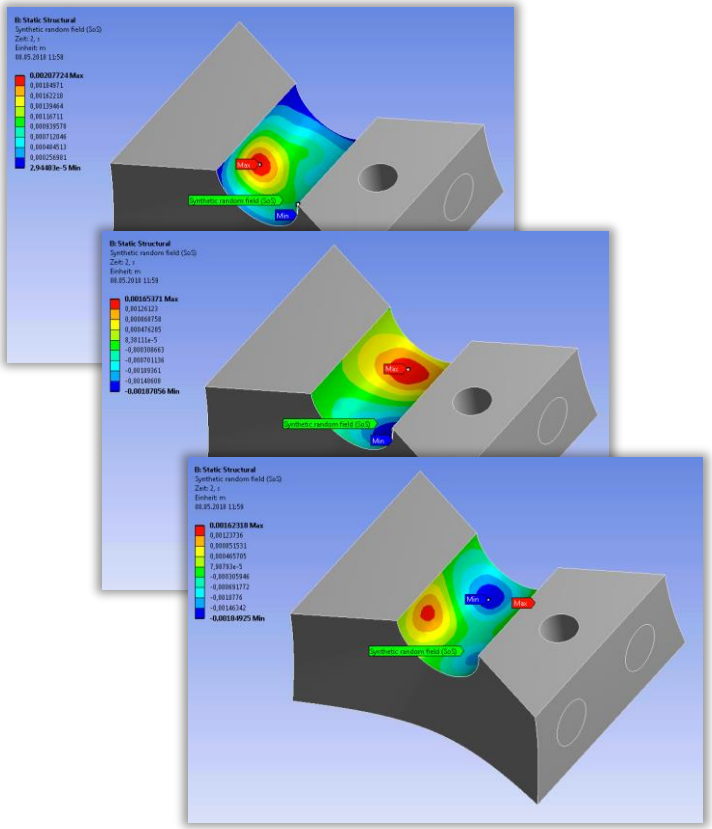
- Create geometric variations directly in ANSYS Mechanical
- Here: Preparation



New features:
ANSYS Mechanical SoS Plugin

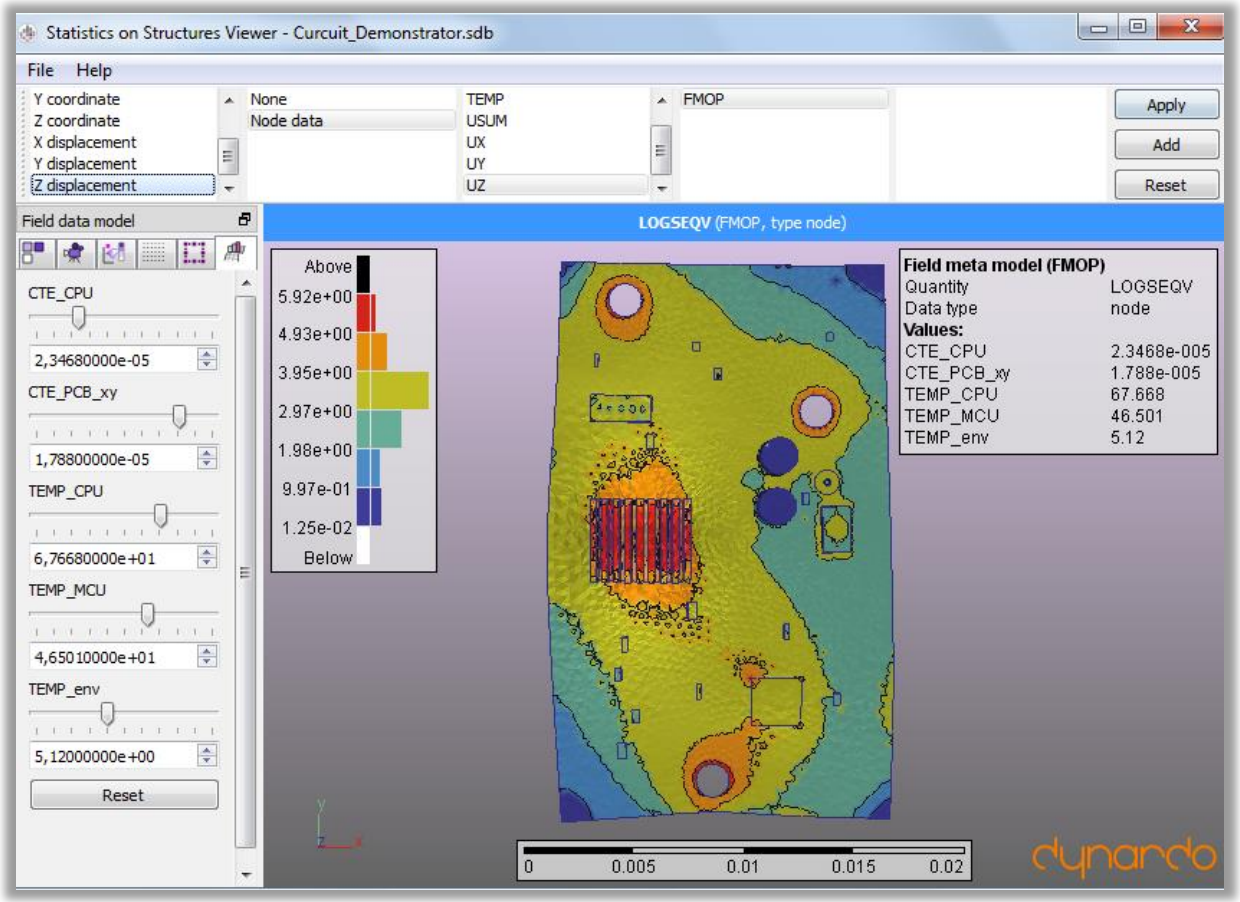
- Visualize scatter shapes, visualize applied variation as contour plot or deformation in ANSYS Mechanical





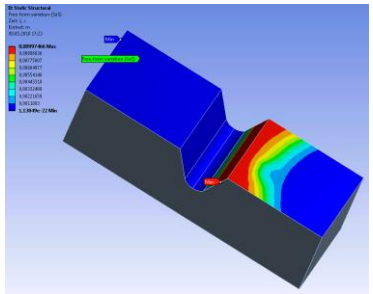
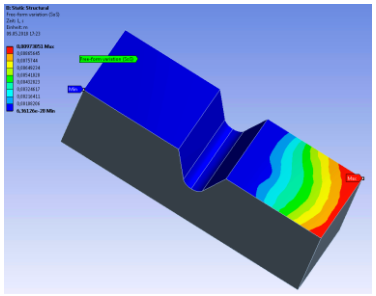
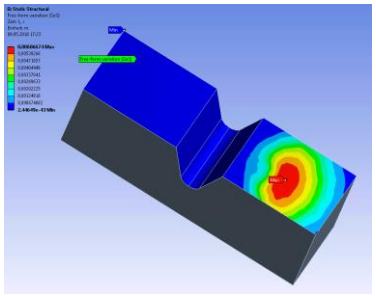
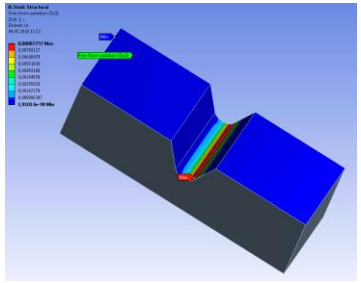
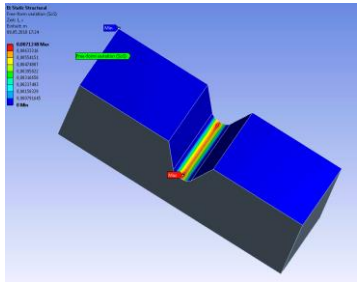
New features:
Interactive 3D/ROM Viewer

- **Standalone SoS Viewer** for FMOP, Random Fields and statistical analysis (read-only mode)
- **Interactive visualization**: Use *sliders* to change parameters of FMOP and RF
- Visualize data as colored **contour plots** and **deformed geometry**



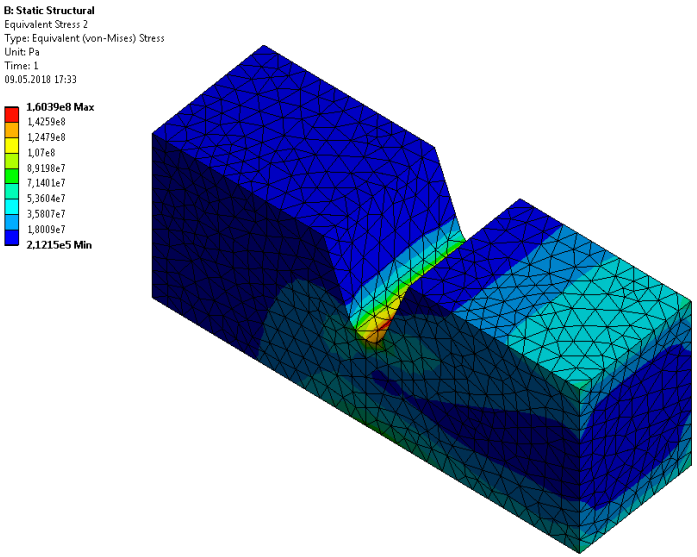
New features:
Free-form shape variations

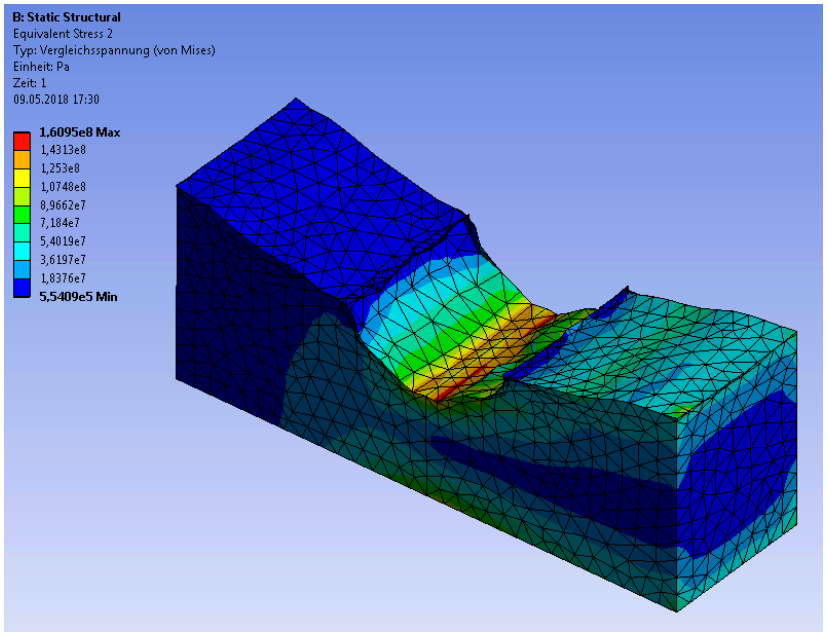
- New random field model type: Free-Form variation models
- What is new ?
  - Classical random fields: Global shapes (“sine”-”cosine”-alike)
  - Free-Form fields: Local shapes using support points
  - Support points can be placed automatically or manually or both

New features:
Free-form shape variations

- Successful mesh morphing even for complex geometries…





New features:
FMOPSolver.DLL

- FMOPSolver.dll / .so for Windows + Linux
- ANSI C API for
  - Dynamic linking with C/C++ software
  - Bindings to Python
  - Bindings to Matlab
  - Etc.
- Allows approximation of
  - signals and
  - FEM solutions (node data / element data) through FMOP

ANSI C/C++

```
305
326 DYNARDO_FMOP_API fmop_error_t FMOP_getModelIdents
327     ( const fmop_db_handle_t database, fmop_dataobjec
349 DYNARDO_FMOP_API fmop_error_t FMOP_getModelParamIdent
350     ( const fmop_handle_t fmop, char *** const param_
370 DYNARDO_FMOP_API fmop_error_t FMOP_getParamLowerBound
390 DYNARDO_FMOP_API fmop_error_t FMOP_getParamUpperBound
407 DYNARDO_FMOP_API fmop_error_t FMOP_getModelTotalAvgF
426 DYNARDO_FMOP_API fmop_error_t FMOP_getModelAvgFCoP (
443 DYNARDO_FMOP_API fmop_error_t FMOP_getModelDim ( cons
469 DYNARDO_FMOP_API fmop_error_t FMOP_getDataPointIndize
499 DYNARDO_FMOP_API fmop_error_t FMOP_getDataPointCoors
500
502
503 /*********************************************/
509
535 DYNARDO_FMOP_API fmop_error_t FMOP_approxField
536     ( const fmop_handle_t fmop, const double * param_
537
```

```
1  param_idents = ctypes.POINTER( ctypes.c_char_p )()
2  num_idents = ctypes.c_ulonglong(0)
3  sos.FMOP_getModelParamIdents ( fmop, ctypes.byref( param_idents ), ctypes.byref ( num_idents ) )
4
5  num_mesh_items = ctypes.c_ulonglong(0)
6  sos.FMOP_getModelDim ( fmop, ctypes.byref ( num_mesh_items ) )
7
8  param_values = ( ctypes.c_double * num_idents ) ( 1., 2., 3., 4., 5., 6. )
9  approx_field = ( ctypes.c_double * num_mesh_items.value ) ()
10 sos.FMOP_approxField ( fmop, param_values, ctypes.byref( approx_field ) )
```

Python example

In development:
SoS script DLL

- Under development
  - Access all functions of SoS using the SoS script language
  - Start SoS scripts (internally as strings)
  - Access data through binary access
- ANSI C API (also usable for embedding into Python, Matlab etc.)

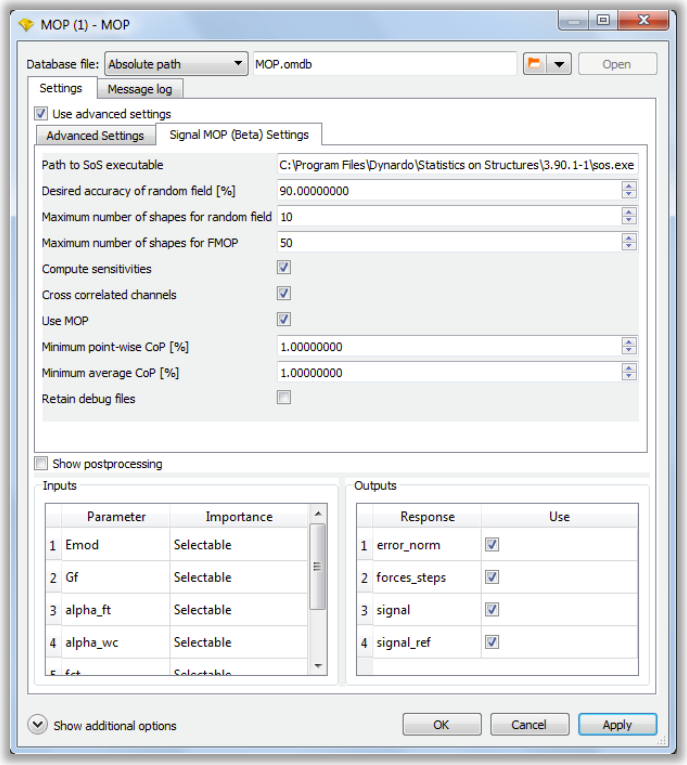
Improvements:
New FMOP backend

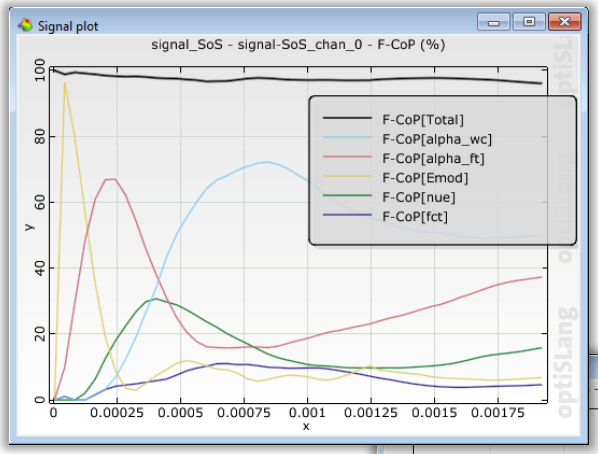
- New meta modelling technologies in SoS
- Currently used to compute sensitivity indices and prognosis quality
  - New: Also exact for arbitrary nonlinear models
- Beta option (SoS script or SignalMOP):
  - Also used for approximation

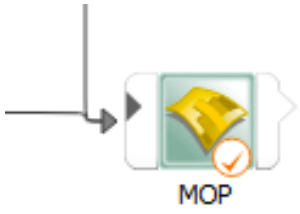
- **FMOP can nearly reproduce same results as point-wise MOP !**

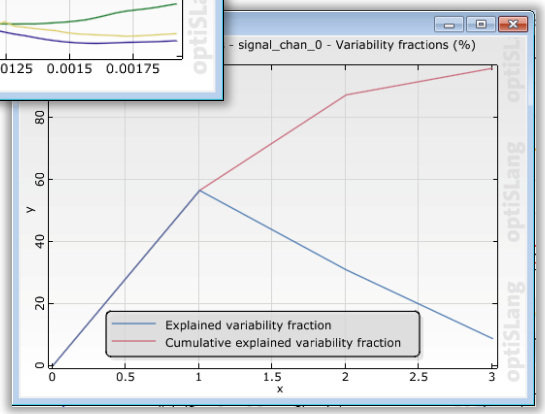
Improvements:
New SignalMOP (Beta option in optiSLang 7.1)

- SignalMOP integrated into MOP and MOP-Solver nodes (using SoS 3.3.4 + 7)
- New post processing options: F-CoP, Scatter shapes, quantile values, variability fractions, linear correlations



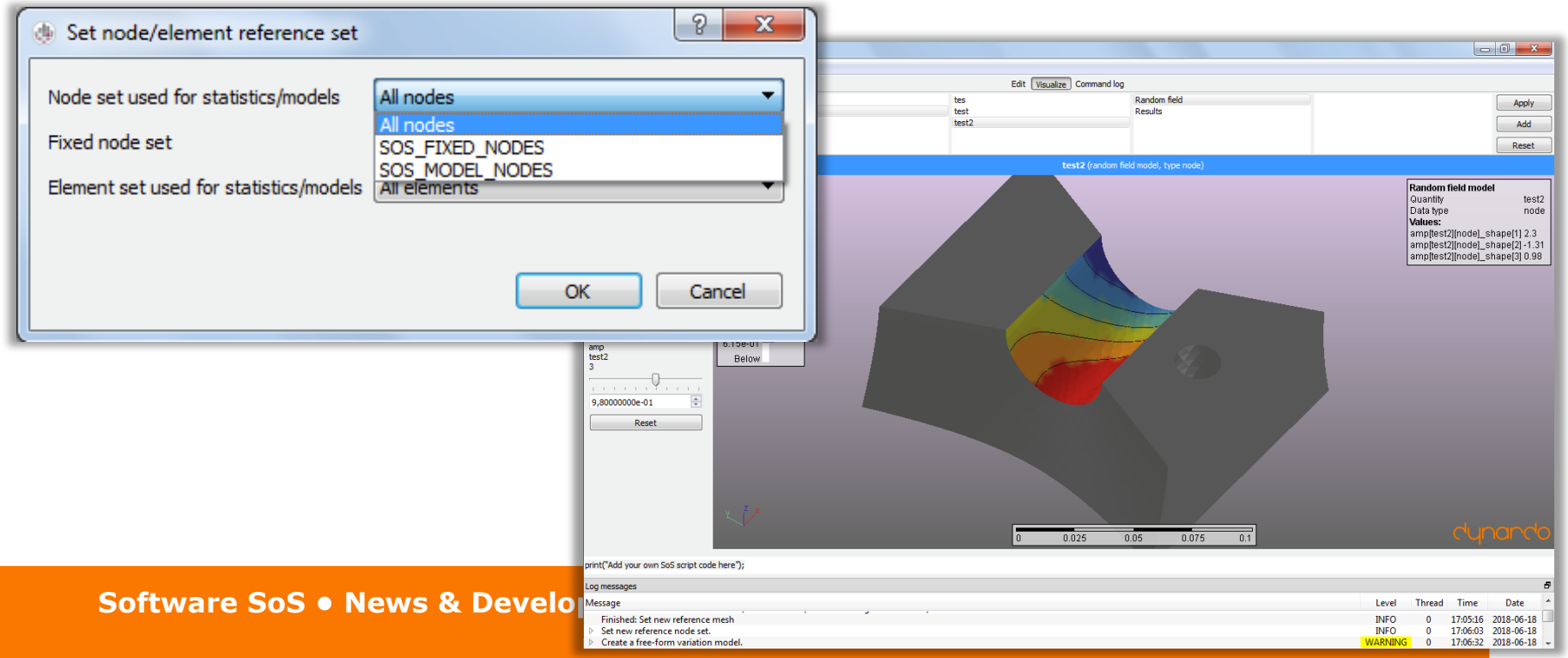






Improvements:
Support for Named Selections / Components

- Components = Named Selections, Node sets, Element sets
- Used for:
  - Import data only for a subset of your FEM mesh
  - Define fixed supports when modifying geometries

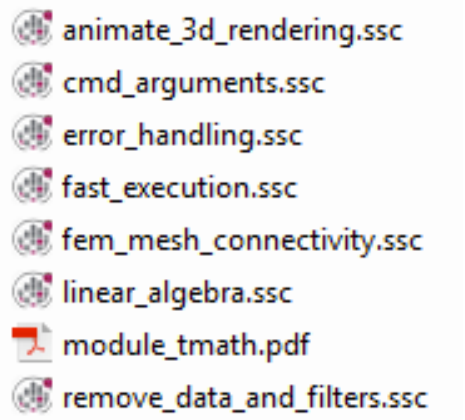


Improvements:
File formats and Import mappers

- Photographic measurements:
  - Import images of different resolution (resample/scale or cut-off)
- ANSYS:
  - Import named selections from CDB
  - Export data to External Data CSV format (as input to Mechanical)
- Abaqus:
  - Extend FEM element library (new: quadratic volume elements)

Improvements:
Advanced SoS script API

- Powerful and fast SoS script API for automated processing
- See examples (C:\Users\Public\Documents\Dynardo\...)
- Used within projects e.g. to
  - Do simple fatigue analysis based on FMOP stress results (implementing rainfall counting)
  - Create a database for worn brake pad measurements
  - Perform time stepping based on nonlinear ROM (through FMOP)
  - Much more….



Script examples in C:\Users\Public\Documents

dynardo

dynamic software & engineering

contact@dynardo.at
Dr. Sebastian Wolff