

INSIDE

TITLE STORY // RDO METHODOLOGY

OPTISLANG® FOR ANSYS®

A standard introduction of CAE-based Robust Design Optimization (RDO) in virtual product development places high demands on process automation, parametric models and algorithmic efficiency as well as operating reliability. Algorithm wizards and user guidance by best practice modular workflows make optiSLang version 4 an easy and flexible to use software tool for RDO projects.

State-of-the-art algorithms

For the last 10 year, optiSLang has been established as a multi-purpose toolbox for CAE-based RDO. During the initial development phase of the software, the focus was concentrated on improving the functionality and efficiency of the underlying algorithms necessary to conduct a sensitivity analysis, optimization or robustness evaluation. The goal was to make the user capable of dealing with large numbers of variables using any nonlinear CAE-solver. With the innovative technology of the Metamodel of Optimal Prognosis (MoP) and improved algorithms, challenging RDO tasks can now be efficiently solved.

Parametric modeling

However, it also became obvious that for a standard introduction of RDO, some more barriers have to be overcome. First of all, the availability and generation of suitable CAE parametric model is a key requirement. Here, the ANSYS Workbench has been established as one of the most powerful parametric modeling environments including bidirectional interfaces to major CAD programs and being capable of collecting all available CAE and CAD data in a central parameter manager.

Process integration and automation

Consequently, the system integration, process automation and job control, which represented further barriers, were also integrated into ANSYS Workbench to update one or multiple designs from the parameter manager. At the same time, users asked us to improve our optiSLang functionality of process integration and process automation. In this context, we decided to develop a direct integration of optiSLang into parametric modeling environments with the same priority as the integration of external CAE-codes into optiSLang's process automation functionality. To reach this goal, optiSLang was recoded from scratch and all optiSLang functionality was designed using C++ code with related interfaces which can be used by optiSLang's GUI the same way as inside third party modeling environments. The first result of this concept was the release of the new optiSLang version 4 inside ANSYS Workbench in 2011. In 2012, we released the stand-alone optiSLang GUI version 4 providing enhanced functionalities in process integration and automation. The new graphical user interface supports the file based process integration, the direct access to parametric modeling CAE environments like ANSYS or SimulationX as well as to programming environments like EXCEL, MATLAB or Python.

For ANSYS Workbench users, the decision to use the integrated optiSLang version or to integrate ANSYS workbench into optiSLang is driven by the availability of all necessary input and output parameters in the ANSYS Workbench parameter manager. If all parameters are available, optiSLang inside ANSYS Workbench is the best solution. If input or output parameters have to be added, the integration of ANSYS Workbench projects into optiSLang and the extension of parameter and process workflows using optiSLang process integration functionality is the solution to solve an RDO problem. For integration of ANSYS Workbench projects, the Integration node as well as the file base communication via optiPlug functionality is available. Thus, optiSLang's functionality of signal processing is included and the CAE workflow can be expanded by additional solvers or pre and post processing tools.

Best practice modules

However, parametric modeling, process automation and integration are just prerequisites before we can start solving an RDO task. To achieve a regular, daily use of tools like optiSLang, we cannot expect every user to be a specialist in defining algorithmic settings and workflows. Therefore, a state-of-the-art optimization software has to provide best practice algorithms with best available defaults and wizard guidance. Furthermore, a modular usage of a sensitivity analysis, optimization or robustness evaluation has to be possible. We think that multi-purpose optimization toolboxes like optiSLang have to be capable of facing this type of paradigm shift just in the same way general purpose FEM

systems did 20 years ago. In that time, experts carefully designed their meshes and programmed the FEM solution procedure as well as the post processing. The broad use of FEM in today's virtual prototyping was only possible after modules like "mesh" and "solve" became available where algorithmic innovations are applied and best practice solution are conducted. The necessary input has to be minimized and users are safely guided to perform their engineering task. In accordance to these requests, optiSLang provides three modules for best practice RDO algorithms and workflows:

1. **sensitivity analysis** to understand the design, to reduce parameter to the most important ones, to check forecast quality of response variation and to automatically generate the best possible meta model
2. **optimization** to improve design performance
3. **robustness evaluation** to check design robustness exposed to scattering material parameter, production tolerances or scattering environmental conditions

As a consequence, in optiSLang inside ANSYS, the 3 modules can be applied with drag and drop onto the scenery of the project page (see figure 1). With these modules, the user input is reduced to an absolute minimum amount, which is the setting of parameter ranges, scatter, constraints and objectives (see figure 2,3). All algorithm settings are generated with best practice defaults and a wizard guided modular workflow. Within the optimization modules, algorithms are available choosing the most efficient and fitting optimization strategy on the basis of the sensitivity analysis and user input (see figure 4).

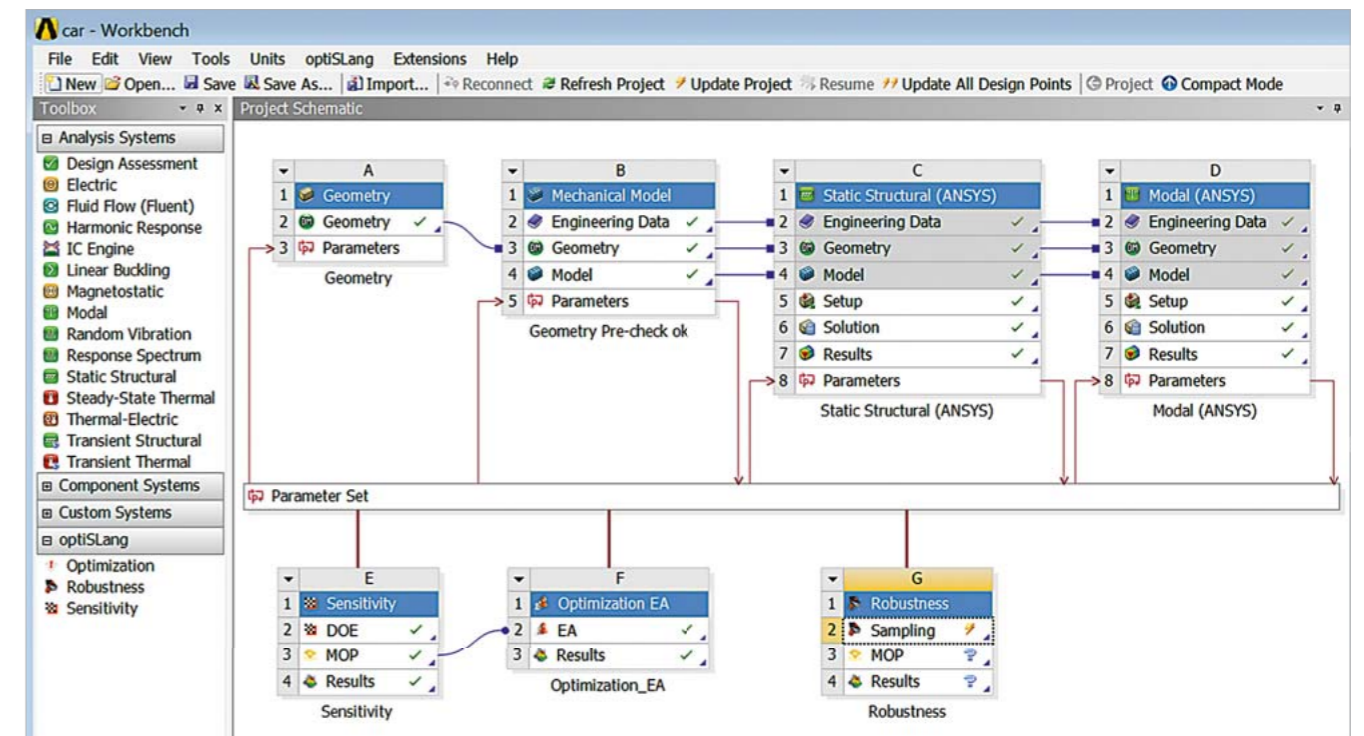


Fig. 1: screen shot of an ANSYS Workbench project page showing the three optiSLang drag and drop modules used to define and RDO application using sensitivity analysis, followed by optimization and robustness evaluation

Coefficient of Prognosis (CoP) and Meta model of Optimal Prognosis (MOP)

Running a sensitivity analysis, optiSLang identifies automatically the most important parameters. To make this procedure as efficient as possible, Dynardo developed the CoP for quantifying the forecast quality of the response variation and as a criteria to find the best possible MOP in the best possible subspace of important parameters. It is also used to quantify the numerical noise at the response values. Finally, the MOP represents the best possible meta model in the best possible subspace of important parameters to maximize the forecast quality of response variations (see figure 5). This methodo-

logy is one key for efficiency. Thus, a “no run too much” philosophy can be implemented for sensitivity analysis and optimization to minimize external CAE solver calls.

Secure workflows

Robust Design Optimization will result in the creation and calculation of a significant number of designs. In real world projects, we have to face design failure because modified geometry cannot be regenerated, mesh algorithms fail or the calculation process shows inaccuracy. As a consequence, developing secure saving and storage procedures have first

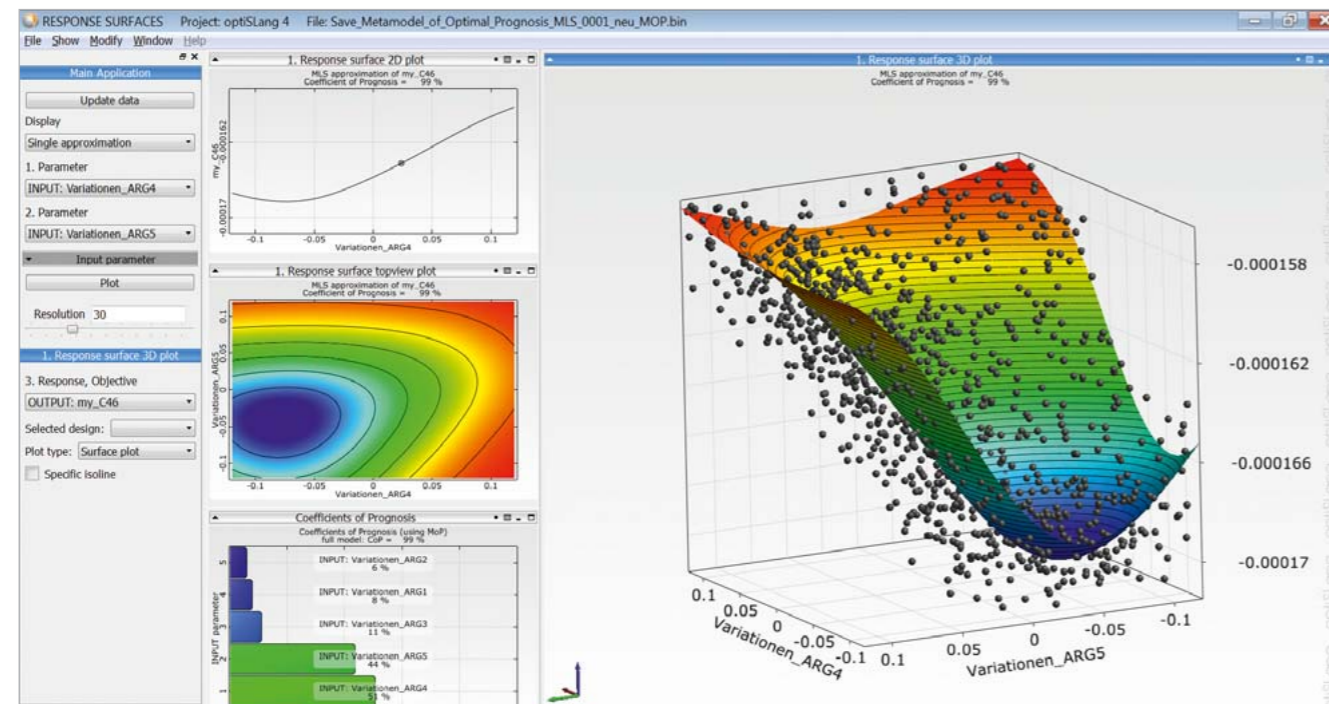


Fig. 2: Post Processing: multiple window GUI for post Processing of meta model og optimal Prognosis

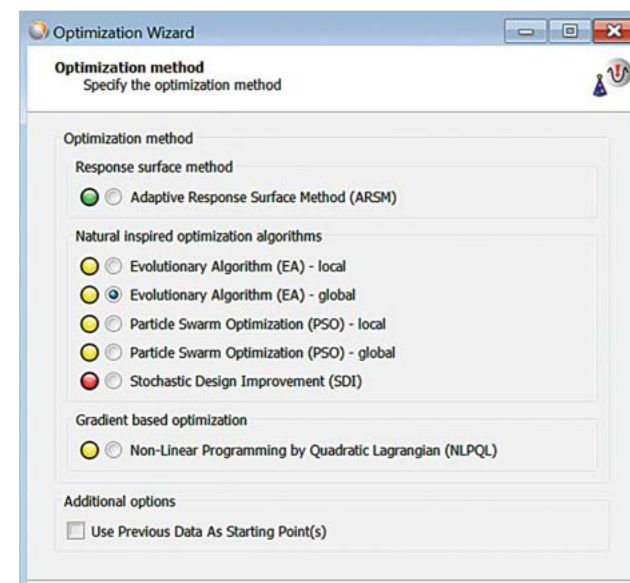


Fig. 3: wizard indicates the most effective optimizer. green traffic light: recommended optimizer, yellow traffic light: possible alternatives, red traffic light: not recommended

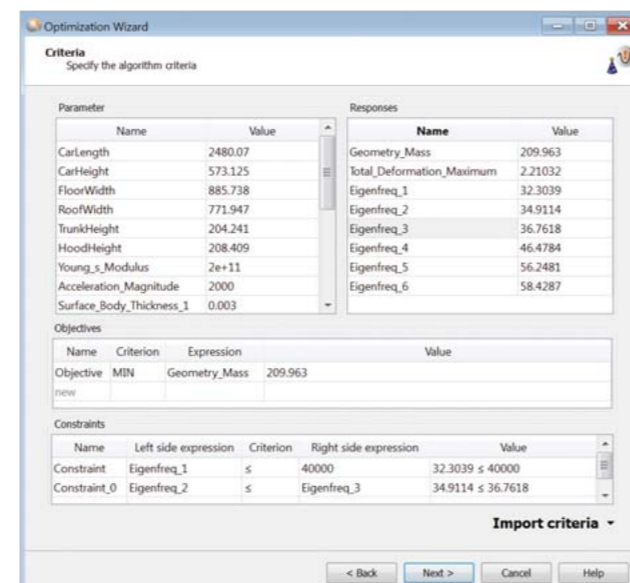


Fig. 4: wizard for defining constraints and objectives

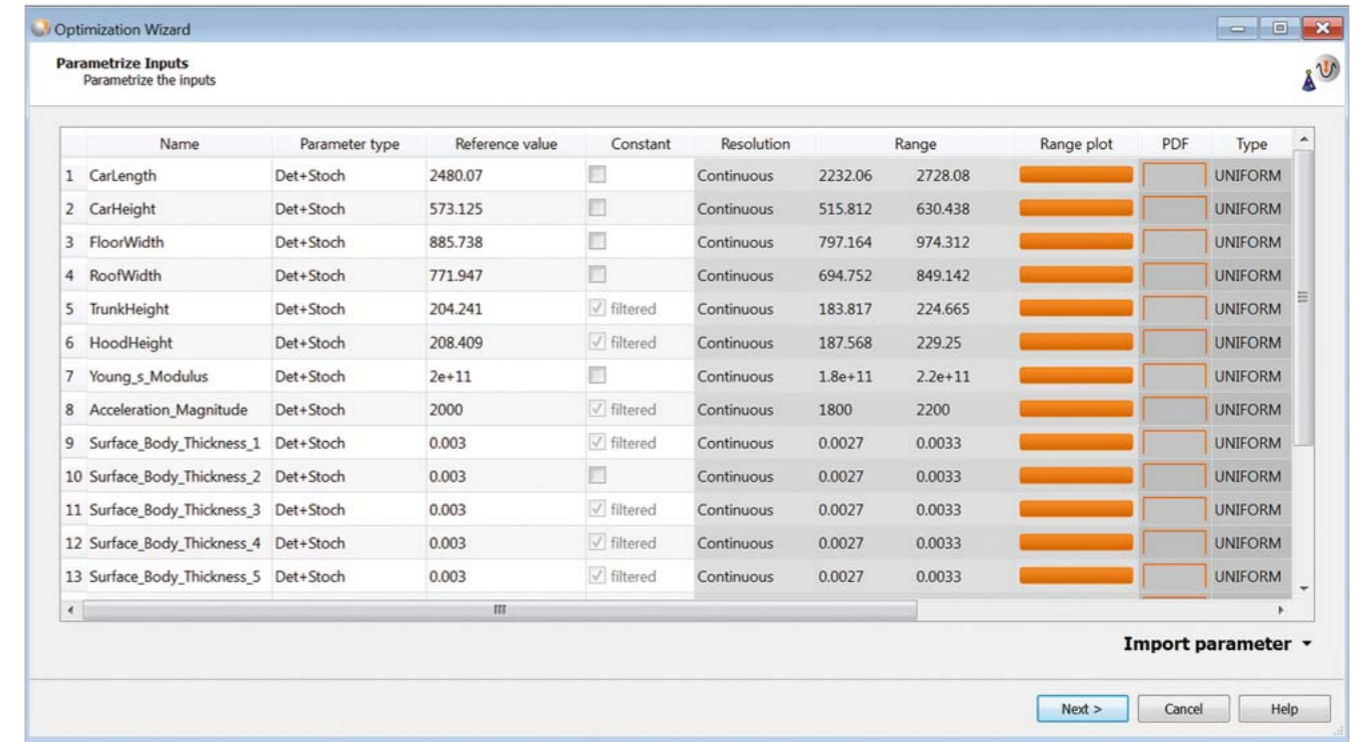


Fig. 5: wizard to define the variation of optimization as well as the scatter of variables

priority. In the optiSLang 4 stand-alone version as well as in optiSLang inside ANSYS Workbench, we integrated a continue crashed session mode preventing any loss of data as well as a design backup to make them available for restart and reevaluation at any time of the workflow.

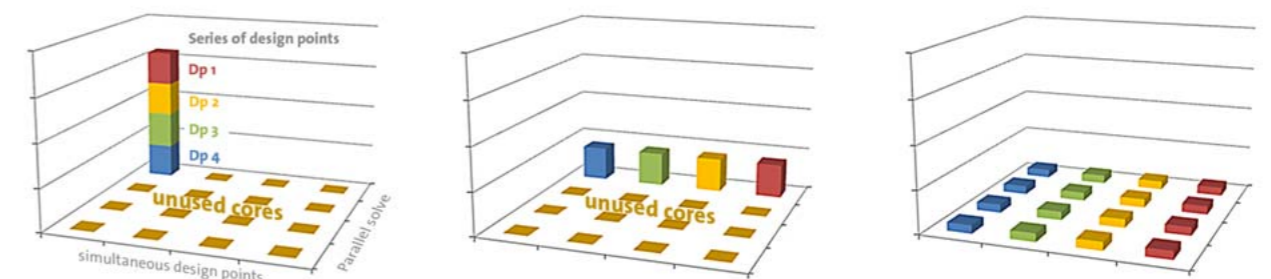
More efficiency by ANSYS HPC support

If one design evaluation needs a significant amount of time to be solved, two ways of speeding up the process are available. First, with ANSYS HPC functionality where every design update can be executed by using multiple cores. Second, for simultaneous execution of multiple designs, parametric pack licenses (from v14.5) are available. Here, necessary licenses for updating a design point are multiplied. Thus, user can appropriately distribute jobs across the available compute resources. For example: 4 HPC Parametric Pack Licenses allow for 32 design points to run simultaneously

and one additional HPC Parametric Pack License for 64 design points. These techniques enable different systematic approaches using remote compute resources. It is not only possible to run the solution process (the solver) remote, but also other parts of the process chain, like result extraction, can be run on the remote server, which is especially useful for large data sets.

Product bundle “optiSLang for ANSYS”

To provide our users access to all that functionality which can be connected to ANSYS Workbench within one license, since April 2013, we have been offering the product bundle “optiSLang for ANSYS” containing optiSLang inside ANSYS Workbench, optiPlug and optiSLang stand-alone including ANSYS Workbench integration node. For distribution, support and improvement, we are glad to work together with the CADFEM GmbH as our long term ANSYS partner.



update of 4 design points using one ANSYS License and a Hardware containing 16 cores

update of 4 design points using one ANSYS License + 1 HPC Pack using multiple cores to solve the design

update of 4 design points using one ANSYS License + 1 HPC Pack + 1 HPC Parametric pack to solve 4 design points using 4 cores each at the same time. Theoretically update can be executed 64 times faster