

LS-OPT[®]: New Developments and Outlook

Nielen Stander and Anirban Basudhar

*Livermore Software Technology Corporation
Livermore, CA 94588*

Abstract

New features available in LS-OPT 5.1 are discussed and illustrated. The main features include three new solver types, Parallel Feedforward Neural Networks, seamless variable de-activation for iterative methods, exporting of selected metamodel formulae, subregion-based Global Sensitivity Analysis, enhanced histogram visualization features and Viewer-based categorization of simulation results.

Introduction

LS-OPT is a simulation-based Multidisciplinary Optimization (MDO) program which functions as a companion code to LS-DYNA[®]. While the solver interface (also referred to as the *Stage*) has mainly focused on LS-DYNA integration, it also features interfaces for a large number of pre-processors, post-processors and Third Party solvers. The main analysis and optimization features of LS-OPT include Multi-Objective Optimization and Reliability-based Optimization. Both Direct and Metamodel-based optimization methods are supported and a variety of metamodel types are available.

While Direct Optimization methods are available, LS-OPT has maintained a strong development activity in the area of metamodel-based methods, mainly for the reason that these provide the following capabilities normally not associated with direct methods:

- Global exploration of the design space: approximate responses for any design can be obtained interactively.
- Reliability and Robustness Analysis and Optimization.
- Global Sensitivity Analysis.
- Outlier Analysis: identifying the locations of sources of noise in non-linear analysis response.
- Tolerance-based Design Optimization.

Version 5 of LS-OPT was released in 2013 [1]. The principal enhancement was Process Optimization which allows the user to integrate a process flow chart into the optimization procedure. Branching and merging of the process flow is supported. A completely redesigned Graphical User Interface (GUI) was therefore introduced. This interface was more modular in its design and has hence simplified the addition of further enhancements without cluttering its simple visual appearance and functionality.

Enhancements for Version 5.1 can roughly be categorized as follows: (i) interfaces for new solver types, (ii) enhanced modeling and optimization methods and (iii) enhanced graphical post-processing features.

New feature development for 5.1 was mostly driven by a need for greater speed and more informative visualization. The basic principle observed throughout has been to keep the interface as simple as possible, while providing more features.

In this paper, the following new features are discussed and illustrated:

- The introduction of LS-OPT, Excel and a Third Party solver as additional solvers that can be used for simulation and post-processing. The Third Party option refers to a popular non-LS-DYNA Finite Element analysis code. The introduction of LS-OPT as a solver type adds the dimension of multi-level (i.e. nested) optimization while Microsoft Excel can be used as a solver or post-processor stage in the process.
- Several features have been implemented to enhance or speed up existing analysis and optimization features. The main feature added in this category is Parallel Neural Networks. In addition, the de-activation of variables has been added for use in iterative methods. This allows the user to seamlessly de-activate a variable at any time before re-starting the iterative solution. Also, a new metamodel optimizer, Differential Evolution, was added for continuous constrained optimization.
- Global Sensitivity Analysis (GSA) in subregions of the design space and the ability to repair existing GSA results.
- Several features have been added to the Viewer (the post-processor of LS-OPT). The main features added are significantly enhanced histogram display features and categorization of simulation results.

In addition to these specific features, a variety of minor enhancements have also been introduced. Most of these were implemented to ensure a more reliable and seamless user experience.

Parallel Neural Networks

Feedforward Neural Networks (FFNN) are known for their ability to provide suitable smooth global approximations of design response. In contrast to some other approximation methods such as polynomials and Radial Basis Function Networks, which require linear regression, FFNNs are expensive to compute. This can have a negative effect on the overall time for a metamodel-based analysis, especially for high dimension design spaces and a large number of responses. The high cost emanates mainly from the fact that constructing an FFNN requires a non-linear regression approach involving optimization. The non-linearity requires that each response be solved independently, thereby significantly increasing the overall cost of regression. The use of Neural Net ensembles and committees, which are subject to user choice, adds a further dimension to the cost. Because 5-10 hidden nodes and 10-20 members are typically used, this can multiply the cost of a primal FFNN construction by between 50 and 200 (actually a higher order choice scales superlinearly). Ensembles and committees improve accuracy and consistency.

Although the computation of metamodels is generally much cheaper than direct solution (and provides much more information) a serial approach to metamodel solution for FFNNs can be

very expensive when many responses are required. Experience has shown that as many as a hundred responses (intrusions at various locations, etc.) are often involved in automotive crash analysis.

In Version 5.1, an algorithm was introduced to solve the FFNNs concurrently. This involves the application of the job scheduler to treat each response and each member of a Neural Network ensemble as a job to be run in parallel. The main features are as follows:

- As a result of the fact that the existing job scheduler in LS-OPT is used for the parallel analysis, all of the graphical job-monitoring features are automatically available for this purpose.
- Remote computation is supported, so if the user already has a cluster setup for LS-DYNA jobs, the FFNN solution may only involve a simple setting.

The dialog for parallel building is depicted in Figure 1. Using this feature, an optimization of a vehicle crashworthiness problem with 1000 simulations, 15 responses and 10 ensemble orders (unusually high) including 30-member committees was conducted. The parallel solver required 75 minutes to complete this analysis using 8 processors. Because of the absence of message passing, the scaling is expected to be close to linear.

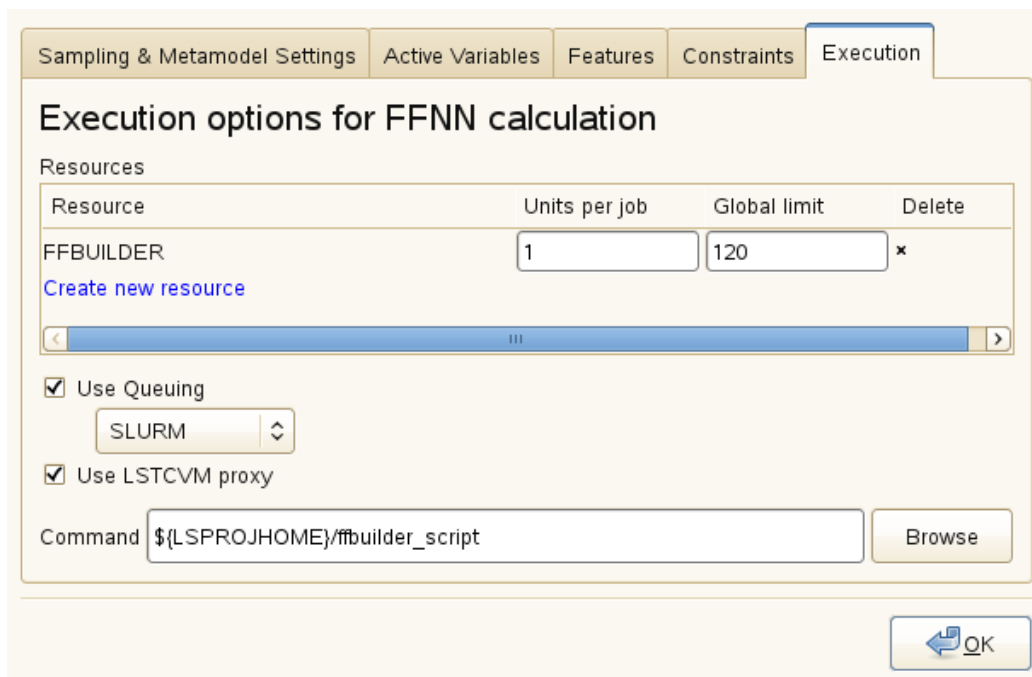


Figure 1: LS-OPT GUI dialog for Parallel Neural Networks builder.

New solver or stage types

Three new stage types have been added in version 5.1 - Excel, LS-OPT and a Third Party Solver. The Excel stage or solver can be run only on a Windows system. The details of the three stages are explained in the following sections.

Multilevel optimization using LS-OPT stage

The basic idea in multilevel optimization is to perform optimization at multiple levels. Each sublevel optimizes a subset of the variable set while maintaining constant values for the variables belonging to preceding levels. In this version, an LS-OPT stage type has been added for this purpose. LS-OPT can call another instance of LS-OPT in a nested setup and run it as a solver. The LS-OPT stage has several unique features compared to others. Unlike any other stage type, the response of an LS-OPT stage itself is optimized with respect to certain variables (referred to as free variables). These optimized responses are then further optimized with respect to the outer level variables. Thus, there are multiple levels of optimization in the problem setup. In the context of multilevel optimization, two of the components of the LS-OPT process flow have particular significance:

- **Stage:** The LS-OPT stage allows the definition of an optimization problem within another optimization. The inner level optimization is simply treated as an LS-OPT stage within the outer optimization process.
- **Global Setup:** Like any other stage, the inner level optimization setup needs to be parameterized to enable the flow of information from the outer level(s). These parameters are treated as constants in the inner level but are variables in an outer level. This behavior is made possible by using a new variable type called "Transfer Variable".

A multilevel optimization setup with an LS-OPT stage and two outer level variables is shown in Figure 2. The responses are defined in the "Responses" tab of the stage dialog (Figure 3). LS-OPT stage responses, which are optimized with respect to the inner level free variables, can be defined as any existing entity in the inner level optimization history. A list of available entities is provided in the response definition dialog.

As mentioned previously, multilevel optimization uses variables of a special type called "Transfer Variables", which are treated as constants for the inner optimization, but are optimization variables in the outer level setup. The global Setup dialog of an inner level optimization consisting of two Transfer Variables is shown in Figure 4. It should be noted that the Transfer Variables for the inner level are defined as continuous variables in the outer level (Figure 2).

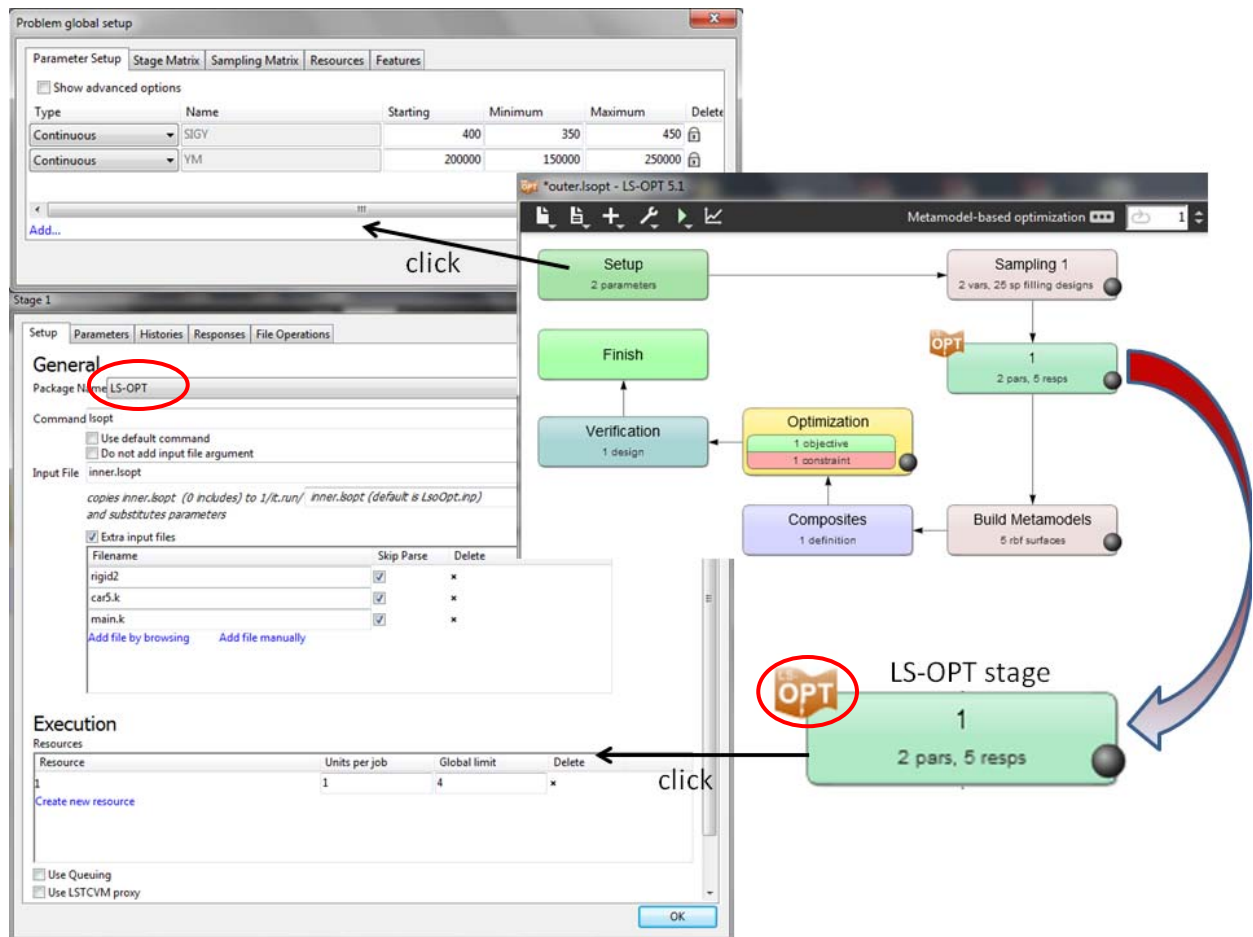


Figure 2 : Multilevel optimization outer level setup. Clicking the stage box opens up the stage dialog where the package name is selected as LS-OPT. The two outer level variables are treated as constant parameters for the inner level optimization in stage "1".

The entire multilevel optimization process in LS-OPT can be briefly summarized as follows.

1. The inner level optimization (i.e. LS-OPT stage) setup is parameterized using Transfer Variables, which are treated as constants at this level. The inner level optimization is performed with respect to the free inner variables for each outer level sample; the Transfer Variables are set by LS-OPT as the values of the outer level variables at the samples. The "optimized" responses (with respect to the free inner level variables) are recorded as responses defined in the LS-OPT stage "Response" dialog of the outer level.
2. The outer level variables are defined in the global "Setup" as continuous or discrete. These are automatically recognized and added in the GUI if the variables exist in any stage input file (including Transfer Variables defined in the inner level .isopt file).
3. The outer level variables are sampled and evaluation of each sample involves an inner level optimization.
4. The outer level variables are optimized based on responses that are results of the inner optimization at all the outer level samples.

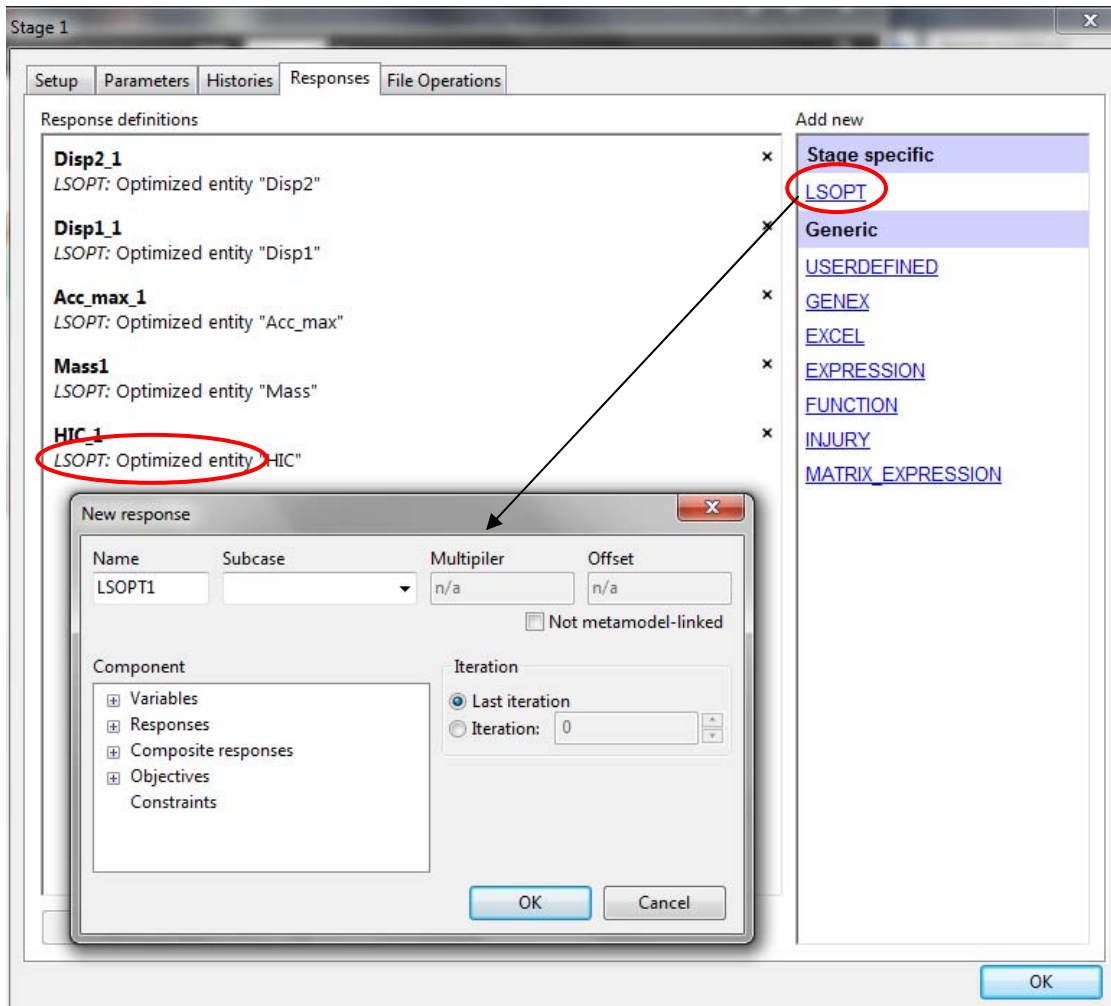


Figure 3 : LS-OPT stage responses. Optimized inner level variable values and the corresponding response, composite, objective function and constraint values can be selected as responses.

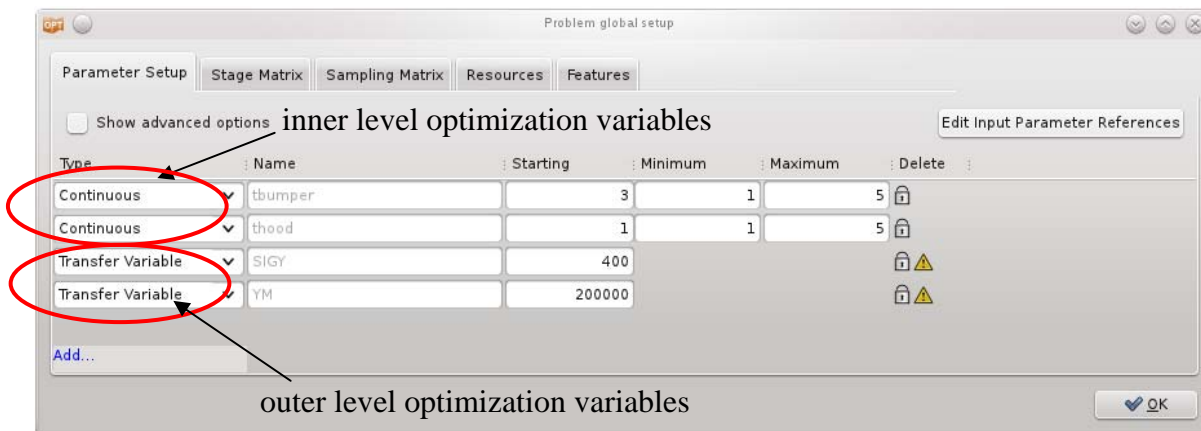


Figure 4 : Variable setup for inner level. Two of the variables are defined as the new type "Transfer Variable" and are variables in the outer level.

Excel Stage

In LS-OPT 5.1, an Excel stage can be used as a solver or a post-processor. It can be seen as being similar to any other solver, with the main differences lying in its parameterization and in the response and history definitions. Because the results need to be computed for several samples within an LS-OPT task, the Excel input file needs to be parameterized. This is achieved using *Input definitions* specified in the Stage dialog. These inputs may correspond to a single Excel cell or a group of cells in the input file, and are substituted for each sample (Figure 5).

The attributes used for *Input definitions* are *Sheet* and *Cell(s)* (these point to a specific location, i.e. cell or group of cells in the Excel document), *Type* and *Value* (used to link LS-OPT design parameters, histories and responses with corresponding fields of the parsed *Excel* document, e.g. a parameter type linking a variable to an Excel cell or a history type linking an LS-OPT history to a group of cells in the Excel document) and *Fill Direction* (specifies whether history values are written horizontally or vertically to the Excel file). For more details, the user may refer to the LS-OPT 5.1 manual.

The Excel responses and histories are defined as the value(s) of a cell or a group of cells in the response or history dialog (Figure 6). The histories and responses specific to *Microsoft Excel* can be defined using the *EXCEL* option listed under *Generic* history and responses interfaces.

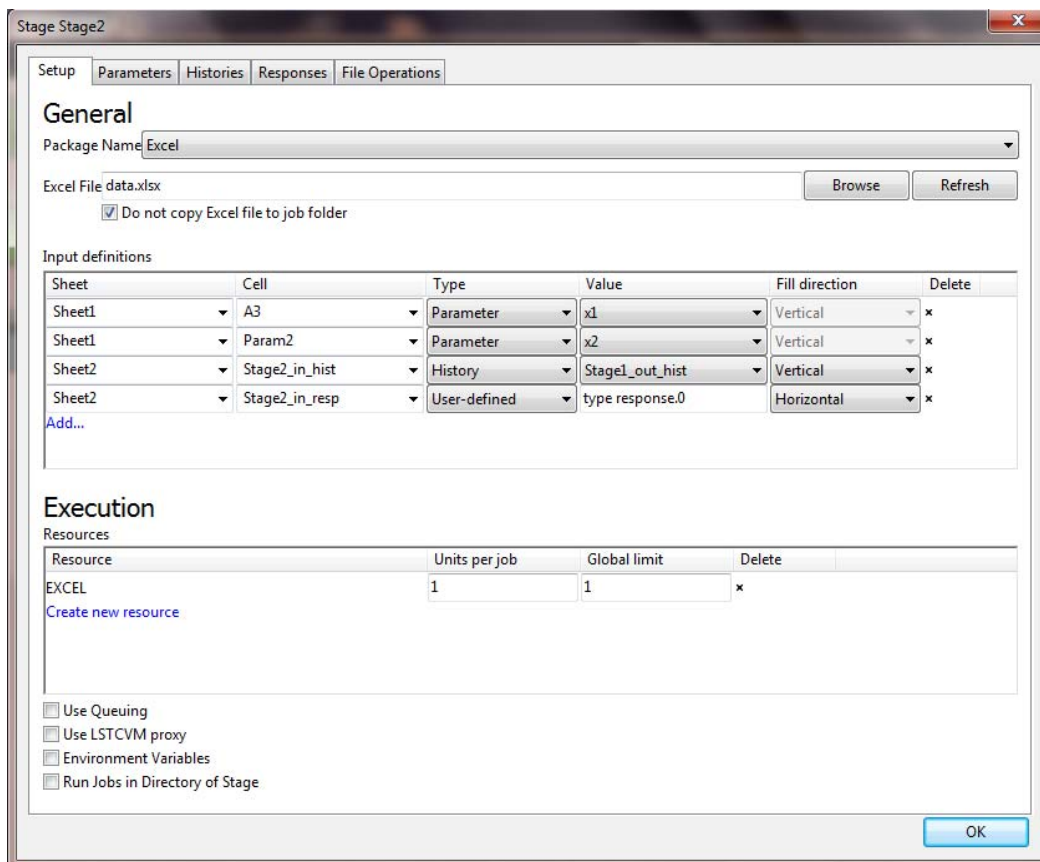


Figure 5 : Input definitions in Microsoft Excel Interface

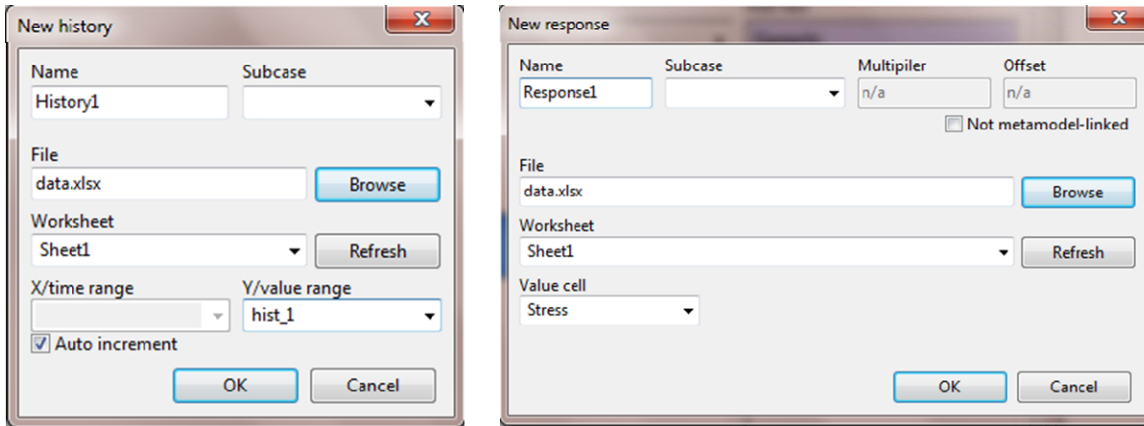


Figure 6 : Microsoft Excel history (left) and response (right) interface

Third Party solvers

In the context of this paper, a Third Party solver refers to a solver other than LS-DYNA. Since LS-OPT is a Multidisciplinary design code, designers almost always use other solvers in conjunction with LS-DYNA. For example, a user may use LS-DYNA for crash simulation while using another FE solver for frequency analysis, and yet another one for stress or durability analysis.

In Version 5.1 an interface has been added for a third popular commercial Finite Element Analysis program. The interface only addresses the parameterization of the simulation which is introduced into the problem using parameter definition options (e.g. *PARAMETER). As with LS-DYNA, syntax rules such as recursive include files and the specification of parameters at any level is supported. The new feature obviates the need for the double angular brackets normally required for introducing parameters in generic solvers. The Third Party interface is available under the User-defined Stage type by simply specifying the package name as a comment on Line one of the main input file. Response and history extraction can be conducted using GenEx (included with LS-OPT) or other commercial extractors.

Variable De-activation

Optimization in a high dimensional variable space requires a large number of simulations. When using Sequential methods, this number can be reduced by de-activating unimportant variables in any (typically the first) iteration. In the past, the user had to create a new setup to extend the optimization using a reduced number of variables and most recent approximate optimum. This step is now seamless and only requires the user to de-activate the unimportant variables by unchecking them before re-starting.

Global sensitivity analysis in subregions

Sensitivity analysis is often performed during a design process for various reasons, such as to get an insight into response variations, screen variables etc. LS-OPT consists two options for sensitivity analysis - Analysis of Variance (ANOVA) and Sobol indices-based Global Sensitivity Analysis (GSA). The GSA option in previous versions provides a global sensitivity measure over

the entire design space defined by the variable bounds. It may, however, be interesting to study the sensitivities over a specific subregion of the space. The capability to do that for multiple subregions is available in LS-OPT version 5.1. The subregion variable bounds are defined in the "Global Sensitivities" dialog of the LS-OPT GUI (Figure 7). It is possible to define subregion bounds only for a subset of the variables; for the other variables, the global design space bounds are then used. It should be noted that the response metamodels are only constructed once, and GSA for each region is done based on these metamodels by generating different sets of samples.

A subregion can be deactivated by unchecking the "Active" box against it. Also, existing GSA results can be overwritten by checking on the "Overwrite" option. By default, sensitivity analysis for a subregion or the global region is skipped if the corresponding results are already present. For the global region, the GSA results are saved in the working directory in files named as *Sobol_GSA.n*, *n* being the iteration number. For a subregion, a similar file named as *Sobol_GSA.subregion_name.n* is saved. It is noteworthy that the combination of "Active" and "Overwrite" options provides the ability to selectively delete or repair GSA results for one or more subregions.

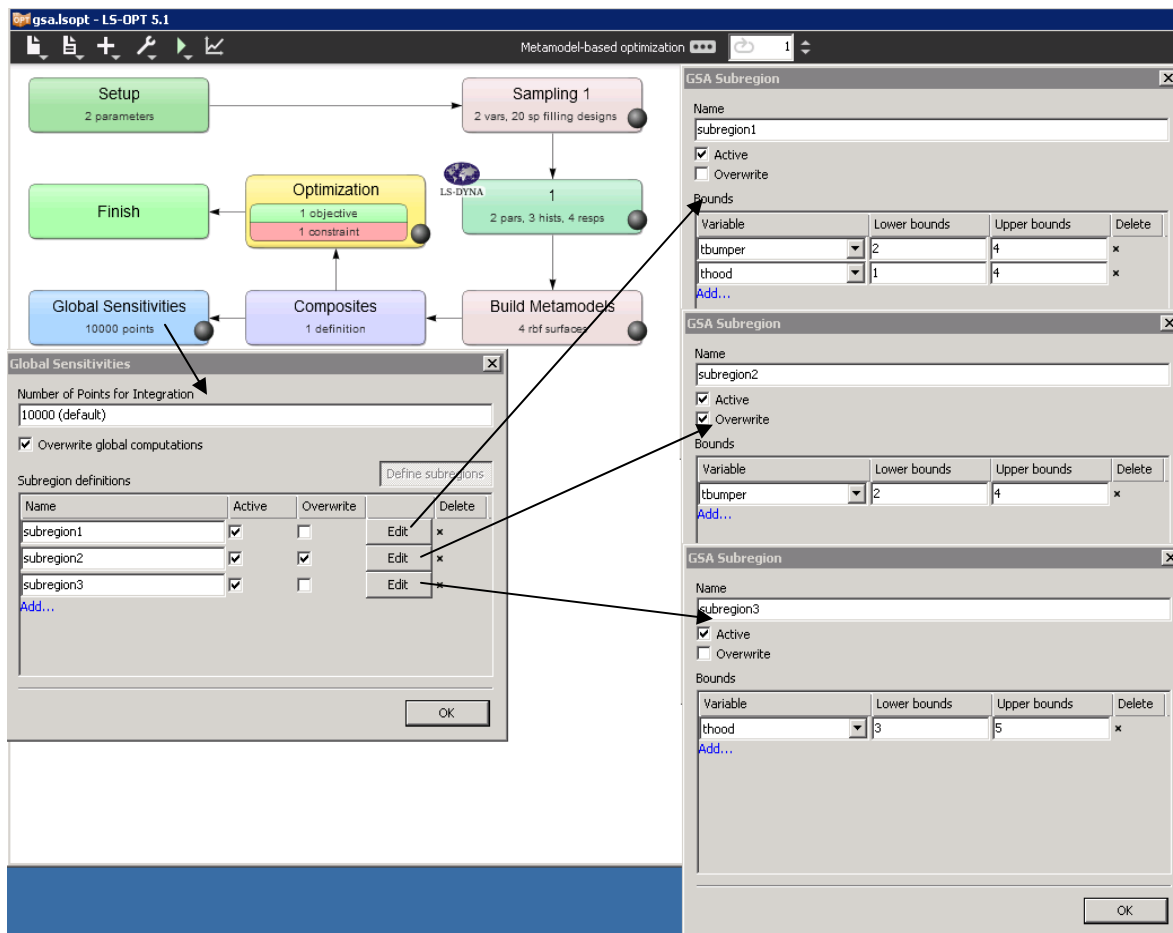


Figure 7 : Global Sensitivity Analysis in subregions. Three subregions with different bounds are defined.

The subregion GSA results can be seen in the Viewer Sensitivities Plot by selecting the region of interest from a list (Figure 8, top). By default, the global region is selected. It is also possible to

visualize the region of space corresponding to a subregion in the Scatter Plot and the Metamodel Surface Plot, and to only plot the points that lie within the subregion (Figure 8, bottom).

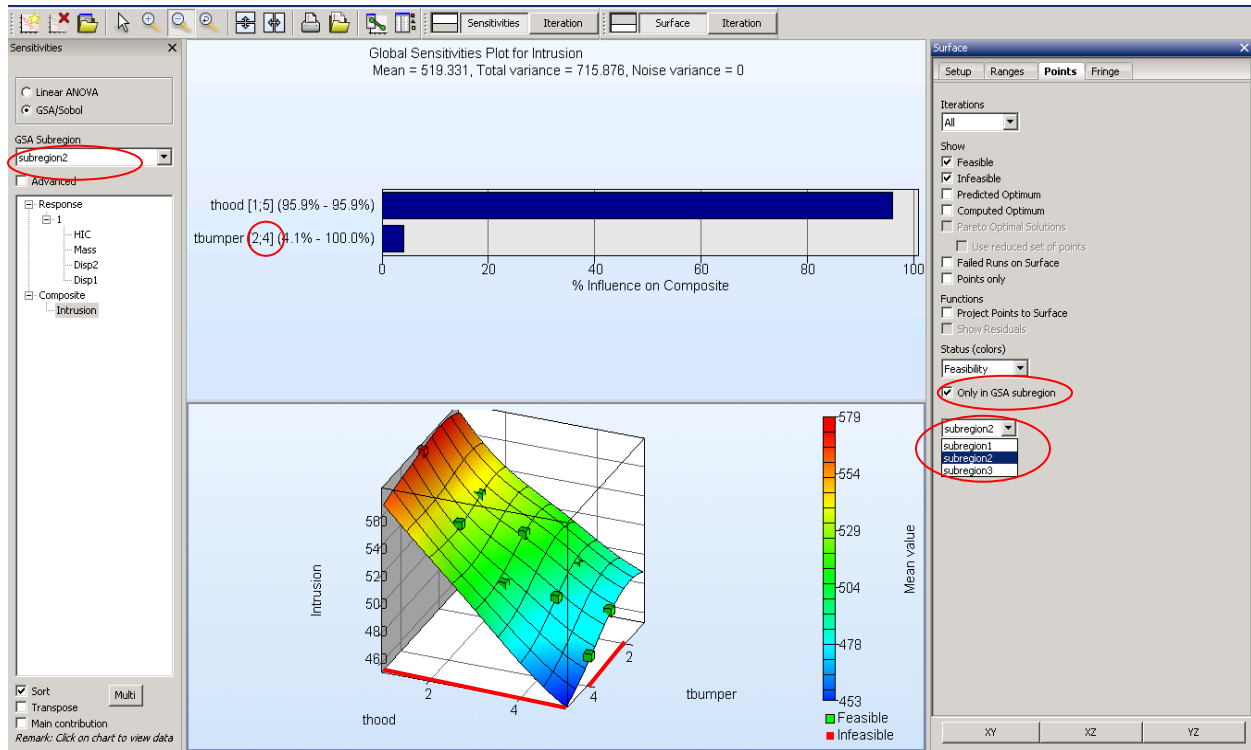
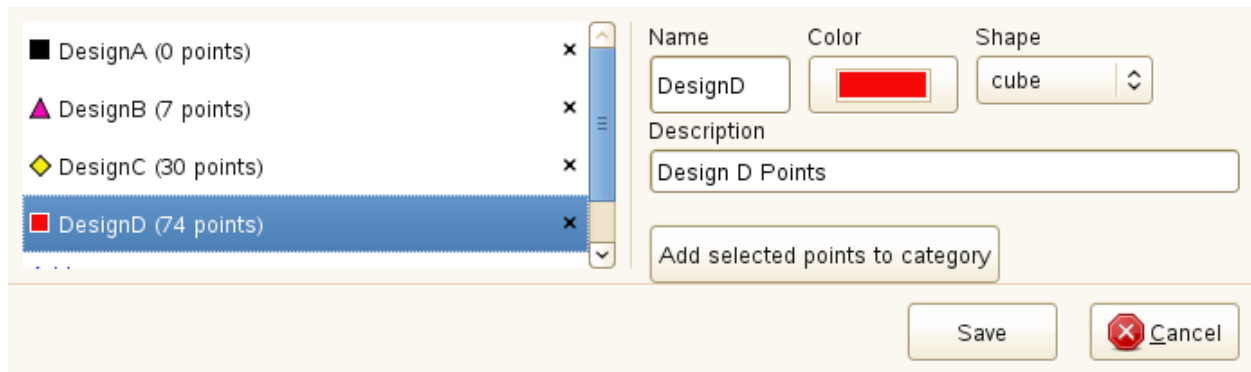


Figure 8 : Subregion GSA result visualization. The axes are colored red in subregion ranges.

Categorization of Simulation Results

The possibility of generating a large number of simulation results also results in a need to group or categorize the results. Designers typically consider some results to be of greater practical importance than others. This may apply in particular to Multi-Objective Optimization where a large number of simulations are optimal, but only some are of practical value. Figure 9 (top) depicts the dialog for defining categories while Figure 9 (bottom) shows the categorization of a few groups of selected points. Point color and type can be selected as distinctive attributes.



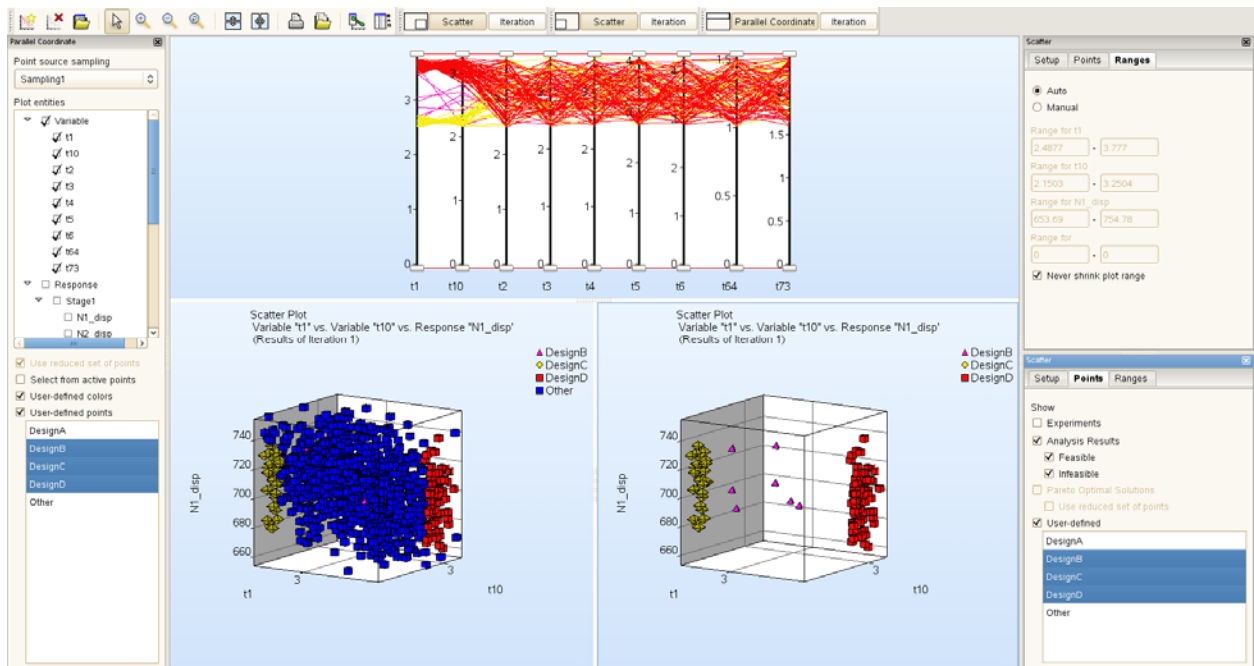


Figure 9 : Dialog and display of user-selected design categories

Histogram Visualization Capabilities

A histogram plot is used to study the statistical variations of responses and variables. The histogram visualization capabilities of the viewer have been improved in this new version of LS-OPT. Three main capabilities have been added:

- Manual axis control of the region of interest on a plot.
- Graphical visualization of statistical entities and constraints on the plot.
- Additional histogram types in addition to the existing frequency histogram.

Manual axis control includes both range and step size. The range control allows the user to focus on a specific part of the plot while the step control provides flexibility in determining the required resolution of axis ticks for better visualization (Figure 10). Both plots in Figure 10 refer to the same histogram, but the tails of the histogram have been discarded in the bottom plot using manual axis range settings. Also, a higher resolution of axis ticks is used in the bottom plot. A magnified view of the axis control settings is shown in Figure 11.

The second enhancement pertains to the visualization of additional entities on the histogram plot. More specifically, the visualization of the statistical mean, standard deviation and the constraint bound using vertical lines has been enabled. In addition, feasibility of the different parts of the histogram range can be visualized using distinct background coloration - green for feasible and pink for infeasible (Figure 12).

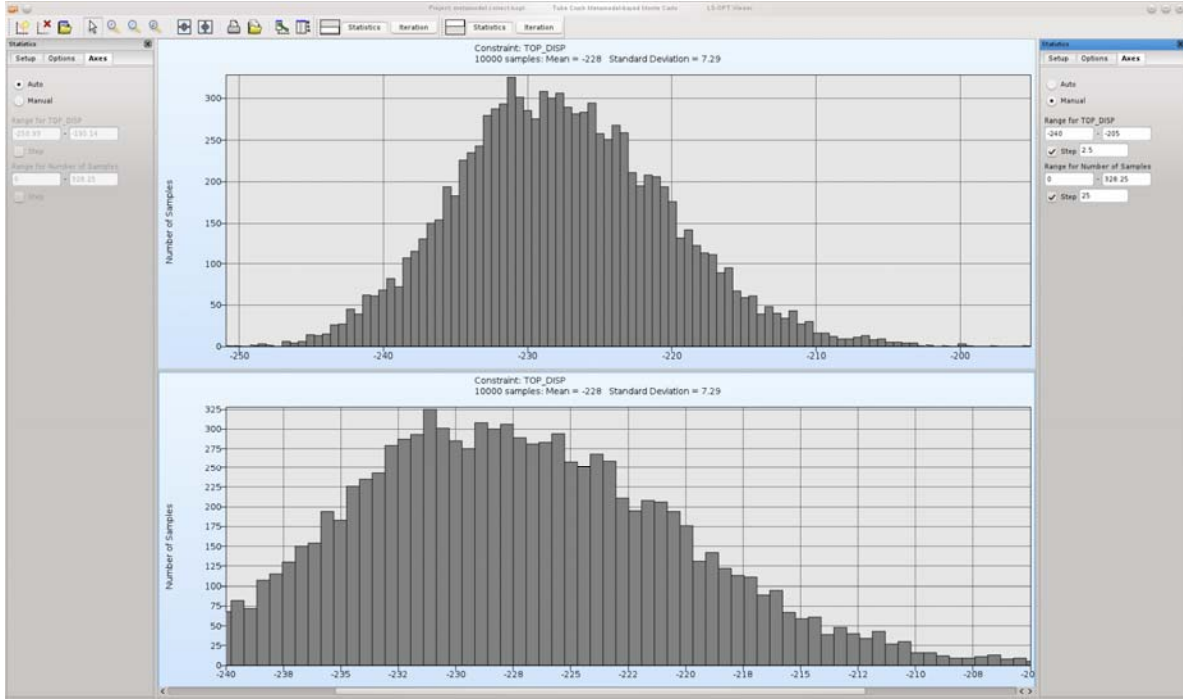


Figure 10 : Histogram manual axis control. The top plot represents the automatic axis settings. Both the range and step size have been manually modified in the bottom plot.

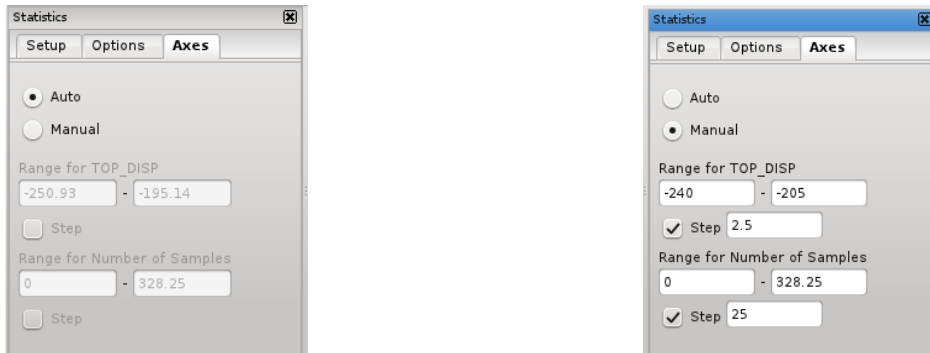


Figure 11 : Comparison of automatic and manual axis settings of the plots in Figure 1.

Finally, LS-OPT 5.1 has two additional options for the histogram type - "Probability" and "PDF" (Figure 13). These plots correspond to the relative frequency histogram and the relative frequency per unit class width histogram. The bin heights in the relative frequency histogram are obtained by dividing the frequency histogram by the total number of samples. Thus, the height of a bin represents the probability of belonging to that bin, and must always lie between 0 and 1. In the "PDF" histogram plot, the probability values of the relative frequency histogram are further divided by the class width or interval, i.e. the width of the bins. Division of probabilities by the bins gives "probability density" values that are independent of the class width or the number of bins. This independence is an important property because of which the probability density function (PDF) is widely used for the representation of probabilistic entities. It should be noted that unlike probabilities, PDF values can be greater than 1.

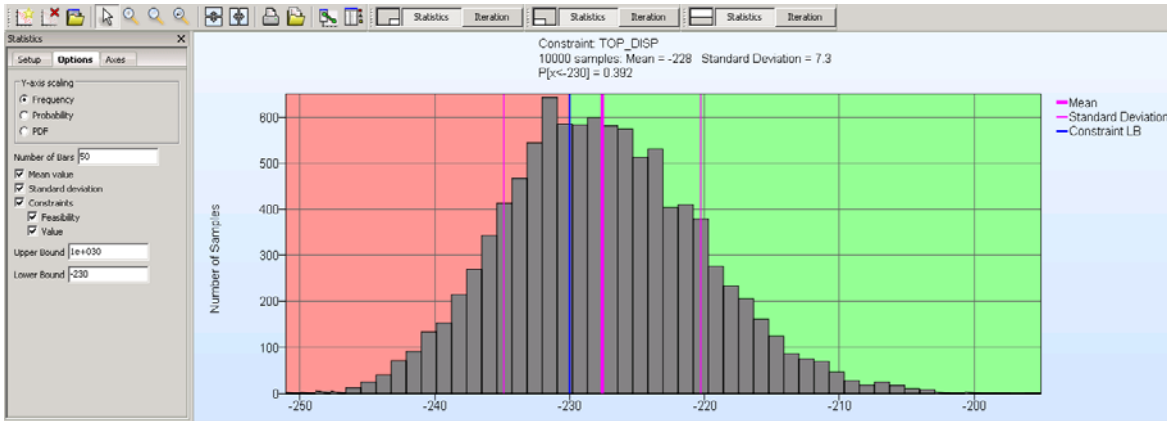


Figure 12 : Visualization of constraint value (blue line), constraint feasibility (background color), statistical mean (thick magenta line) and standard deviation (thin magenta line) on the histogram plot.

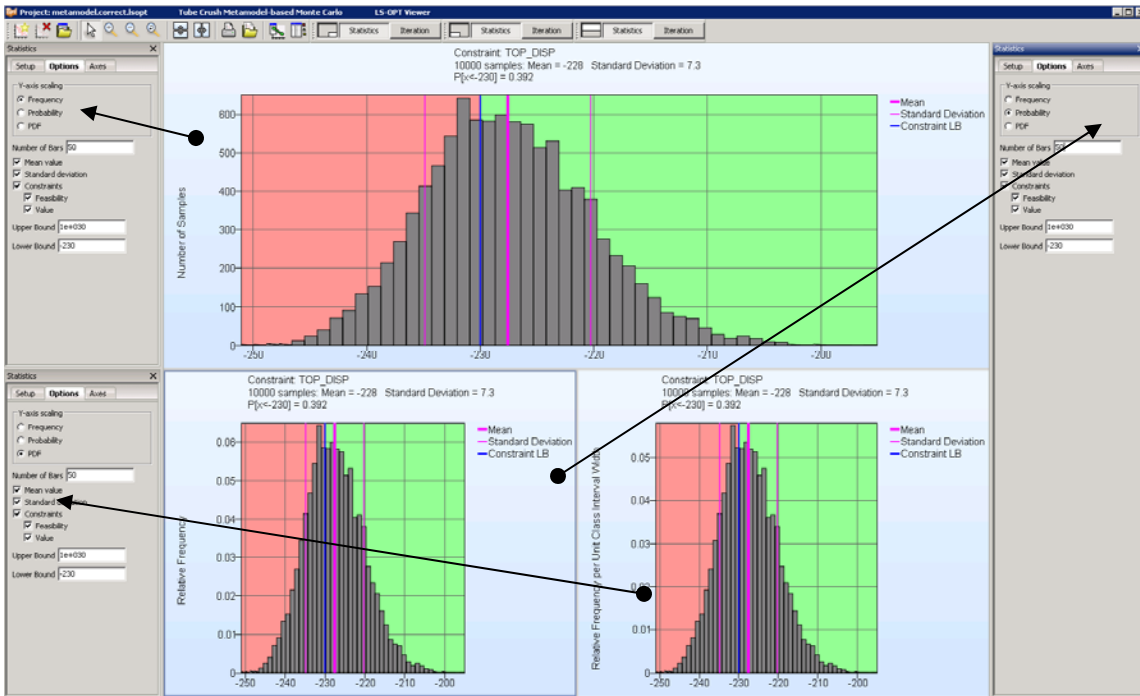


Figure 13 : Frequency, Probability and PDF histograms for the same entity. The difference in ordinate values should be noted.

Exporting metamodel formulae

The 5.1 version of LS-OPT writes the formulae for an approximated metamodel into the lsopt_output file. Thus, apart from visualizing the metamodel surface in LS-OPT, one can use the formulae to compute response values or for visualization in any other program. In case of a high fidelity approximation, the availability of a mathematical formula allows it to be used in any future computation without the need to process the data used previously for the approximation. The formulae written by LS-OPT are directly compatible with Matlab. This feature is currently available for polynomials and radial basis function metamodels.

Closure

With the most recent iteration, the capability of the LS-OPT design tool has been significantly enhanced while its simplicity for simulation-based optimization has been maintained or improved. The recent emphasis has been on broadening its applicability by supporting additional solver types. Enhanced transparency of the result database is provided by a number of Viewer refinements while certain computational bottlenecks such as the building of Neural Networks have been addressed through parallelization.

In addition to providing better core algorithms and a more comprehensive solver interface, the future development activity of LS-OPT will likely focus on providing better tools for collaborative design, introducing further parallel solution options where needed and addressing special applications such as parameter identification, sheet metal forming and tolerance-based design.

Acknowledgements

The testing and validation of LS-OPT Version 5.1 conducted by support engineer Imtiaz Gandikota is greatly appreciated. The authors also wish to thank developers Åke Svedin, Katharina Witowski and Trent Eggleston for their development contributions to Version 5.1.

References

1. Stander, N., Roux, W.J., Goel, T., Basudhar, A., Craig, K. LS-OPT User's Manual, Version 5, June, 2013