LS-OPT® TRAINING CLASS

OPTIMIZATION AND ROBUST DESIGN

TUTORIAL PROBLEMS

LS-OPT Version 4.2

Compiled by: Nielen Stander (Ph.D.) and Willem Roux (Ph.D.)
Livermore Software Technology Corporation

July, 2011

TABLE OF CONTENTS

INTRODUCTION

This tutorial problem set allows LS-OPT users to exercise aspects of mathematical optimization, design optimization and robust design. LS-OPTui is used to create or modify the input.

A first part of the problem set attempts to simulate the design process for examples that are typical for the LS-DYNA® user. Detours have been introduced to investigate features related to a particular design step in more detail. The examples are

- Crashworthiness optimization of a vehicle
- Mode tracking
- Multidisciplinary Design Optimization
- Material parameter identification

Another example has an explicit algebraic formulation. The purpose is to introduce the user to the mathematical aspects of approximations, accuracy and convergence. The example is

- Nonlinear explicit problem

A further special section is dedicated to understanding different optimization strategies.

The final part of the problem set teaches the user how to assess reliability of a design and investigate the sources of variability on the FE model. The categories are:

- Reliability
- Outlier Analysis

The run times for the examples vary between a few seconds (simple explicit problems) and ~10 min. (3GHz) (Iterative optimization using LS-DYNA for crash optimization). The longest DYNA simulation time is about 15s.

1.1 Software tools

The following tools are used in this tutorial:

- LS-OPT together with LS-OPT*ui* and Viewer (Requires Ver. 4.2 or later)

- LS-DYNA (Requires Ver. 971 *Revision R4.2.1* or later). Both the Single and Double Precision versions are required.
- LS-PrePost (included in the LS-OPT distribution)

1.2 General procedure for problem solution

1. Run the command file. This can be done using
   - LS-OPTui or
   - batch mode.
2. Use the Viewer to interpret the results. Guidelines for result interpretation are given at the end of this chapter.

1.3 Using LS-OPTui for input

The graphical user interface LS-OPTui can be used to prepare a command file for LS-OPT. The interface allows definition of the pre-processing and simulation tools, formulation of the design problem, definition of the LS-DYNA response variables and monitoring and control of the analysis runs.

LS-OPTui allows the creation of a command file as well as to read an existing command file.

1.4 Standard files contained in all directories

Some standard files are commonly used for directory management:

- **`cl`**: Script file to manually clear the **work** directory of all files except the necessary ones.
- **`clean`** Script file executed by LS-OPT to automatically clear the **run** directories after each iteration and while LS-OPT is running. The user must use UNIX commands, e.g.:

  ```
  rm –rf d3plot*
  ```

  The baseline and optimum results are not cleared.
- **`com.xxxx`** Command input file for LS-OPT. To be created or modified by the user. The tool LS-OPTui can be used.

1.5 Output files

`lsopt_input`	Echo of the input data. (View→Input)
`lsopt_output`	Log of all the internal steps taken to solve each example. The interim and final results are given in this file. (View→Output)
`lsopt_report`	A short summary of the results. Optimization problems only. (View→Summary Report)

1.6 Execution of LS-OPT

Graphical User Interface

Type: **`lsoptui com.xxx`**

Batch mode

Type: **`lsopt com.xxx`**

where **`com.xxx`** is the name of the command file.

Remarks:

- A typical restart only requires the selection of **Run** in the **Run** window or the **`lsopt <command_file_name>`** to be typed. If many or radical changes have been made to the input file, when a run is repeated, inconsistencies may develop and incorrect results may be computed. In this case, restart the run by clearing the work directory using the "Clean Start from Iteration" option in the GUI. A "Clean" feature is also available in the **Tools** task.

- LS-OPTui features a **Repair** option that enables the user to make changes in the design data and rerun the individual steps. The options can be viewed by selecting the "Repair" task.

1.7 Using LS-OPTui for post-processing

Type `viewer com` in the working directory or select the **Viewer** button from the GUI.

1.8 Result interpretation

1. **Simulation Statistics**
   a. **Correlation Matrix**
      Matrix of correlation values, scatter plots and histograms of all variables and simulation results.
   b. **Scatter Plots**
      2D or 3D scatter plots of simulation results.
   c. **Parallel Coordinate Plot**
   d. **Histories**
      Show time history curves of simulations as well as file-based curves.
   e. **Statistical Tools**
      Interactive tools for histograms, mean, standard deviation and probability of exceeding constraints
   f. **Correlation Bars**
      Correlation bar charts
2. **Metamodel Plots**
   a. **Surface Plots**
      i. Rotatable metamodel surfaces with isolines, constraint contours, feasible regions.
      ii. Points with point list, links to value spreadsheet and LS-PrePost.
      iii. Constraints and feasible regions.
   b. 2D Interpolator
      Matrix of 2D metamodel surface plots
   c. **Metamodeling Accuracy**
      i. The comparison between the predicted and computed results.
      ii. **Errors.** Metamodeling error measures.
   d. **Sensitivity**
      The magnitude and sign of the individual variable sensitivities and the associated 90% confidence interval of their significance can be visualized. Global sensitivity analysis.
   e. Virtual Histories
      Predicted history anywhere in the design space
3. **Optimization**

   a. **Optimization History**
      i. The optimization **history** of the variables and responses. Computed vs. predicted responses, composites, constraints and objectives.
      ii. The RMS/Max. error, R^2 history of the responses.
      iii. The upper and lower bound (**move limit** or **sub-region bound**) history of the variables.
      iv. Movement of the design variables relative to the bounds of the design space.

   b. Variables

   Optimal variable values. Confidence intervals for parameter identification.

4. **Pareto Optimal Solutions**

   a. Tradeoff

   2D or 3D scatter plots to visualize the Pareto-optimal front for multi-objective problems. 4D visualization is enabled using colors.

   b. **Parallel Coordinates**

   Exploration and elimination of optimal designs from the Pareto set by interactively moving constraints

   c. **Hyper-Radial Visualization**

   The exploration of Pareto Optimal designs by interactively adjusting the importance of each objective.

   d. **Self-Organizing Maps**

Postprocessing of simulations using LS-PREPOST: Click on or near points on graphs in LS-OPTui. The variable data will be presented in tabular form. LS-PrePost can be selected to post-process a selected design

1. **Viewing printed results:** Use **View** *Command File*, *Input*, *Output* or *Summary Report* (top left). The Summary Report is the most convenient way of viewing details of results in tabular form.

1.9 Reference documents

The principal reference documents are:

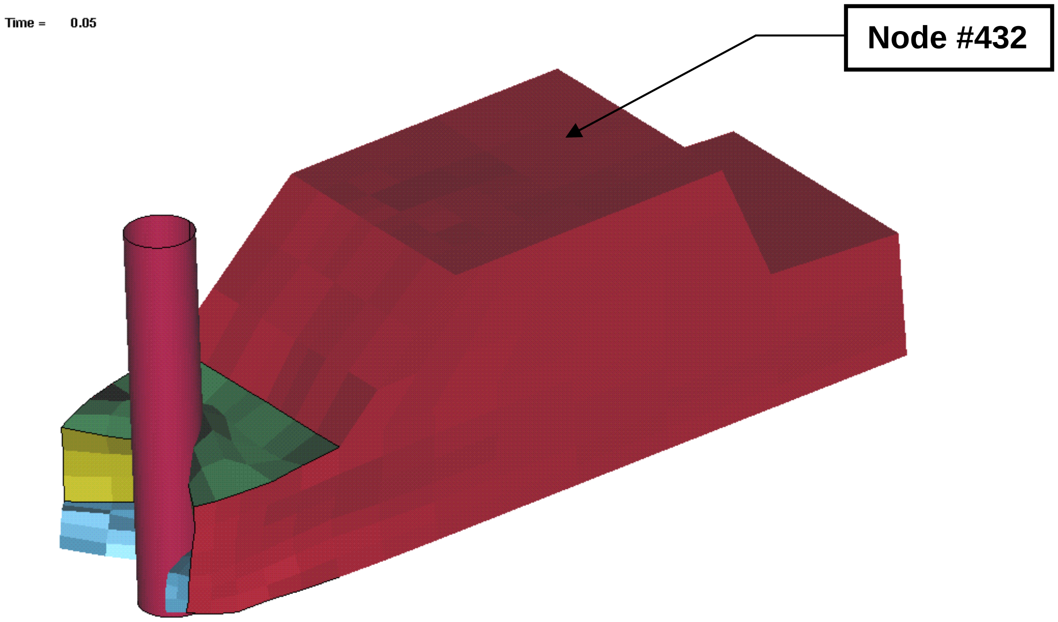
1. **LS-OPT User's Manual, Version 4.2,** *Livermore Software Technology Corporation, Livermore, CA, May 2011.* (access by selecting “Help” in the GUI)

2. **Class Presentation notes,** *Livermore Software Technology Corporation, Livermore, CA, May 2010.*

SIMPLE OPTIMIZATION AND VIEWING RESULTS

Problem description

The problem is of a simplified vehicle moving at a constant velocity and crashing into a pole. The figures show the deformed vehicle after 50ms and the part numbers.



Objectives of this example

The problem illustrates the following features:

- Formulating the optimization problem
- Viewer functionality. Interpretation of results.

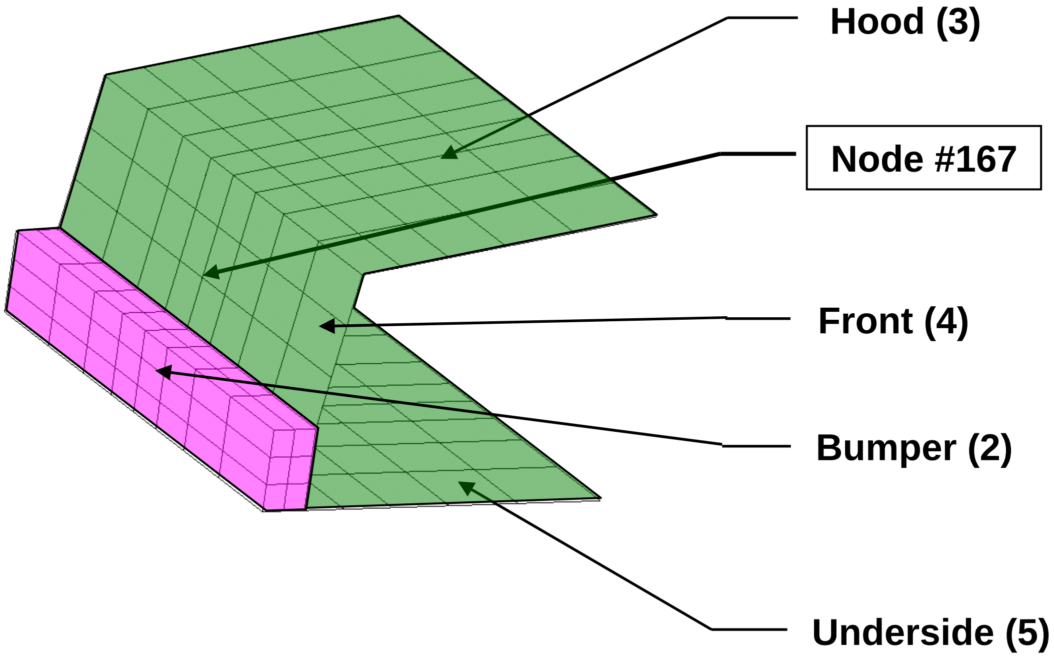
Design criteria

The criteria of interest are the following:

- Head injury criterion (HIC) of a selected point (15ms)
- Component Mass of the structural components (bumper, front, hood and underside)
- Intrusion computed using the relative motion of two points

Units are in *mm* and *sec*.

Design variables



The design variables are the following gauges:

- Hood, front and underside **`(thood)`**
- Bumper **`(tbumper)`**

Note: Parts 3, 4 and 5 are grouped under a single design variable **`thood`** while part 2 is identified as **`tbumper.`**

Post-processing study using the Viewer

Directory name: **`DESIGN_OPTIMIZATION/SIMPLE/VIEWER`**

Starting file: **`com.simple.start`**

The files in the directory have the following meanings:

`main.k`	Main (root) file with parameter specification
`car5.k`	Include file specified in `main.k`
`rigid2`	Include file specified in **`car5.k`**
`com.simple.correct`	Final design command file (modified using the .start file)

Design formulation

The design formulation is as follows:

Minimize

HIC (15ms) at node **432**

subject to

Intrusion (50ms) < 550mm

The intrusion is measured as the distance between nodes **167** and **432**.

Setup:

1. Open the file `com.simple.start` using the GUI.
2. Select View→File in the menu bar to view the following input files:
    a. View the parameter (`*PARAMETER`) definition in the file *main.k*. Note the inclusion of the file *car5.k*.
    b. View the parameter use (e.g. `&thood`) in the file *car5.k*.
3. Inspect each of the pages in the GUI to see how the input was specified. Confirm that you will be running a Single Stage optimization with 20 design points.
4. Now, formulate the design problem as follows:

   a. Select `HIC` as the objective function to minimize by selecting "HIC" in the **Objectives** tab.
   b. In the **Constraints** tab, select `Intrusion` as the constraint function and set the upper bound to 550mm.
5. This option runs one iteration.
6. Run the example by selecting **Run**.

Exercises:

Open the Viewer and select the following plot options:

1. **Scatter plot:**
   1.1 Find the baseline design (Design 1.1) and display the Finite Element mesh using LS-PrePost. Find nodes 432 and 167.
   1.2 Verify that there are 5 infeasible designs.
   1.3 Select the infeasible points to display them in a table. Verify in the table that they all have constraint violations.
   1.4 Select Intrusion as the fourth (or color) variable.
   1.5 Select "No entity" on the *z*-axis in order to view the sampling scheme arrangement of the points (Space Filling).
2. **Parallel Coordinate Plot:**
   2.1 Display the Variables, Constraints and Objectives.
   2.2 Find a feasible point with the lowest HIC value. Verify that tbumper = 5, thood = 2.1, Intrusion = 532, HIC = 194 (approximately).
   2.3 Find a feasible point with the lowest Intrusion value. Verify that tbumper = 5, thood = 5, Intrusion = 455 and HIC = 395 (approx.).
   2.4 Slide the upper bound of the Intrusion to 520mm and find the feasible point with the lowest HIC value. Verify that tbumper = 5, thood = 3.15, Intrusion = 505 and HIC = 282.
3. **Histories**
   3.1 Display all the Acceleration histories and determine which ones represent feasible designs.
   3.2 Display the Acceleration histories for designs 11, 16 and 20 (Select under the Options tab of the History plot display).
4. **Metamodel Surface**
   4.1 Display the metamodel surface for the HIC function. Include all the simulation points.

4.2 Select *Isolines*, *Constraints* and *Predicted Value* for the Optimum. Switch off the *Gridline* display. Use the Ctrl+Left Button to rotate the 3D display.

4.3 Project all the points to the surface and select the *XY* view button to display a top view of the plot.

5. **2D Interpolator**

5.1 Select both variables as well as HIC, Mass and Intrusion.

5.2 Select: *Constraints*, *Predicted value*, *Transpose* and *Link ranges col/row*.

5.3 Check whether the design (tbumper = 4, thood = 1) is predicted to be feasible. Verify that the predicted Intrusion value is approximately 575.

6. **Metamodel Accuracy**

6.1 Study and compare the PRESS values for HIC, Mass, Disp2 and Disp1.

7. **Sensitivities**

7.1 Using Linear ANOVA, which variable is the most important for influencing the HIC value?

7.2 Is it possible to state with confidence that thood is more important for Disp1 than tbumper?

7.3 In the GSA/Sobol plot, display the sensitivities for all the responses. Which variable seems to be the most important over all.

7.4 Display the Transpose and observe which response is the most sensitive to thood.

8. **Predicted histories**

8.1 Select the acceleration history and view its thood-sensitivity.

8.2 Plot the predicted acceleration at thood = 2.1, tbumper = 3.

8.3 In the Options tab, plot the design nearest to the predicted Acceleration history. Which design point does it represent.

8.4 Plot the maximum residual for the predicted Acceleration history.

8.5 Plot the maximum residual for the predicted Disp1 history.

9. **Optimization History**

9.1 View the HIC and Intrusion optimization histories on the same plot by selecting the "Split Vertical" icon on the menu bar.

9.2 Display the table with both the baseline and optimum values by clicking near Iteration 0 and then using the Shift key to also select Iteration 1.

10. **Integrated plotting**

10.1 Plot both the Parallel Coordinate Plot and the Scatter Plot in the same window by using the vertical split function. Select an infeasible point on the scatter plot. Then select + (for adding points) on the popup table and add all the other infeasible points by clicking the red points on the scatter plot.

11. **File viewing**

11.1 Return to the GUI and view the Summary Report using the "View" selection on the menu bar.

SETTING UP AN OPTIMIZATION RUN

Directory name: **`DESIGN_OPTIMIZATION/SIMPLE/SINGLESTAGE`**

The files in the directory have the following meanings:

`com.singlestage.correct`	Design command file (to be used for checking if needed).
`main.k`	Main (root) file with parameter specification
`car5.k`	Include file specified in **`main.k`**
`rigid2`	Include file specified in **`car5.k`**

Design formulation

The design formulation is as follows:

Minimize

HIC (15ms) at node **432**

subject to

Intrusion (50ms) < 550mm

The intrusion is measured as the distance between nodes **167** and **432**.

Basic setup

1. Confirm the that **`main.k`** has the following keyword information:

```
*PARAMETER
rtbumper,3.0,rthood,1.0
```

2. Confirm that, in **`car5.k`,** the element thicknesses are labeled, e.g. the **`thood`** parameter is labeled as follows:

```
*SECTION_SHELL
2,2,0.0,0.,0.
0.,0.,0.
&thood,&thood,&thood,&thood
```

tbumper is defined in a similar fashion.

The values of the variables in the ***PARAMETER** statement will be substituted during the optimization process.

3. Using LS-OPT*ui*, Develop the command file using the following steps:

Step	Remark
Strategy: Choose the Single Stage	
Solvers: Specify the dyna executable **ls971_single** and the input file **main.k**	
Variables: Switch the constants to variables. Define the size of the design space as [1,5] for each of the thickness variables.	Omit the initial range so that the optimization starts from the full design space.
Sampling: The approximation is set to RBF network.	This selection is the default. Use the default point selection scheme (*Space Filling*). Use the default number of points (= 10).
Responses: Define the responses. The **Intrusion** constraint consists of the difference between the *x*-displacement of node 432 (**Disp2**) and node 167 (**Disp1**). The HIC response is evaluated at node 432.	A standard or expression composite is used to define the intrusion.
Add the **Mass** of parts 2,3,4 and 5	
Objectives: Select the objective function.	
Constraints: Select the constraint and set the bounds	The "Strict" and "Move" options can be ignored.
Select to "Omit Last Verification Run" and **Run** the problem. Save the data to DESIGN_OPTIMIZATION/SIMPLE/ SINGLESTAGE.	

Exercises:

1. **The accuracy of the responses.** Study the approximation error indicators and the plots of the computed vs. the predicted results (**Accuracy** plots).

   1.1 Fill in the approximation errors of the results. These quantities can be found in the **Accuracy** plot.

	Starting Value	Sqrt PRESS %	R^2
Mass	0.41	0	1
Disp1	-161	2.6	.2
Disp2	-737	.9	.99
HIC	68	43	.8

   1.2 Using the two quantities RMS error and R^2, what conclusions, if any, can you make for each approximation about the level of 'noise' or modeling error?

2. Study the Sensitivity charts (linear ANOVA).
   2.1 Which variable appears to be the most important?
   2.2 How are the values (of the main bar) in the plot derived?

3. Study the Point selection scheme.
   3.1 Use the **Scatter Plot** option and switch to *2D*.
   3.2 Select the point status as *Feasibility*.
   3.3 How many infeasible designs are there in the design set?

4. **Verification run:** Unset "Omit Last Verification Run" (*Run* page) and **Run**. A single run will be done as part of the *2nd* iteration (directory 2.1) to verify the predicted optimum.

   - Obtain the starting and final results of the optimization run from the Optimization History plot by clicking near Iteration 0 and Iteration 1 respectively.

- A spread sheet format of the computed results of both the starting and optimum points can be obtained by selecting both (use Ctrl or Shift while clicking in the Point List at the extreme left column)

Study

4.1 The change in each of the variables and responses

4.2 The accuracy of the starting point and the optimum point after the first iteration. These results are tabulated below.

	Start		**Optimum**	
	Comp.	Approx.	Comp.	Approx.
tbumper	3		5	
thood	1		1.7	
Mass	0.41	0.41	0.6	0.6
Intrusion	576	572	546	550
HIC	68	108	133	87
Max. Constr. violation	26	23	0	0

5. **Design Sensitivities:**

5.1 Study the final design sensitivities in **View→Summary Report.** Confirm the estimated change in each of the following quantities for a **0.1** mm change in the Hood thickness and Bumper thickness respectively.

	Hood	Bumper
Mass	0.02	0.006
Intrusion	-3	.4
HIC	+10	-4.5

Add more points using the *Repair* feature

Directory: **`DESIGN_OPTIMIZATION/SIMPLE/ADD_POINTS`**

Starting file: **`com.add.start`**

The purpose of the example is to improve the accuracy of the design model. The design formulation is the same as in the previous section.

Setup:

1. First do a Single Stage run using **`com.add.start`**, exactly as before. Confirm that the number of sampling points = 10 and that the last verification run is omitted (Run tab). Then make the following changes:

2. **Sampling** panel:

   2.1 Make sure that RBF and Space Filling are selected.
   2.2 Change the number of simulation points to 20.

3. Select the **Repair** task to *Add Metamodel Points* and *Run*. A sampling table is created for 10 additional runs and the old metamodel is deleted.

4. Set the task to metamodel-based optimization. Do *not* use a "Clean start", but include the verification run and Run the problem. 10 Additional runs will now be done.

	Start		**Optimum**	
	Comp.	Approx.	Comp.	Approx.
tbumper	`3`		`3`	
thood	`1`		`1.68`	
Mass	`0.41`	`0.41`	`0.56`	`0.56`
Intrusion	`576`	`575`	`550`	`550`
HIC	`68`	`39`	`179`	`147`
Max. Constr. violation	`26`	`25`	`0`	`0`

4.1 Comment on the accuracy for 20 points vs. the original 10 points.

4.2 View the feasible region by using the *Constraints* option in the Metamodel surface plot.

Add a constraint

Directory: **`DESIGN_OPTIMIZATION/SIMPLE/ADD_POINTS`** (same as previous)

Starting file: **`com.add.correct`**

The purpose of the example is to add a constraint to the design. You should have a database of 20 runs generated previously.

Setup:

1. Add a Mass upper bound constraint of 0.5

   1.1 Repair the optimum using the Optimize repair option.

   1.2 View the feasible region by using the *Constraints* option in the Metamodel surface plot. What do you observe in terms of the influence of the mass constraint?

   1.3 Switch to a Sequential Strategy, do a clean start from Iteration **2** and run a metamodel-based optimization to produce a *new* verification run. Tabulate the results:

	Start		**Optimum**	
	Comp.	Approx.	Comp.	Approx.
tbumper	3		1.24	
thood	1		1.87	
Mass	0.41	0.41	0.5	0.5
Intrusion	576	575	544	550
HIC	68	39	204	252
Max. Constr. violation	26	25	0	0

Solving the optimization problem with a discrete variable

Directory: `DESIGN_OPTIMIZATION/SIMPLE/DISCRETE`

Starting file: **`com.discrete.start`**

Setup:

1. Modify the thood variable so that it has to be selected from the set {1, 2, 3, 4, 5}. This is done by changing the variable type from "Variable" to "Discrete variable" and filling in the text box with **`1 2 3 4 5`**. Use **`1`** as a starting variable.
2. Set the *Sampling Type* to **Discrete**.
3. On the **Sampling** page, select 20 points.
4. Conduct a single stage run.

Exercise:

1. Compare the predicted results of the continuous and discrete analyses.

	Continuous	Discrete
tbumper	1.24	1.28
thood	1.87	2
HIC	252	266
Mass	0.5	0.53
Intrusion	550	550

Comment on the feasibility of the solution.

2. Study a tbumper-thood plot of the surface and points to get an impression of the *discrete space filling* point selection scheme. Are there any feasible points amongst the sampling points?
3. In the metamodel surface plot, select the Constraints option to explain the infeasible result.

Automated iterative process using a linear approximation

Directory: **`DESIGN_OPTIMIZATION/SIMPLE/ITERATE`**

Starting file**: `com.iterate.start`**

Setup:

1. On the **Strategy** page, select the SRSM strategy. Select a tolerance of 0.01 to be satisfied by either the design or objective changes.
2. Use the **Sampling** page to confirm that the polynomials and *D*-Optimality criterion have been selected. *Select 5 points per iteration*.
3. Select the Hybrid ASA as core optimizer algorithm.
4. Select 10 iterations and run the problem.

Exercises:

1. **Convergence.** Study the **Optimization History** of the variables and responses.
   1.1 What happens to the move limits (= region of interest) (blue lines) of **`thood`** and **`tbumper`**?
   1.2 What are the trends in **`thood`** and **`tbumper`**?
   1.3 Comment on the convergence behavior of the HIC response. Does it seem converged?
   1.4 Is this also true for the Intrusion and Mass?
   1.5 What is the mass trend during optimization?
   1.6 Compare the optimal result to the result obtained using 20 points for a single stage run.

	***Single Stage* (20 simulations)**		***SRSM* (46 simulations)**	
	Comp.	Approx.	Comp.	Approx.
tbumper	`1.24`		`1.59`	
thood	`1.87`		`1.77`	
Mass	`0.5`	`0.5`	`0.5`	`0.5`
Intrusion	`544`	`550`	`551`	`550`
HIC	`204`	`252`	`236`	`233`

Max. Constr. violation	`0`	`0`	`0.07`	`0`

2. **Accuracy**.

    2.1 Study the **Optimization History**. Comment on the accuracy of the HIC response (on the **Value** plot of HIC).
    2.2 What is the RMS error trend of HIC?
    2.3 What is the R^2 trend of HIC?
    2.4 In the Optimization History, observe the accuracy of the **`Disp2`** response.

3. Would it be possible to repair an optimization result by adding or modifying a constraint in this iterative run (i.e. without rerunning any simulations)? Why/why not?

4. Comment on the differences between single stage and iterative solutions as far as utility and accuracy is concerned.

OPTIMIZATION STRATEGIES

Directory: **`DESIGN_OPTIMIZATION/OPTSTRATEGIES`**

This example uses the same FE model as the crashworthiness problem. The purpose is now to investigate the difference in accuracy between the various optimization strategies. Three strategies are chosen, namely (i) single stage, (ii) sequential and (iii) sequential with domain reduction. Refer to the Manual Section 20.6 (Setting parameters for metamodel-based optimization strategies) for more details.

This example has the following features:

- The use of Radial Basis Function networks as metamodel.
- The display of constraints and feasible regions.
- The investigation of different optimization strategies.

Single stage

Directory:
`DESIGN_OPTIMIZATION/OPTSTRATEGIES/SINGLESTAGE`

Starting file: **`com.singlestage.start`**

Setting up the problem

1. Change the strategy to **Single Stage**.
2. Make sure the metamodel is set to Radial Basis Function networks and point selection to Space Filling. These are the defaults for Single Stage.
3. Select 25 points.
4. Run.

Exercise

1. In the Scatter plot (2D selection), view thood vs. tbumper. Are there any boundary points?

2. Record the *predicted* optimum value and PRESS error as follows:

Name	Value	Square root PRESS error (%)
`tbumper`	3.4	-
`thood`	1.6	-
`Disp2`	-711	0.65
`Mass`	0.58	0.008
`HIC`	151	32.
`Acc_max`	2.3e6	2.8

   The error measures can be obtained by selecting the **View→Summary Report** option or by studying the headers of the relevant "Accuracy" plots.

3. What is the meaning of the PRESS error?

4. Study the HIC metamodel plot (the objective function in this case). Select the Setup→Constraints option. Also select isolines instead of gridlines. Plot the points for this option.
   a. Which is the feasible region
   b. Which is the infeasible region
   c. Determine by inspection of the plot where the optimum solution must be. Select the *Predicted value* option in **Setup** to plot the optimum to check your answer.

Sequential strategy

Directory: **`DESIGN_OPTIMIZATION/OPTSTRATEGIES/SEQUENTIAL`**

Starting file: **`com.sequential.start`**

Setting up the problem

1. Change the strategy to **Sequential**.
2. Check that the sampling type is Radial Basis Function Network.
3. Use the selection for "First iteration Linear, D-Optimal" (default).
4. Check that the point selection scheme is Space Filling.
5. Use the default number of points.
6. Adjust the iteration limit in the GUI Run panel to 5 and Run.

Exercise

1. In the Scatter plot (2D selection), view `thood` vs. `tbumper`.
   a. Are there any boundary points?
   b. Set the Point Status to "Iterations" to determine at which iteration the boundary points are created.
   c. Explain how boundary points are created.

2. Record the *predicted* optimum value and PRESS error as follows:

Name	Value	Square root PRESS error (%)
`tbumper`	5	-
`thood`	1.6	-
`Disp2`	-713	0.48
`Mass`	0.67	0.008
`HIC`	108	33
`Acc_max`	2.54e6	3

The error measures can be obtained by selecting the View→Summary Report option or by studying the headers of the "Accuracy" plots.

Compare the PRESS errors to the PRESS errors of the single stage optimization run.

Sequential strategy with domain reduction

Directory:
`DESIGN_OPTIMIZATION/OPTSTRATEGIES/`
`DOMAIN_REDUCTION`

Starting file: **`com.sequential_reduction.start`**

Setting up the problem

1. Select the Sequential with Domain Reduction strategy.
2. Change the sampling type to Radial Basis Function Network.
3. Use the default number of points.
4. Use the selection for "First iteration linear, D-Optimal" (default).
5. Confirm that the point selection scheme is Space Filling.
6. Adjust the iteration limit in the GUI Run panel to 5 and Run.

Exercise

1. In the Scatter plot (2D selection), view thood vs. tbumper. How are these points distributed?

2. Record the predicted optimum value and PRESS error as follows:

Name	Value	Square root PRESS error (%)
`tbumper`	5	-
`thood`	1.64	-
`Disp2`	-713	0.69
`Mass`	0.68	0.007
`HIC`	138	29
`Acc_max`	2.55e6	9

The error measures can be obtained by selecting the View→Summary Report option or by studying the headers of the "Accuracy" plots.

Compare the PRESS errors to the PRESS errors of the *Single Stage* and previous *Sequential* optimization runs.

IMPORTING ANALYSIS RESULTS

The purpose of this example is to import a user-defined table of results into the GUI and to enable an optimization to be conducted using these results. A description of the steps can also be found in the User's Manual Section 17.9.

Directory:
`DESIGN_OPTIMIZATION/IMPORT_RESULTS`

Starting file: A text file with comma separated variables **`AnalysisResults.csv`**. There is no input command file.

Setting up the problem

The steps for importing user-defined analysis result files using the GUI are as follows:

1. Add a second header line to the given "`AnalysisResults.csv`" file using "`dv`" for design variables *tbumper* and *thood* and "`rs`" for responses *Disp2*, *Disp1*, *Acc_max*, *Mass* and *HIC*. This header line is just below the *name* header.

   Start `lsoptui:`

2. *Strategy panel:* Choose Single Stage.

3. *Solvers panel:* Browse for the "`AnalysisResults.csv`" file using the "Import user results" tab.

4. Specify a name for the analysis case and "Add" the case.

5. *Variables and Responses panels.* It is required to check the variables and responses in these panels. The variables and responses will be displayed and automatically associated with the correct analysis case.

6. Adjust the variable bounds to [1,5].

7. *Sampling panel*. Select the metamodel as RBF network with Space Filling.

8. *Menu bar*. Choose the "Repair" task.

9. *Run panel.* Select "Import results (.csv)" and "Run".

10. View the Summary Report.

11. Switch to Metamodel-based optimization.

12. Optimization:
    a. Define HIC as the objective.
    b. Use a Composite-Expression to define the constraint:

$$\mathit{Intrusion} = \mathit{Disp1} - \mathit{Disp2} < 550\text{mm}.$$

    c. In the Run panel, select "Omit last Verification Run" and then "Run". An optimization history is created.

Exercise

1. Display the simulation design points on the Parallel Coordinate plot:
    a. What is the HIC value of the design with the smallest intrusion?
    b. What are the HIC and Intrusion values for the design with the lowest Max. Acceleration?
2. Compare the predicted optimization result with the result of the previous single stage optimization.

SIMPLE MULTI-OBJECTIVE OPTIMIZATION

This example illustrates two approaches to the computation of the Pareto Optimal Frontier. The first approach uses *direct* optimization while the second approach is *metamodel*-based. The example uses the finite element model of the vehicle impacting a pole as was used for single objective optimization (see page 11).

Pole crash problem using Direct GA

Minimize Mass
Minimize Intrusion (50ms)

subject to

HIC (15ms) < 250

The parameters are the same two variables as before, namely `tbumper` and `thood`.

Special topics: Multi-objective optimization, Pareto optimality

Directory: **`DESIGN_OPTIMIZATION/MULTIOBJECTIVE/SIMPLE/DIRECT`**

Starting file: **`com.moo.direct.start`**

Setting up the problem

1. Select the task as *GA Direct Simulation* and select as Strategy to create the Pareto Optimal Frontier.
2. Select Mass and Intrusion as objectives for defining a two-objective optimization problem. De-select all other responses.
3. Set a HIC upper bound constraint of 250. De-select all other constraints.
4. On the **Algorithms** page, set the population size to 20 and the number of generations to 20.
5. Run.

Exercise

1. Display the Tradeoff plot for iteration 20.
2. How is the Utopian point defined? Determine the objective function values of the Utopian point.

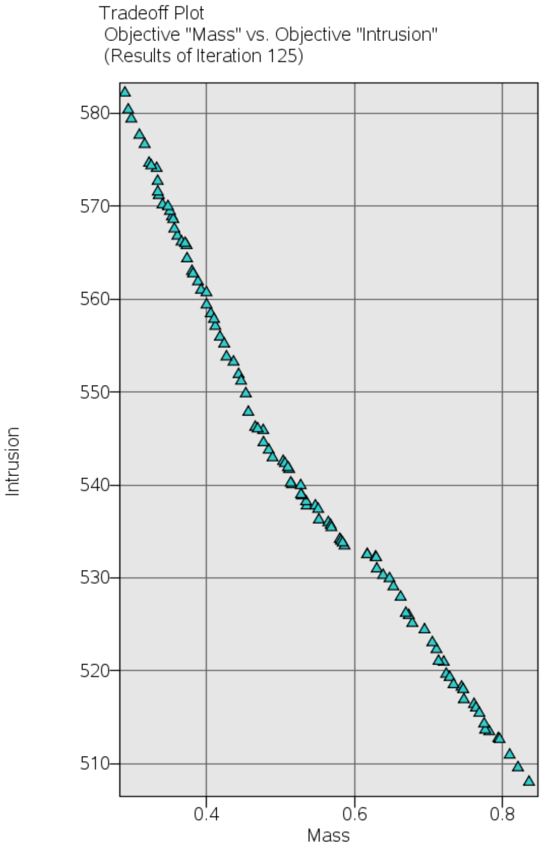
Intrusion	*Mass*
525	0.29

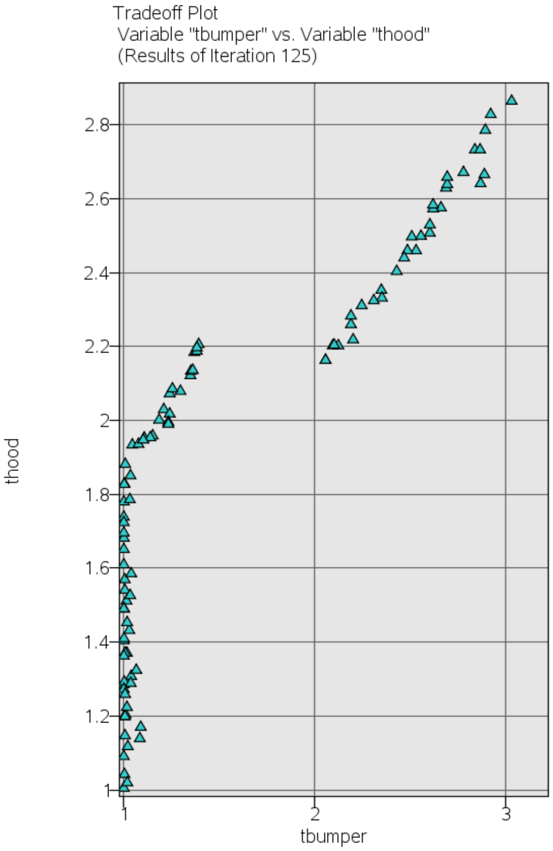
3. Add the Parallel Coordinate plot by splitting the screen vertically. Check "Select from active points" on the Parallel Coordinate plot and slide the upper bounds of the *Mass* and *Intrusion* functions down to about 0.65 and 560 resp. leaving a limited number of Pareto candidate points. What do you observe on the Tradeoff plot?
4. Add the SOM plot by splitting the screen again and choose a cell representing a set of designs with the lowest mass. Observe the Parallel Coordinate plot and Tradeoff plot as well as the table representing the point data. Also click on some of the adjacent cells on the SOM plot and observe the parallel plot. *Note:* (i) Points can be selected with '=', added with '+' or subtracted with '-'. (ii) "Active" points can be totally unset by selecting the '-' button on the table and then "rubberbanding" the entire set of points in the Tradeoff plot.
5. *HRV:* Select the HRV plot. Unselect "Scale weights". Slide the weights of the objective functions to [0;1] and [1;0] and observe what happens to the Pareto Front and the point closest to the Utopian point (colored in purple).
6. Document the point closest to the Utopian point for the following weight selections. *Hint:* Use the text box option in the HRV control window to set the weights.

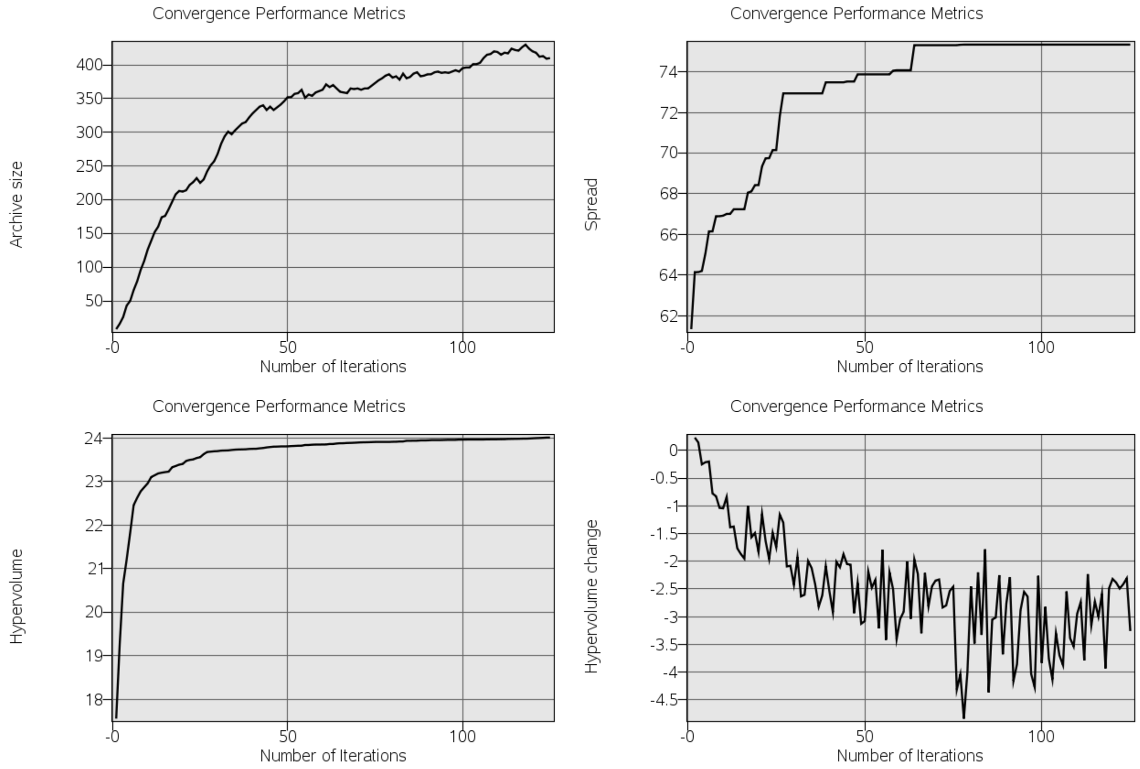
w_{Mass}	$w_{Intrusion}$	*tbumper*	*thood*	*Mass*	*Intrusion*
0	1	4.7	2.2	.80	525
1	1	1.0	1.9	.48	545
1	0	1.0	1.0	.29	583

7. Using the Optimization History (MOO Performance Metrics) feature, tabulate the following:
    a. Archive size at 10 iterations
    b. Archive size at 20 iterations
    c. The Dominated Hypervolume of the Pareto Frontier at 20 iterations.
    d. What happens to the Spread of the Pareto Frontier during the optimization?

“Exact” optimization result using a population of 50 for 125 generations






Convergence Performance Metrics
Archive size
400
350
300
250
200
150
100
50
-0
50
100
Number of Iterations
Convergence Performance Metrics
Spread
74
72
70
68
66
64
62
-0
50
100
Number of Iterations
Convergence Performance Metrics
Hypervolume
24
23
22
21
20
19
18
-0
50
100
Number of Iterations
Convergence Performance Metrics
Hypervolume change
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-0
50
100
Number of Iterations

Pole crash problem using Metamodel-Based optimization

Directory:
DESIGN_OPTIMIZATION/MULTIOBJECTIVE/SIMPLE/METAMODEL

Starting file: **`com.moo.metamodel.start`**

Setting up the problem

1. Select the task as *Metamodel-based Optimization*.
2. *Strategy*. Pareto Optimal Frontier using Sequential Optimization.
3. *Sampling*. Confirm that RBF and Space Filling are selected.
4. *Objectives*. Select Mass and Intrusion as objectives. De-select all other.
5. *Constraints*. Set a HIC upper bound constraint of 250.
6. *Run*. Set the iteration limit to 10 and Run.

Exercise

1. Display the Tradeoff plot for iteration 10.
2. Determine the objective function values of the Utopian point.

Intrusion	*Mass*
525	0.29

3. Add the Pareto Parallel Coordinate plot by splitting the screen vertically. Check "Select from active points" on the Parallel Coordinate plot and slide the upper bounds of the *Mass* and *Intrusion* functions down to about 0.65 and 560 resp. leaving a limited number of Pareto candidate points. Check that the HIC upper bound is set correctly at 250. What do you observe on the Tradeoff plot?
4. Add the Pareto SOM plot by splitting the screen again and choose a few cells representing a set of designs with the lowest mass. Observe the Parallel Coordinate plot and Tradeoff plot as well as the table representing the point data.
5. *HRV:* Select the HRV plot. Unselect "Scale weights". Slide the weights of the objective functions to [0;1] and [1;0] and observe what happens to the Pareto Front and the point closest to the Utopian point (colored in purple).

6. Document the point closest to the Utopian point for the following weight selections. *Hint:* Use the text box option in the HRV control window to set the weights.

w_{Mass}	$w_{Intrusion}$	*tbumper*	*thood*	*Mass*	*Intrusion*
0	1	5.0	2.4	.86	525
1	1	1.2	1.9	.51	545
1	0	1.0	1.0	.29	581

7. When comparing the direct with the metamodel-based optimization, what are the possible reasons for differences between the results?
8. Extend the metamodel-based optimization to 20 iterations (by changing the number on the run page) and compare the results again. You may have to set a lower tolerance for stopping, e.g. 0.0001.

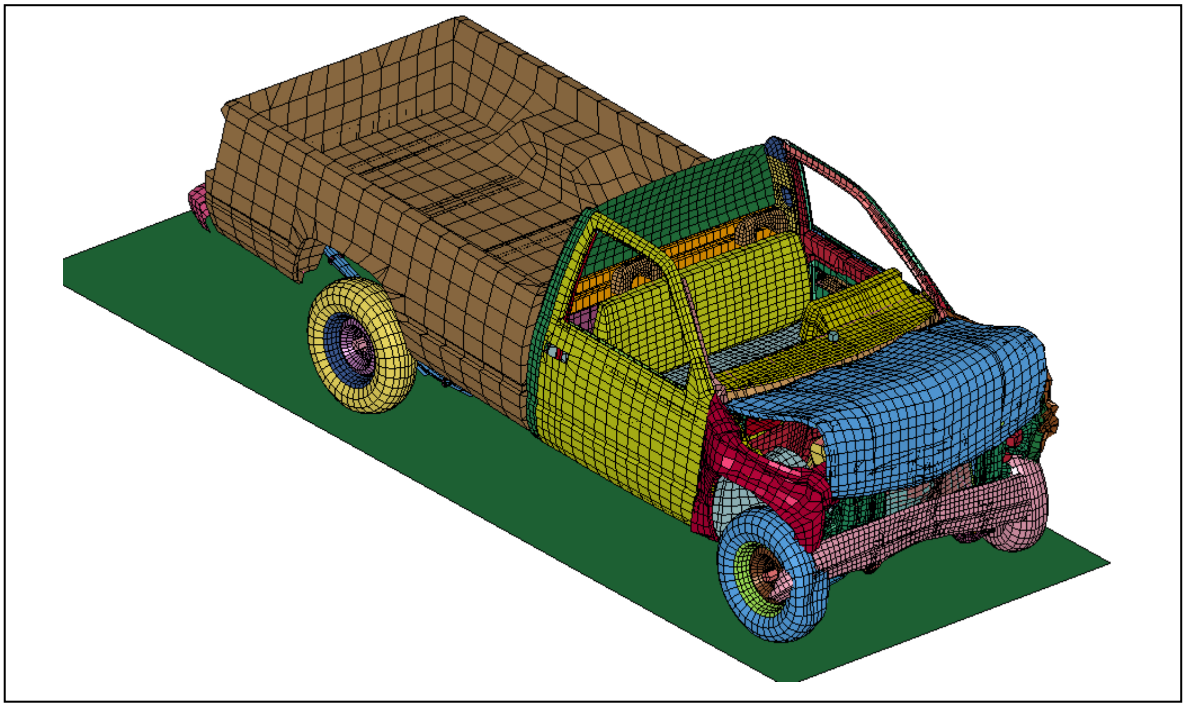
w_{Mass}	$w_{Intrusion}$	*tbumper*	*thood*	*Mass*	*Intrusion*
0	1	2.9	2.9	.83	512
1	1	1.0	1.9	.50	544
1	0	1.0	1.0	.29	582

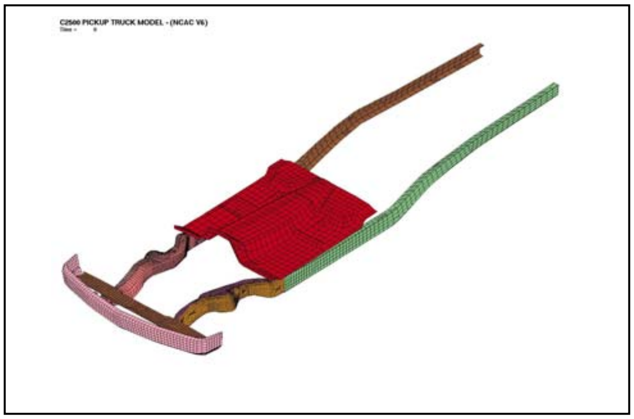
9. Using the Optimization History (MOO Performance Metrics) feature, tabulate the following:

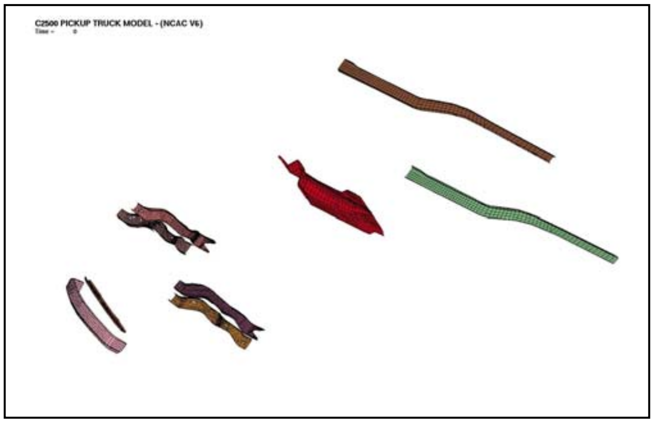
    a. Archive size at 20 iterations
    b. The Dominated Hypervolume of the Pareto Frontier at 20 iterations.
    c. What happens to the Spread of the Pareto Frontier during the optimization?

ADVANCED MULTI-OBJECTIVE OPTIMIZATION

This example demonstrates the steps for optimizing a design with multiple objective functions in the design formulation. Multiple objectives are not easily visualized and require special post-processing tools. The example is an automotive crashworthiness problem defined as follows. The exploded view shows the parts affecting the frontal crash performance. There are 9 design variables, each a thickness of a part. *Note*: *This example takes about 20 minutes to run, largely due to fitting the RBF networks to 1000 points.*







The design formulation is as follows:

Minimize

Mass and peak acceleration

Maximize

Time-to-zero-velocity and maximum displacement

subject to constraints on variables and performance:

Maximum displacement < 721mm
Stage 1 pulse < 7.48g
Stage 2 pulse < 20.20g
Stage 3 pulse < 24.50g

Thus there are four objectives, requiring advanced tools for viewing the Pareto Optimal Frontier.

Special topics: Multi-objective optimization in higher dimension, Pareto optimality

Directory: **DESIGN_OPTIMIZATION/MULTIOBJECTIVE/ADVANCED/DETERMINISTIC**

Starting files: **com.moo.start, UserResults.csv**

Setting up the problem

1. Select "Create Pareto Optimal Frontier" in the **Strategy** panel.
2. Study the input data as defined in the GUI
3. View the UserResults.csv file using **View→File** (this step not required to import the file).
4. Import the UserResults.csv file as follows:
    a. In the **Solvers** page, choose the option *Import User Results*.
    b. Browse for the .csv file.

   c. Save the solver data by using *Replace*.
   d. Select the **Repair** task.
   e. Select the *Import Results (.csv)* option and **Run**.
   f. Check the screen output and the Summary Report (*lsopt_report* file)

5. On the Sampling page, select *Compute Global Sensitivities*.
6. Switch the task to *Metamodel-based Optimization*.
7. In the Run tab, select to ‘omit the last verification run’ and run the problem.

<u>Viewer</u>

1. Using the *Optimization History* feature, record the predicted starting and optimum values of the objectives and constraints and trends for the four objectives and 4 constraints.

   How is this single optimum point calculated although there are multiple objectives?

   What do you know about the feasibility of the starting design?

	Starting	**Optimum**	**Trend**
Disp (max.)	703	721	↑ , ≤ 721
Accel (min.)	169,000	140,000	↓
Time_to_zero_vel (max.)	0.084	0.085	↑
Mass (min.)	1.812	1.789	↓
Stage1Pulse	*7.86*	7.13	≤ 7.48
Stage2Pulse	*21.3*	19.7	≤ 20.2
Stage3Pulse	*24.6*	24.4	≤ 24.5

2. Create *Tradeoff*, *Parallel Coordinate* and *HRV* plots in a multi-window environment. This can be done by splitting the windows in the plot selector (the first pop-up after selecting the Viewer) or in the menu bar.

   a. Search for a small set of good points from the Pareto Optimal set. This is done by moving the bounds of the objectives (actually composites) in the *parallel coordinate plot* as follows.

	Lower Bound	Upper Bound
Disp	720.8	-
Accel	-	133,000
Time_to_zero_vel	0.090	-

Write down the ranges of optimal design variables *t3*, *t4* and *t5* from the Parallel Coordinate plot. Select from the *reduced* point set and from the *active* (bounded) points only.

	Lower	Upper	Optimal point for equal weighting
t3	2.5	2.6	2.5
t4	2.8	2.9	2.8
t5	2.7	2.8	2.7

b. Select equal weights for the objectives in the HRV plot and select the best (purple colored) point (see last column in table above). What are the objective and constraint values for this point?

	Optimum (Optimization History plot)	**Optimum (Pareto Plot with equal weights)**	**Bound**
Disp (max.)	721	721	≤ 721
Accel (min.)	140,000	131,000	-
Time_to_zero_vel (max.)	0.085	0.092	-
Mass (min.)	1.79	1.79	-
Stage1Pulse	7.13	7.45	≤ 7.48
Stage2Pulse	19.7	19.5	≤ 20.2
Stage3Pulse	24.4	23.8	≤ 24.5

c. Use the *Pareto* parallel coordinate plot to predict the feasible design with the smallest mass. See center column below.

d. Use the parallel coordinate plot of the *simulation results* to find the feasible design, from all the simulations, with the smallest mass. Record the computed results (last column below).

	Starting	**(c) Min. Mass (Predicted)**	**Trend**	**(d) Min.Mass (Simulation)**
Disp (max.)	703	721	↑ , ≤ 721	719
Accel (min.)	169,000	140,000	↓	249,000
Time_to_zero_vel (max.)	0.084	0.092	↑	0.090
Mass (min.)	1.812	**1.804**	↓	1.815
Stage1Pulse	*7.86**	7.40	≤ 7.48	6.59
Stage2Pulse	*21.3**	19.2	≤ 20.2	19.9
Stage3Pulse	*24.6**	24.2	≤ 24.5	22.8

*violation

3. Use the Global Sensitivity Analysis to estimate the variables that contribute (i) the most and (ii) the least variation to the design.

USER-DEFINED OPTIMIZATION PROBLEM

Directory: **`DESIGN_OPTIMIZATION/USER_DEFINED`**

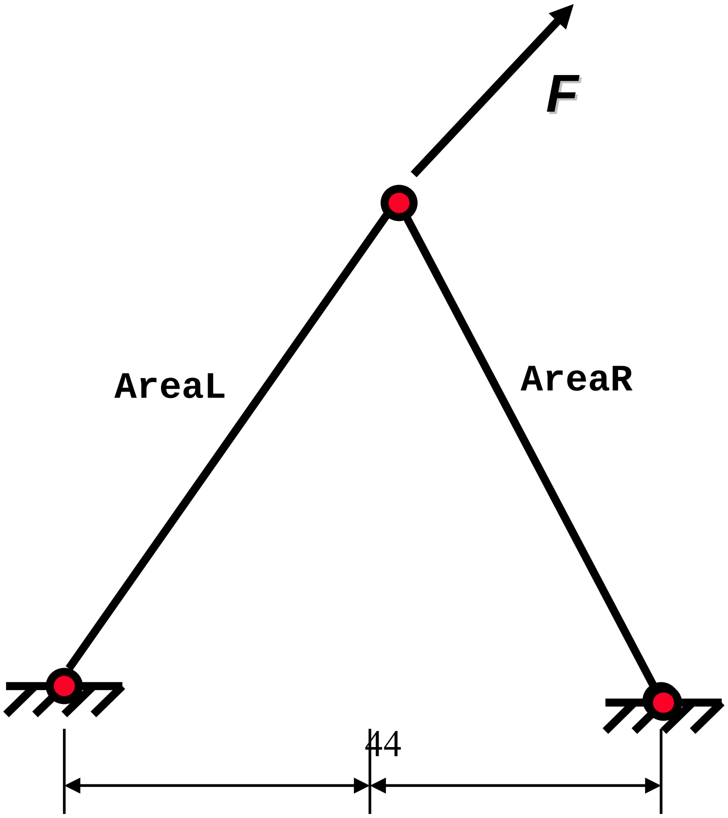
Special topics: LS-OPT parameter type, Neural nets, Pareto optimality, dependent variables

Problem description:

The problem is of a simple two-bar truss. A linear analysis is conducted using the user written program below. The height of the structure = 1. The force components are: Fx = +24.8kN, Fy=198.4kN.

The criteria are weight and stress. The stresses are limited to an absolute value of 1.0. Three design variables are chosen, namely the cross-sectional area of the bars and the base measurement between the supports. The baseline design has the following values:

	Baseline	Lower Bound	Upper Bound
Base	0.8	0.1	1.6
AreaL	2	0.2	4
AreaR	2	0.2	4



Base   Base

Analysis File: `2bar`

```
#!/usr/bin/perl
#
#  2BAR truss
#
#  Open output files
#     Each response is placed in its own file
#
   open(WEIGHT,">Weight");
   open(STRESSL,">StressL");
   open(STRESSR,">StressR");
#
#--Compute the responses
#
   $length = sqrt(1 + <<Base>>*<<Base>>);
   $cos  = <<Base>>/$length;
   $sin  = 1/$length;
   $Weight = (<<AreaL>> + <<AreaR>>) * sqrt(1 + <<Base>>*<<Base>>) / 2;
   $StressL = ( 24.8/$cos + 198.4/$sin)/<<AreaL>>/200;
   $StressR = (-24.8/$cos + 198.4/$sin)/<<AreaR>>/200;
#
   print WEIGHT $Weight,"\n";
   print STRESSL $StressL,"\n";
   print STRESSR $StressR,"\n";
#
#  Signal normal termination
#
   print "N o r m a l\n";
#
```

Note the labeling of the variables using the double angular brackets **`<<name>>.`** These will be replaced by numbers.

The purpose of the example is to illustrate the following:

- How to define a user-defined problem
- The definition of dependent variables in the user interface.
- The application of neural nets.

Sequential optimization with domain reduction

Directory: **`DESIGN_OPTIMIZATION/USER_DEFINED/LINEAR`**

Setup:

Create a command file as follows:

1. *Solver:* Select the **User-defined** solver option.
2. Enter the solver command as **`perl`**.
3. Browse for the input file name: **`2bar`**.
4. *Variables:* Enter the variable data. Select the lower and upper bounds and choose initial ranges of [Base:1.0; AreaL:2.0; AreaR:2.0].
5. *Responses*: Enter the responses as **USER-DEFINED**. The solver dumps the results into individual files: **`Weight, StressL, StressR`**, therefore the response command must write the value in the file to standard output, e.g. **`cat Weight`** (Linux) or **`type Weight`** (Windows).
6. *Objective*: Minimize the Weight.
7. *Constraints*: Bound the stresses from above and below as [-1;1] in both bars.

Exercises:

1. *Strategy:* Set the **Strategy** to *SRSM* and do a baseline run only.
2. *Run:* Report the stresses associated with the baseline design by using the Scatter Plot or Parallel Coordinate plot.

StressL	0.73
StressR	0.54
Weight	2.6

3. Run 5 linear iterations using a linear approximation and the *D*-Optimality criterion for point selection and tabulate the optimal variables and responses:

	Predicted	Computed
Base	0.33	-
AreaL	1.6	-
AreaR	0.58	-
StressL	1	0.92
StressR	1	1.1
Weight	1.14	1.13

Reducing the number of variables by constraining the bar areas

Starting file: **`com.userdef.constrained.start`**

Directory:
`DESIGN_OPTIMIZATION/USER_DEFINED/`
`CONSTRAINED_VARS_LINEAR`

Setup:

1. Choose the SRSM strategy.
2. Change **`AreaR`** to a variable which is dependent on **`AreaL`**, by clicking on the **Variable** button in the **Variables** window and using the formula **`AreaR = AreaL/2.0`**. (Just type in **`AreaL/2.0`** in the box opposite **Dependent**).
3. Remove the range values so that the problem starts running from the full design space.
4. Save the modified data under any name.
5. Run 5 iterations.

Exercises:

1. Tabulate the optimal variables.

	Computed	Predicted
AreaL	1.4	
AreaR	0.7	
Base	0.42	
StressL	0.97	1.0
StressR	1.0	0.95
Weight	1.17	1.19

2. Do you think the solution has converged?

Neural Net optimization

Starting file: `com.userdef.neural.start`

Directory: **`DESIGN_OPTIMIZATION/USER_DEFINED/NEURAL`**

Setup:

1. Using **`com.neural.start`** as a starting file, set the strategy to *Sequential*
2. Confirm that the range is *absent* in the **Variables** panel, i.e. you are using the full design space.
3. Change the metamodel selection from **RBF** to **Feedforward Neural Net (**Sampling panel). Use default settings. Make sure the first iteration is set to Linear, *D*-Optimal.
4. Run 5 iterations.

Exercises:

1. Use the **Metamodel Surface** selection in **Viewer** to study the metamodel.
   a. *Point selection*. Display a Base-AreaL XY plot. Select "Points only" with "Iterations" as status. Select "All Previous" under the "Iterations" heading. View the points for different iterations 1-5.
   b. *Point feasibility*. Study the feasibility of the points using the red/green color coding as well as studying the data list.
   c. *Feasible region*. Select **Setup→Constraints**. Unset the "Points only" check box. Select the highest iteration. Make sure that the *Weight* surface and all the points are displayed (**Points→All**). Confirm that the two constraints $\sigma_L < 1; \quad \sigma_R < 1$ are active by studying the constraint boundaries and the position of the "Predicted Value" cursor. Also study the final design results in **View→Summary Report**.
   d. *Metamodel Accuracy*. Study the metamodel accuracy through the iterations by using the slider to change the iteration numbers from 1 to 5.

2. Confirm the optimal design as:

	Computed	Predicted
AreaL	1.45	▨
AreaR	0.72	▨
Base	0.45	▨
StressL	0.96	1.0
StressR	1.08	1.0
Weight	1.19	1.19

3. *2D Interpolator*: Find the predicted StressL and StressR value for AreaL = 1.0 and Base = 0.4 using the final metamodel. (1.55, 1.33).

Trade-off using Neural Net approximations

Starting file: `com.neural.correct`

Directory: `DESIGN_OPTIMIZATION/USER_DEFINED/NEURAL` *(Note this is the same folder as previously).*

Setup:

Use the sequential neural net optimization as in the previous section.

1. Set the **Create Pareto Optimal Frontier** option in the **Strategy** panel.
2. Because the above selection changes the strategy and metamodel type (hence the warning!), reset them to Sequential and Feedforward Neural Net.
3. Add StressL as a second objective.
4. Save the data under a new name.
5. Select the **Repair→Optimize** task. Set the iteration number to 5 (should be the last full iteration) and **Run**.

Exercises:

1. Select the **Tradeoff** option in **Viewer**. Select the StressL objective for the *X*-axis and the Weight objective for the *Y*-axis and observe the Pareto optimal curve representing the trade-off between stress limit (L) and structural weight.
2. When using minimum weight design, confirm the trends of the variables as a result of *tightening* the L stress constraint at the current optimum.

   Also use the *Self-Organizing Maps* chart to confirm your choices.

Entity	**Trend** (+/-)
`AreaL`	+
`AreaR`	+
`Base`	?
`Weight`	+

3. Using the Pareto Front, predict the minimum weight design for a stress constraint of 2.0 by resetting the constraint bounds to [-2;2] for both bars and rerunning the Pareto curve generation for iteration **5**. *Hint*: Use any of Tradeoff, Parallel Coordinate or HRV plot types to find a suitable design.

Entity	Value
AreaL	0.76
AreaR	0.38
Base	0.45
Weight	0.66
StressL	2.0
StressR	2.0

MODAL ANALYSIS AND TRACKING

Directory: **`DESIGN_OPTIMIZATION/MODAL_ANALYSIS`**

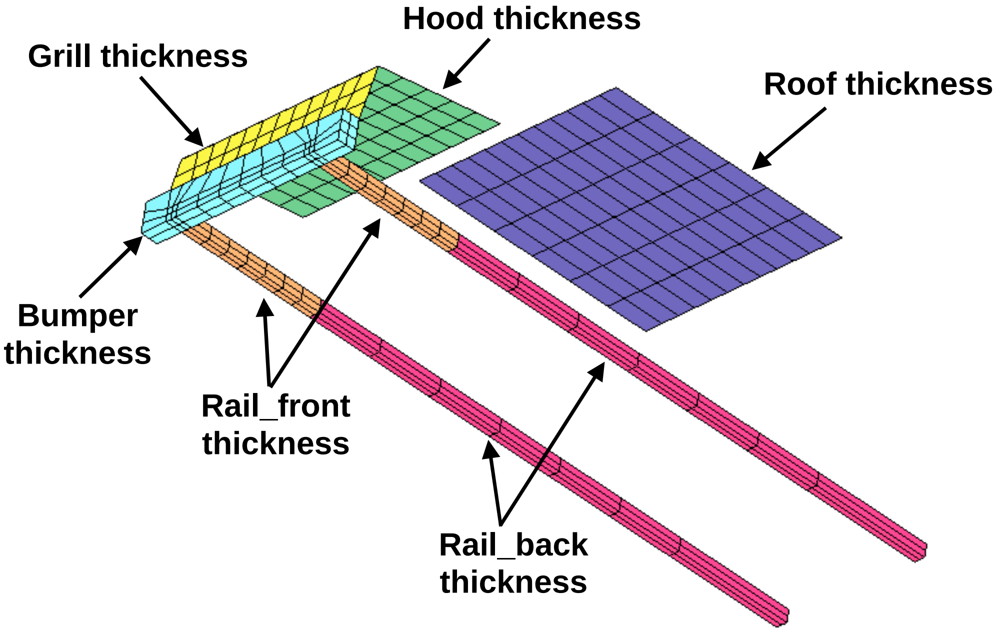
Special topics: Variable screening, mode tracking

Problem description

This example illustrates the following new features:

- Running an LS-DYNA implicit (double precision) solver.
- Variable screening
- Mode tracking done for an optimization problem with frequency criteria

The figure shows a modification of the geometry of the crashworthiness optimization problem. Rails have been added, and the combined bumper-hood section is separated into a grill, hood and bumper. The mass of the affected components in the initial design is 1.149 units while the torsional mode frequency is 1.775Hz. This corresponds to mode number 10.



The optimization problem is defined as follows:

Minimize Mass ($\boldsymbol{x}$) of the parts *2*, *3*, *4*, *5*, *6* & *7*.

subject to 1.7 Hz < Torsional mode frequency($\boldsymbol{x}$) < 1.9 Hz

Bounds on design variables: $\boldsymbol{x} \in [\mathbf{1},\mathbf{6}]$

Other data:

- Input file: `car6_NVH.k`
- Solver: `ls971_double`

Single iteration with all variables and variable screening

Directory: **`DESIGN_OPTIMIZATION/MODAL_ANALYSIS/SCREENING`**

Setup:

Identifying the Torsional mode:

1. Set up the design optimization problem as defined above. Use the double precision version of LS-DYNA (`ls971_double`).
2. Add the mass response.
3. Select 1 as the "Baseline Mode Number" for the frequency extraction on the **Responses** page. We have yet to identify the Torsional mode number, so "1" is as good a number as any.
4. Using the Sequential strategy, run the baseline case. Save the file under any name.
5. Manually inspect the mode shapes. The mode can be visualized by clicking on the single point in the scatter plot (Viewer) and selecting LS-PrePost to visualize the FE model and animation. Identify a pronounced twisting mode with the lowest mode number.

DOE task:

1. Select the DOE Study task.
2. Enter the previously identified mode number as the “Baseline Mode Number” for the frequency constraint. (Remember to "Replace").
3. Add the **New Mode Number** and **Generalized Mass** as two additional responses in addition to Mass and Frequency. Selecting these responses aids visualization of the mode tracking.
4. Use a linear approximation with default settings. Save the work under any name.
5. Go to the **Run** panel.
6. Check the "Clean Start" box and Run. *A clean start is important since responses have been added which changes the design problem.*

Exercise:

1. Which mode numbers are found when the twisting mode is tracked for the 10 different designs. (7, 10 and 11)
2. What is the lowest value of the generalized mass found for the 10 different designs and for which mode (0.79; 11).

3. What is the highest value of the generalized mass found for the 10 different designs and for which mode (1; 10).
4. Confirm that the three most important variables are **`tbumper`**, **`troof`** and **`trailb`**. Can this be stated with absolute certainty? Use both the Linear ANOVA as well as GSA plot to make your decision.

Optimization with all variables

Directory: **`DESIGN_OPTIMIZATION/MODAL_ANALYSIS/ITERATE`**

Starting file: **`com.frequency.iterate.start`**

Setup:

1. Set the strategy to SRSM and check the sampling panel for the default settings of the metamodel (lin. polynomial) and point selection (D-optimal).
2. Set the limit on the number of iterations to 5 (or more if time is available)
3. Run the optimization.

Exercise:

1. Study the optimization history of the objective and constraints.
2. Study the optimization history of the mode number. The variation of the mode number due to tracking can also be viewed in the Scatter plot.
3. Record the optimal values of the mass and frequency.
4. What is the new mode number?
5. What is the value of the generalized mass?

	Value
tbumper	3.8
troof	1
trailb	1.2
trailf	1
tgrill	1
Mass	0.74
Frequency	1.7
Mode	11
Generalized Mass	0.98

Optimization with screened variables

Directory:
`DESIGN_OPTIMIZATION/MODAL_ANALYSIS/ITERATE_SCREENED`

Starting file: **`com.frequency.iterscreened.start`**

Setup:

1. Use the SRSM strategy.
2. Remove the unimportant variables determined previously by setting them to constants (`trailf` and `tgrill`).
3. Set the limit on the number of iterations to 5 (or more if time is available)
4. Run the optimization.

Exercise:

1. Compare the optimum result of the screened and unscreened example.

	All variables	Screened
tbumper	3.8	2.5
troof	1	1
trailb	1.2	1.6
trailf	1	-
tgrill	1	-
Mass	0.74	0.88
Frequency	1.7	1.7
Mode	11	11

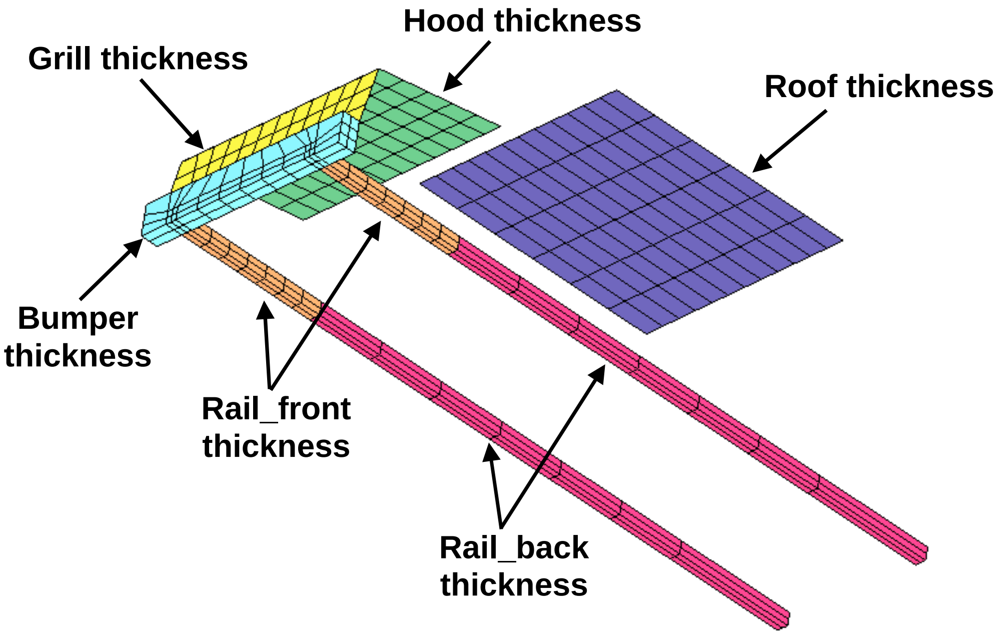
MULTIDISCIPLINARY OPTIMIZATION

Directory: **DESIGN_OPTIMIZATION/MDO**

Problem description

This example has the following new features:

- LS-DYNA is used for both explicit crash and implicit NVH simulations.
- Multidisciplinary design optimization (MDO) is illustrated with a simple example.
- Scaling of the constraints is performed to ensure that the optimizer treats their violations equally.



The figure shows a modification of the geometry of the crash example attempted previously. Rails are added, and the combined bumper-hood section is separated into a grill, hood and bumper. The mass of the affected components in the initial design is 1.328 units while the torsional mode frequency is 1.775Hz. This corresponds to mode number 10. The Head Injury Criterion (HIC) based on a 15ms interval is initially 17350. The initial intrusion of the bumper is 536mm.

The optimization problem is defined as follows:

Minimize Mass ($\boldsymbol{x}_{\text{crash}}$)

subject to HIC($\boldsymbol{x}_{\text{crash}}$) < 900
Intrusion($\boldsymbol{x}_{\text{crash}}$) < 500mm
Torsional mode frequency($\boldsymbol{x}_{\text{NVH}}$) = 1.8Hz

Nodes 184 and 432 are used for the intrusion calculation and node 432 for HIC. Both a crash analysis and modal analysis need to be conducted. Scale all 3 constraints by the respective bound.

The purpose of the example is as follows:

1. Select the important variables for each discipline using *Sensitivities*.
2. Optimize the design with the limited number of partially shared variables.
3. Use the frequency constraints and mode tracking.
4. Solve a problem with an infeasible solution.

Variable screening: single iteration with all variables

Directory: **`DESIGN_OPTIMIZATION/MDO/SCREENING`**

Starting file: **`com.mdo.screening.start`**

Setup:

1. Use the GUI to open the **`com.mdo.screening.start`** file.
2. Check the availability of the variables from the input files by clicking on each variable in the variables panel.
3. Use all variables, except **`tgrill`**. (Change **`tgrill`** to a constant).
4. Set the task to DOE Study and Run.

Exercise:

1. Using the *Global Sensitivities* selection, estimate the 3 most important variables of the CRASH discipline. Use all the available variables for NVH.

	CRASH	**NVH**
tbumper		✓
troof		✓
trailb		✓
trailf		✓
thood		n/a

Optimization using screened variables

Directory: **`DESIGN_OPTIMIZATION/MDO/ITERATE`**

Starting file: **`com.mdo.iterate.start`**

Setup:

1. Set the task to Metamodel-based Optimization.
2. Select the SRSM strategy.
3. Variables panel: Select the 3 most important variables of the CRASH discipline (observed in the previous section).

   Note: If thood is not selected for CRASH, change it to a constant, since it is also absent in the NVH model (the NVH model has no hood to begin with!).

4. Responses panel:
   a. Normalize the HIC constraint by adding a composite expression HIC/900.
   b. Normalize the Intrusion constraint: Intrusion_normalized Intrusion/500.0
   c. Then set the constraint limits for these two normalized constraints to 1.0 (so as to have the same effect as the original constraint). Use the normalized constraints instead of the original (not normalized) ones.

Normalization is done to avoid conditioning problems when choosing both large and small constraints. A constraint with a large number will inflate its importance in the problem; hence constraints with relatively smaller numbers tend to be ignored.

5. Set the frequency constraint as an equality constraint of 1.8 by setting both the lower and upper bounds to 1.8.

6. Set the number of iterations to 5 (or more, time allowing) and do the optimization.

Exercise:

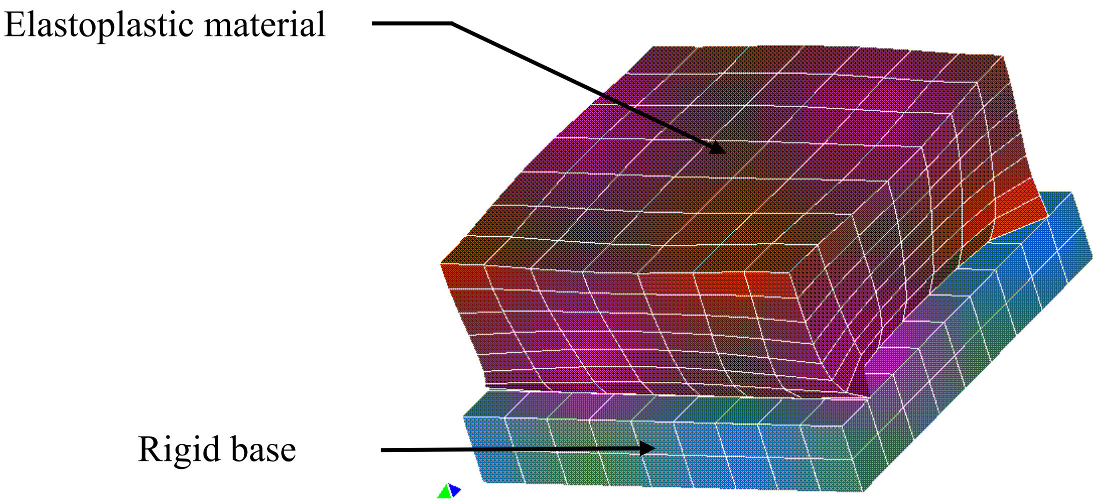
1. Study the feasibility of the solution. Which response has the largest normalized constraint violation? Use **View→Summary Report**.

2. Use the 2D Interpolator to study the sensitivity of all functions to a variable which is only present in the CRASH case. What do you see?

SYSTEM PARAMETER IDENTIFICATION

Directory: **`PARAMETER_IDENTIFICATION/MATERIAL`**

Special topics: Point-based and history-based parameter identification

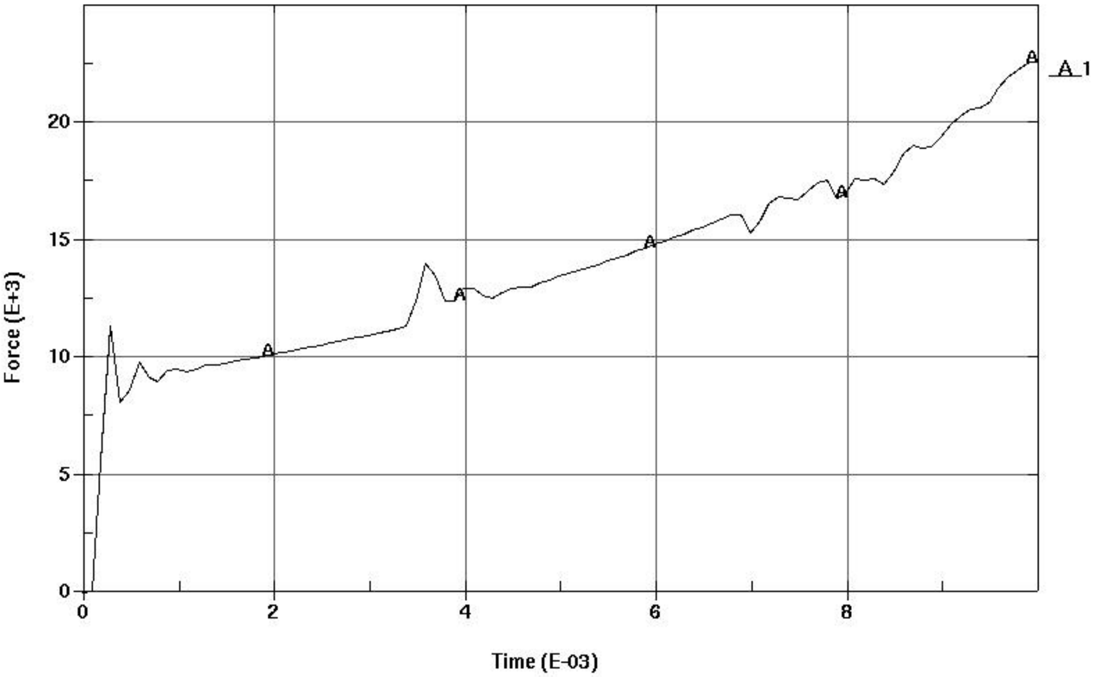


Problem description

The material parameters of a foam material must be determined from experimental results, namely the resultant reaction force exerted by a cubic sample on a rigid base. The problem is addressed by minimizing the residual resultant reaction force (**`rcforc`** binary database) with the material parameters Young's modulus E (**`YMod`**) and Yield stress Y (**`Yield`**) as unknown variables. The Mean Squared Error is computed using the formula below:

$$\varepsilon = \frac{1}{P}\sum_{p=1}^{P} W_p\left(\frac{f_p(\mathbf{x})-G_p}{s_p}\right)^2 = \frac{1}{P}\sum_{p=1}^{P} W_p\left(\frac{e_p(\mathbf{x})}{s_p}\right)^2$$

The "experimental" resultant forces are shown below. The results were generated from an LS-DYNA run with the parameters (E=10^6, Y=10^3). Samples are taken at times 2, 4, 6 and 8 ms:



Time	Force
2	10000.
4	13000.
6	15000.
8	17000.

New Points illustrated by this example:

- How to define a history.
- How to do parameter identification.
  - using a point-based Mean Squared Error composite function.
  - using a history-based Mean Squared Error composite function.

- How to construct crossplots.
- Using multiple simulation models in the same optimization problem (multi-case).

Point-based Mean Squared Error

Directory:
PARAMETER_IDENTIFICATION/MSE_POINT/SINGLECASE

Setup:

1. Parameterize the material data in the LS-DYNA keyword file as:

   ```
   1,.001,&YMod,.3,&Yield,10.,0.
   ```

2. Choose the SRSM strategy with linear polynomial approximations.
3. Use the starting values of (*E*=700,000; *Y*=1,500).
4. Use [5e5; 2e6] and [5e2; 2e3] as bounds on the variables.
5. Extract the *Z*-slave force **history** **`Force`** from **`RCFORC.`** The interface ID = 1**.**
6. Use response-function expressions to compute the forces, e.g. **`Force(0.002)`**.
7. Set up the MSE composite called **`MSE`** from the table above. Select **`MSE`** as the objective function to minimize. (The composite to be selected is the "standard" type in which Target values can be defined, not a Composite-Expression).
8. Set the number of iterations to 5 and **Run**.

Exercise:

1. **Accuracy:**
   1.1 Study the accuracy of the response forces at the various times. Use the **Accuracy** option as well as the **Optimization History** option in which the error history can be viewed.
2. **Convergence:**
   2.1 Confirm the optimal parameters and MSE value:

	Value
YMod	1.1e6
Yield	993
MSE	2.4e5

   2.2 Study the MSE history.

2.3 Comment on the nature of the optimization history of the Yield stress vs. that of the Young's Modulus. Try to find a reason for the differences in the convergence behavior of the two variables by studying the Global Sensitivity bar chart for MSE.

3. **Confidence Intervals:**

3.1 Study the table of 90% Confidence Intervals of **`YMod`** and **`Yield`** using the View → Summary Report option (scroll to the bottom). The confidence intervals can also be viewed using the "Variables" selection under "Optimization" (main plot selector).

	Value	Lower bound	Upper bound
YMod	1.1e6	-4.5e6	6.8e6
Yield	993	874	1114

3.2 Comment on the confidence levels of the individual parameters.

History-based MSE

Directory:
`PARAMETER_IDENTIFICATION/MSE_HISTORY/SINGLECASE`

Problem description:

The following files are provided:

▪ `Test1.txt`	Measured data of Exp. 1
▪ `foam1.k`	Model representing Exp. 1

The experimental results are:

Displacement	Force Resultant
0.36168	10162
0.72562	12964
1.0903	14840
1.4538	17672

Note that the abscissa is not time but displacement.

Setup:

1. Select the SRSM strategy.
2. Specify **`foam1.k`** as the input file.
3. Specify the parameter bounds as [5e5,2e6] for the Young's modulus and [5e2,2e3] for the yield stress.
4. Go to the "Histories" panel:
   a. Create **`Disp1`** as the *z*-displacement at node 296 (nodout).
   b. Create **`Force1`** as the *z*-slave reaction force at interface 1 (rcforc).
   c. Create a Crossplot **`F_vs_d`** using **`-Disp1`** and **`Force1`**. Note Disp1 is *negative*.
   d. Import **`Test1.txt`** using the history "File" option.
5. Go to the "Responses" panel and add a "MeanSqErr" residual called **`MSE`**. The "**MeanSqErr**" option is selected from the list on the left.
6. Select **`MSE`** as the objective.

7. Run the example with an iteration limit of 3.

Exercises:

1. Select **Optimization→History** and the **MSE** composite
2. View the optimal curve matching by selecting Histories under Simulations and then selecting **F_vs_d** and **Test.**
3. Also view all the histories by selecting **All** in the iteration control window. Use *Iteration* color coding.
4. View the "Mean Squared Error Residuals" table at the end of View→Summary Report. Also view **Optimization→Variables** for the 90% Confidence interval.
5. Confirm the confidence intervals for the two parameters from the lower-most table. Comment on the significance of the confidence intervals.
6. In **Metamodel→Histories** study the sensitivity of F_vs_d to the Young modulus and Yield stress variables.

Parameter name	Value	90% Confidence Interval	
		Lower	Upper
YMod	7.2e5	-7.5e6	8.9e6
Yield	1013	892	1134

Point-based MSE: multiple cases *(optional)*

Directory:
PARAMETER_IDENTIFICATION/MSE_POINT/MULTICASE

Starting file: **com.msepoint.multi.start**

Problem description:

Experimental results are added to the previous ones using a second test. These test results are needed to improve the conditioning of the regression problem for identifying the Young's modulus. The following files are provided:

▪ **foam1.k**	Model representing Exp. 1
▪ **foam2.k**	Model representing Exp. 2
▪ **com.msepoint.multi.start**	Command file

The experimental results are:

foam1.k		**foam2.k**	
Time (ms)	Force Resultant 1000’s	Time (ms)	Force Resultant 1000’s
2	10	0.091	0
4	13	0.195	6.32
6	15	0.298	16.3
8	17	-	-

Setup:

Incorporate the two experiments in a parameter identification run as follows:

1. Start lsoptui with the file **com.msepoint.multi.start**. Add Case2 with input file **foam2.k**.
2. Make sure both variables are valid for both cases (Variables panel).
3. In the sampling panel, set Case2 as a *duplicate* of Case1. This ensures that points of the Composites (which is common to both cases) will be plotable.

4. Extract seven computational results corresponding to the ‘test’ results above using the necessary function expressions, e.g. **`Force1(0.002)`**, **`Force2(0.00009)`** based on the extracted histories **`Force1`** and **`Force2`** respectively. Note that **`Force2`** belongs to Case2.
5. Use starting values of (700,000;1,500) as the values for (*E*,*Y*).
6. Define a MSE composite formulation using the 7 tabulated experimental values. This will ensure the minimum mean squared error.
7. Run the optimization using a sequential linear approximation strategy (iteration limit 4).

Exercises:

1. **Convergence:**

   1.1 Confirm the optimal parameters and MSE value:

	Value
YMod	`1.04e6`
Yield	`996`
MSE	`1.7e5`

   1.2 Study the history of the force residual.

   1.3 Comment on the nature of the optimization history of the Yield stress vs. that of the Young’s Modulus. Try to find a reason for the differences in the convergence behavior of the two variables by studying the Sensitivity bar chart and Accuracy plots.

2. **Confidence Intervals:**

   2.1 Study the table of 90% Confidence Intervals of **`YMod`** and **`Yield`** using the View → Summary Report option (scroll to the bottom):

	Value	Lower bound	Upper bound
YMod	`1.04e6`	`.83e6`	`1.25e6`
Yield	`996`	`937`	`1055`

   2.2 Comment on the confidence levels of the individual parameters and compare them to the single case calibration.

History-based MSE: multiple cases

Directory:
PARAMETER_IDENTIFICATION/MSE_HISTORY/MULTICASE

Starting file: **com.msehistory.multi.start**

Setup:

The previous example shows that the Young's modulus cannot be confidently identified by test points in the plastic range. Therefore a second test file (Test2) has been added which consists of test points in the linear range of force vs. deformation. Modify the starting file as follows:

1. Add Case2 with input file foam2.k
2. Change the Case2 sampling scheme to **Duplicate**.
3. Histories panel:
    a. Select *Case2* in the history tree-list.
    b. Add Disp2, the *z*-displacement at node 288
    c. Add Force2, the *z*-reaction slave force at interface 1
    d. Add the crossplot of these values as F2_vs_d2 (-Disp2)
    e. Add Test2 as an imported history from Test2.txt
4. Response panel:
    a. Add MSE2 using Test2 and F2_vs_d2
5. Add MSE2 as an objective
6. Run the examples with an iteration limit of 4.

Exercises:

1. Study the Optimization History.
2. View F1_vs_d1 vs. Test1 using the **Simulations→Histories** selection.
3. View F2_vs_d2 vs. Test2
4. Note down the optimal values and confidence intervals for the two parameters (**View→Summary Report**, scroll to the bottom):

Parameter name	Value	90% Confidence Interval	
		Lower	Upper
YMod	1.0e6	8.0e5	1.20e6
Yield	1010	946	1075

5. Compare the confidence intervals to those of the single case.

RELIABILITY ANALYSIS

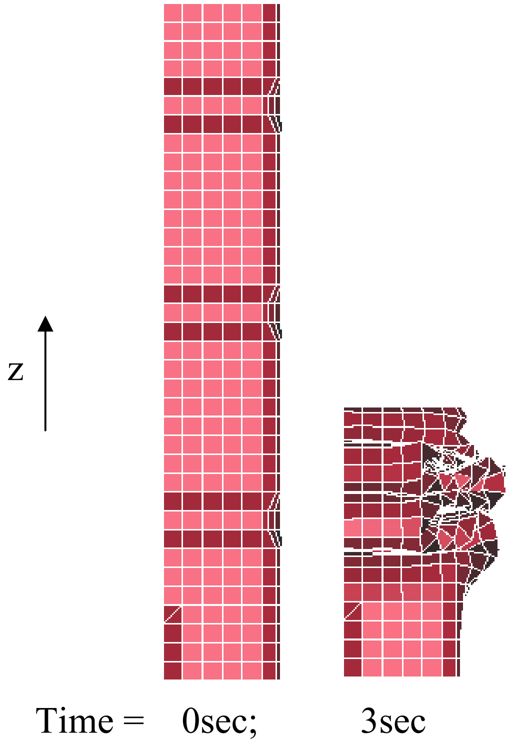
Reliability analysis is used to compute the probabilities of events.

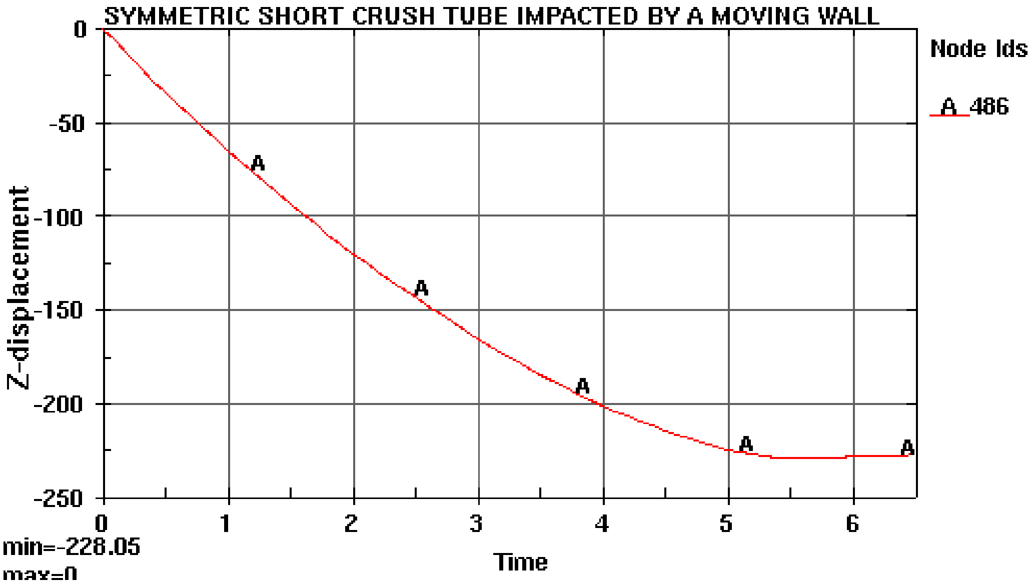
This example demonstrates:

- Monte Carlo Analysis, and
- Metamodel-based Monte Carlo Analysis.

Problem description

This example is a Monte Carlo analysis of a steel tube being crushed. The effect of both a variation in material thickness and a variation in the plastic stress-strain curve is investigated. The geometry is shown in the figure in its original and partially deformed state. The z-displacement of the upper tube boundary is also shown below as a history. The minimum value of the z-displacement is used as the response variable, and the response is compared to a crush distance of the selected nominal design.





A scale factor (SIGY) is used to modify the plastic stress-strain curve. The binary LSDA LS-DYNA database is used for the extraction of the maximum displacement (largest negative value given the direction of the z-axis, hence the MIN selection) response.

Monte Carlo Analysis

Setting up the problem

1. Open the file `com.monte_carlo.start` in the directory `PROBABILISTIC/RELIABILITY/MONTE_CARLO`.
2. Change the task to `Direct Monte Carlo Analysis.`
3. Change the `T1` constant to be a noise variable with a variation around the nominal value described by a *normal* distribution with a mean of 1 and a standard deviation of 0.05. Note that you have to change the task, change the variable type, and create the distribution before you can assign the distribution.
4. Also change `SIGY` to have a variation around the nominal value described by a *normal* distribution with a mean of 1 and a standard deviation of 0.05.
5. A Latin Hypercube Sample (stratified Monte Carlo) experimental design should be used to reduce the cost of the stochastic simulation. Use 10 experiments to allow the simulation to finish in reasonable time.

6. Constrain the displacement constraint to be more than -230 (a lower bound therefore).
7. Save your work under any name.
8. Run the Monte Carlo Analysis.

Viewer

1. View the variables and response using a `Correlation Matrix` plot.
2. Create a `Scatter Plot` of `TOP_DISP` on the y-axis and the T1 variables respectively on the x-axis. Hint: you can get this plot by double clicking on the corresponding small plot in the correlation matrix plot.
3. View the statistics of `TOP_DISP` response using the `Statistics` functionality or by double clicking in the `Correlation Matrix` plot on the corresponding histogram plot. Use the `Summary` and `Bounds` options to:
    a. Verify that the mean is -227 and standard deviation is 7.8
    b. Verify that the probability of the response being larger than -220 is 0.2.
    c. Verify that the probability of the response being less than -230 is 0.4.
    d. Verify that the probability of the response being less than -235 is 0.3.
4. Identify, if possible, the variable contributing the most to the variation of the crush distance using a `Correlation` plot. Is it possible to select the most influential variable given the confidence intervals?

DYNA Stats

Use `lsoptui Dynastats` panel to display statistics in LS-PREPOST.

1. Create as fringe plots:
    a. The mean of the `z_displacement`.
    b. Standard deviation of the `z_displacement`.
    c. Correlation of the `z_displacement` and the `TOP_DISP` response.
2. Generate and display the three fringe plots.
3. Create as history plots:
    a. Mean and standard deviation of the `TOP_DISP_HIST` history.
    b. Individual histories from each LS-DYNA run of the `TOP_DISP_HIST` history.

4. Generate and display the two history plots.
5. Create and display a fringe plot showing the probability of the *z_displacement* being less than -230.

Metamodel-based Monte Carlo Analysis

Setting up the problem

1. Open the file `com.metamodel.start` in the directory `PROBABILISTIC/RELIABILITY/METAMODEL`.
2. Change the task to `Metamodel-based Monte Carlo Analysis.`
3. Change the `T1` constant to be a noise variable with a variation around the nominal value described by a *normal* distribution with a mean of 1 and a standard deviation of 0.05. Note that you have to change the task, change the variable type, and create the distribution before you can assign the distribution.
4. Also change `SIGY` to have a variation around the nominal value described by a *normal* distribution with a mean of 1 and a standard deviation of 0.05.
5. Verify that the metamodel is a quadratic surface and that the experimental design is suitable.
6. Constrain the displacement constraint to be more than -230 (a lower bound therefore).
7. Save your work under any name.
8. Run the Metamodel-based Monte Carlo Analysis.

Viewer

1. View the response values using the `Metamodel Surface` plot. Note that you have to select `Points, Iterations = All` to view the actual response values.
2. Considering the `TOP_DISP` response in the `Statistical Tools`:
    a. Verify that the metamodel is selected to compute the statistics.
    b. Verify that the mean is -228 and standard deviation is 7.3 using the `Summary Plot type`.

    c. Verify that the probability of the response being larger than -220 is 0.15 using the `Bounds plot type`.
    d. Verify that the probability of the response being less than -230 is 0.39.
    e. Verify that the probability of the response being less than -235 is 0.15.

3. Identify the variable contributing the most to the variation of the crush distance using the `Stochastic Contribution` plot.

DYNA Stats

Use `lsoptui Dynastats` panel to display statistics in LS-PREPOST. Always use a quadratic response surface to compute the results.

1. Create as fringe plots:
    a. The mean of the `z_displacement`.
    b. Standard deviation of the `z_displacement`.
2. Verify that a quadratic response surface is used to compute the results.
3. Generate and display the two fringe plots.
4. Create, generate, and display the following history plots:
    a. The statistics of the `TOP_DISP_HIST` history.
    b. Individual histories from each LS-DYNA run of the `TOP_DISP_HIST` history.
5. Using single variable mode (contribution analysis)
    a. View as a fringe plot:
        a. The standard deviation of the `z_displacement` due to the `SIGY` variable.
        b. A plot of the index of the variable contributing the most to the `z_displacement.`
    b. View as a history plot:
        a. The standard deviation of the `TOP_DISP_HIST` due to each of the variables.

RELIABILITY BASED DESIGN OPTIMIZATION (RBDO)

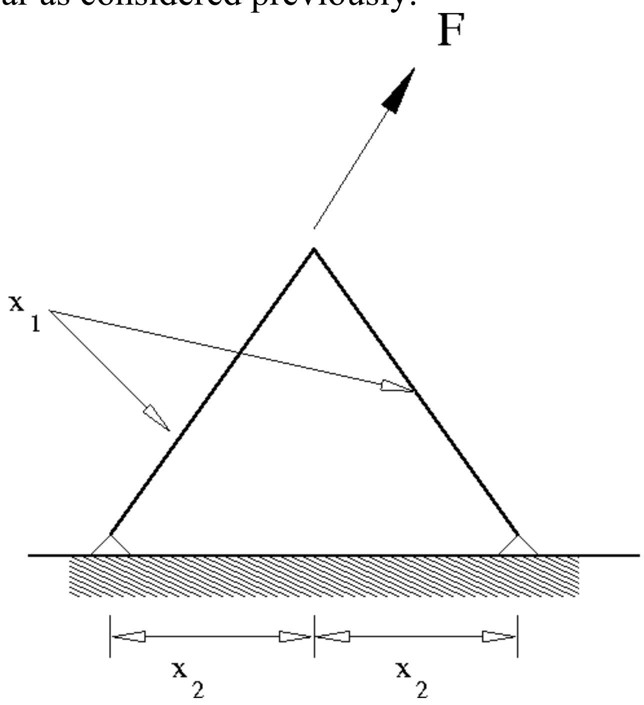
Reliability Based Design Optimization (RBDO) includes the variation of the design variables into the design problem.

This example demonstrates:

- Reliability Based Design Optimization (RBDO)
- Creating statistical distributions and assigning them to design variables.
- Probabilistic constraints.

Problem description

This is the two-bar as considered previously.



The two-bar truss problem. The problem has two variables: the thickness of the bars and the leg width as shown. Both the thickness and the leg width have uncertainties associated with them. The weight of the structure is minimized and the probability of the maximum stress in the bars exceeding the failure stress is constrained.

A value of the `base` variable and `area` variable must be obtained that minimizes mass while respecting a probability of not exceeding the allowable stress value.

Setting up the problem

1. Open the file `com.2bar.rbdo.start` in the directory `PROBABILISTIC/RBDO`.
2. Change the `area` variable to have a variation around the nominal value described by a *normal* distribution with a standard deviation of 0.1. Note that you have to change the task to `RBDO/Robust Parameter Design` to assign the distribution.
3. Change the `base` variable to have a variation around the nominal value described by a *uniform* distribution with a range of 0.2.
4. Verify that the metamodel is a quadratic surface and that the experimental design is suitable.
5. Set the objective to be the weight of the structure.
6. Change the constraint such that the probability of exceeding the upper bound on the stress constraint does not exceed 0.05.
7. Save your work under any name.
8. Run the RBDO job.

Viewer

1. Verify that the optimum value of the `area` variable is computed as 1.6. This can be done using the `Opt History` functionality.
2. Verify that the optimum value of the `base` variable is computed as 0.38.
3. Verify that the probability of exceeding the upper bound of the constraint has converged to 0.05 using the `Optimization History` functionality.
4. Using the `Stochastic Contribution` functionality, verify that:
    a. The standard deviation of the stress due to all the variables is 0.06.
    b. And that almost all of the variation of the stress is caused by the `area` variable.

ROBUST PARAMETER DESIGN

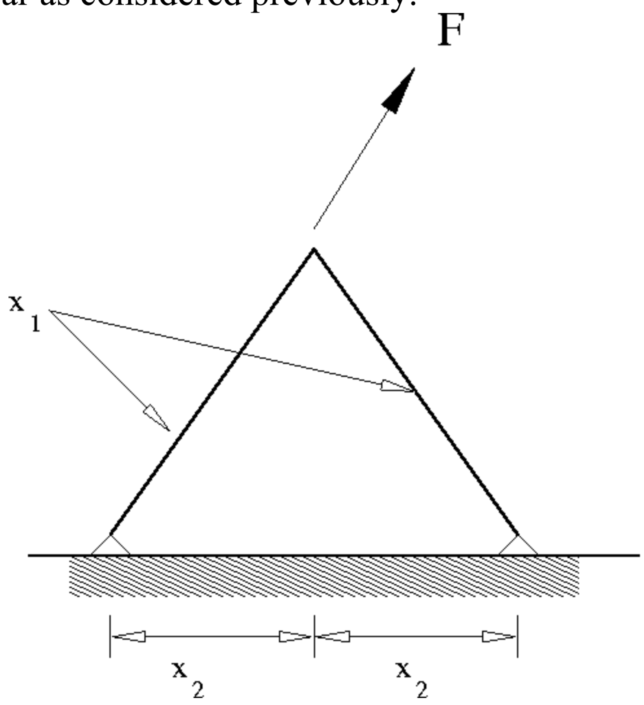
Robust parameter design selects designs insensitive to the variation of given parameters.

This example demonstrates:

- Robust Parameter Design

Problem description

This is the two-bar as considered previously.



The two-bar truss problem. The problem has two variables: the thickness of the bars and the leg widths as shown. The bar thicknesses are noise variables while the leg widths are adjusted (control variables) to minimize the effect of the variation of the bar thicknesses. The maximum stress in the structure is monitored.

A value of the `base` variable must be obtained that makes the stress response insensitive to variation of the `area` variable.

Setting up the problem

1. Open the file `com.2bar.robust.start` in the directory `PROBABILISTIC/ROBUST_PARAMETER_DESIGN.`

2. Change the `area` variable to be a noise variable described by a normal distribution with a mean of 2.0 and standard deviation of 0.1. Note that you have to change the task to RBDO to assign the distribution.
3. The `base` variable should be left unchanged. Verify that it is a control variable with a starting value of 0.8, allowable minimum of 0.1, and allowable maximum of 1.6.
4. Change the metamodel and experimental design to consider interaction between the variables.
5. Set the objective to be the standard deviation of the existing stress response. Note that you have to create a response that is the standard deviation of the existing stress response in order to do this.
6. Save your work under any name.
7. Run the robust parameter design job.

Viewer

1. Verify that the value of the `area` noise variable remains unchanged. This can be done using the `Optimization History` functionality.
2. Verify that the optimum value of the `base` control variable is computed as 0.5.
3. Verify that the standard deviation composite has converged.
4. Using the `Metamodel` facility to investigate how the value of the stress standard deviation changes with the variables.

BIFURCATION/OUTLIER ANALYSIS

Bifurcation analysis investigates scatter in the results, with the goal of understanding bifurcations in the results.

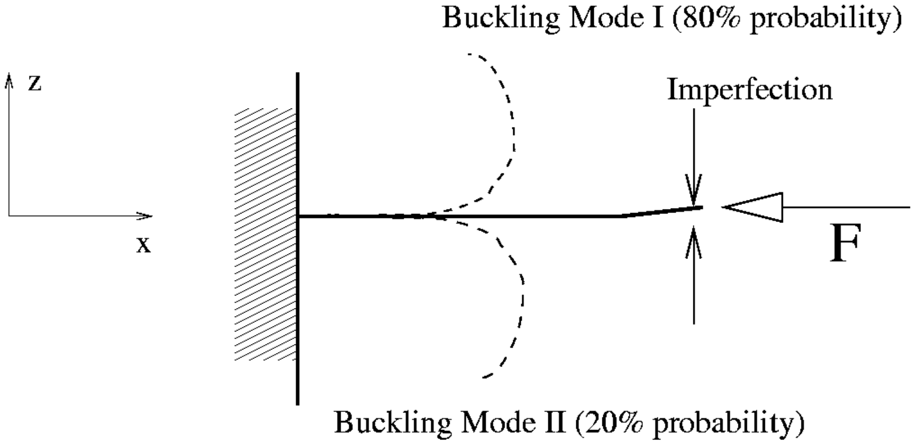
This example demonstrates:

- Monte Carlo analysis
- Identification of different buckling modes in the structure

This example is also presented in the LS-OPT User's Manual.

Problem description

The plate as shown has two buckling modes. Buckling in the positive z-direction occurs with a probability of 80% while buckling in the negative z-direction occurs with a probability of 20%. Assigning a distribution (in this case uniform) to an imperfection at tip nodes allows control of the probability of buckling.


Buckling Mode I (80% probability)
z
x
Imperfection
F
Buckling Mode II (20% probability)

A Latin hypercube experimental design is used for the Monte Carlo analysis. We analyze only five points. Given the probability of 20% of buckling in the negative z-direction and a Latin hypercube experimental design, one run will buckle in the negative z-direction. The next section will demonstrate how to find out which run contains the different buckling mode.

Setting up the problem

1. Open the file `com.outlier` in the directory `PROBABILISTIC/OUTLIER`.

2. Run the Monte Carlo job.

Viewer

1. Use Viewer to determine the minimum and maximum displacements of the tip by plotting the `tip_z` vs. the tip `tip_x` response using a `Scatter` plot.
2. Identify the LS-DYNA jobs associated with the maximum and minimum values of the `tip_z` response.

DYNA Stats

Use the `lsoptui Dynastats` panel to display the bifurcations in LS-PREPOST.

1. Create, generate, and display a fringe plot of the standard deviation of the *z-displacement*.
2. Create a bifurcation plot of the plot created in the previous step.
   a. Select to overlay the FE model of the job with the maximum value. Also overlay the FE model of the job with the minimum value.
   b. Select the maximum and minimum overall values by considering the whole model.
   c. Display this plot. You should see three FE models with the bifurcation clearly visible.
3. Create another bifurcation plot considering the `z_displacement` values at a node:
   a. Determine the node in the structure where the maximum variation of the z-tip displacements occurs by plotting the range of the `z_displacement.`
   b. Overlay the the FE model of the job with the maximum value at this node. Also overlay the FE model of the job with the minimum value at this node.
4. Using the history statistics of the z-tip displacement (`NHist_Z`):
   a. At what analysis time did bifurcation start?
   b. During which analysis time interval can the bifurcation be viewed?
   c. Identify the LS-DYNA jobs associated with the maximum and minimum values.

ROBUSTNESS OF METAL FORMING (OPTIONAL)

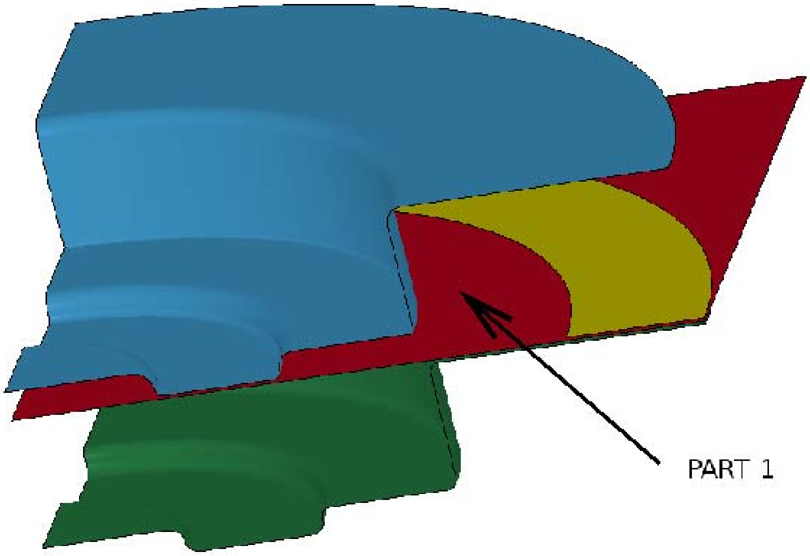
Metal forming requires the analysis of adaptive results at specific coordinates. The results can then be compared even though the node locations and numbers differ between FE models.

This example demonstrates:

- Robustness of metal forming,
- Mapping results from adaptive meshes, and
- Using a stochastic field described with a sinusoidal perturbation.

Problem description

The structure shown is a simple metal forming problem. Part 1, modeled as adaptive, is the work piece being deformed.



A Monte Carlo analysis is done to estimate the scatter that can occur in practice. The variables are:

- YIELD, the yield strength of the material used in the work piece;
- FS1, the scaling of the contact force between the work piece and the punch;
- FS2, the scaling of the contact force between the work piece and the die;
- FS3, the scaling of the contact force between the work piece and the blank holder; and
- POFF, the offset of the stochastic field.

Setting up the problem

The results from the Monte Carlo analysis should already exist in your directory `PROBABILISTIC/METALFORMING`. If not, run the file com.metal_MC.

Viewer

1. Identify the most important variable for the maximum thickness reduction by viewing the correlation of the maximum percent thickness reduction (*prc_thick_red_max*) with the variables.
2. Verify the variable identified in 1 using `Scatter` plots or a `Correlation Matrix` plot. Plot the response (*prc_thick_red_max*) against the various variables.

DYNA Stats

1. Set the metal forming options to map the results of part 1 (follow the coordinates instead of the nodes).
2. Plot the variation of the sheet thickness. The standard deviation should have a maximum value of order 0.03 — 0.05.
3. Set the FLD options to use the curve 90 in the LS-DYNA input file. Plot "*maxima flc-eps1*".