



LS-OPT[®] TRAINING CLASS

OPTIMIZATION AND ROBUST DESIGN

TUTORIAL PROBLEMS

LS-OPT Version 6.0

Nielen Stander (Ph.D.)
Willem Roux (Ph.D.)
Anirban Basudhar (Ph.D.)

Livermore Software Technology Corporation

May, 2019

Copyright © 1999-2018 by Livermore Software Technology Corporation
All Rights Reserved

TABLE OF CONTENTS

INTRODUCTION	5
1.1 Software tools	6
1.2 General procedure for problem solution.....	6
1.3 Using LS-OPTui for input	6
1.4 Standard files contained in all directories	6
1.5 Output files	6
1.6 Execution of LS-OPT	7
1.7 Using LS-OPTui for post-processing	7
1.8 Result interpretation	8
1.9 Reference documents.....	10
SIMPLE OPTIMIZATION AND VIEWING RESULTS	11
Problem description.....	11
.....	12
Post-processing study using the Viewer.....	13
SETTING UP AN OPTIMIZATION RUN FROM SCRATCH.....	17
Basic setup for a single iteration	17
Add a constraint and repair the optimization run	22
FINDING A CONVERGED SOLUTION.....	23
Setting up the problem.....	23
Exercise.....	23
DISCRETE OPTIMIZATION	26
Single iteration.....	26
USER-DEFINED OPTIMIZATION PROBLEM	28
Problem description:.....	28
Sequential optimization with domain reduction.....	30
Reducing the number of variables by constraining the bar areas.....	32
IMPORTING ANALYSIS RESULTS.....	33
Setting up the problem.....	33
Exercise.....	34
DIRECT OPTIMIZATION.....	35
Setting up the problem.....	35
Exercise.....	35
MATERIAL PARAMETER IDENTIFICATION	37
Problem description	37
Ordinate-based MSE.....	39
Ordinate-based MSE: multiple cases.....	41
Point-based Mean Squared Error (optional).....	43

Parameter identification with hysteretic response: multiple cases	45
GISSMO FAILURE MODEL CALIBRATION (SHEAR LOAD CASE)	48
.....	
Problem description	48
FULL-FIELD CALIBRATION	51
Problem description	51
Dynamic Time Warping	52
MODAL ANALYSIS AND TRACKING	57
Problem description	57
Identifying the mode	59
Optimization	61
MULTIDISCIPLINARY OPTIMIZATION	62
Problem description	62
Variable screening: First iteration with all variables	64
Iterative optimization using screened variables	65
MULTI-OBJECTIVE OPTIMIZATION	66
Pole crash problem using Direct GA	66
Setting up the problem	66
Exercise	67
Pole crash problem using Metamodel-Based optimization	70
Setting up the problem	70
Exercise	70
MULTI-LEVEL OPTIMIZATION	72
Two-level pole crash problem	72
Problem description	72
Setting up the problem	72
Exercise	74
RELIABILITY ANALYSIS	75
Problem description	75
Monte Carlo Analysis	76
Setting up the problem	76
Viewer	77
DYNA Stats	77
Metamodel-based Monte Carlo Analysis	78
Setting up the problem	78
Viewer	79
DYNA Stats	79
CLASSIFIER-BASED RELIABILITY ANALYSIS WITH SHAPE	
PARAMETRIZATION	81
Problem description	81

Setting up the problem	82
Viewer.....	82
RELIABILITY BASED DESIGN OPTIMIZATION (RBDO).....	83
Problem description	83
Setting up the problem.....	84
Viewer.....	84
ROBUST PARAMETER DESIGN.....	85
Problem description	85
Setting up the problem.....	86
Viewer.....	86
TOLERANCE OPTIMIZATION WITH IMPORTED METAMODEL	
.....	87
Imported metamodel-based Monte Carlo analysis with a fixed tolerance	
.....	89
Setting up the problem.....	90
Exercise:.....	90
Multi-objective tolerance and mass optimization of vehicle	92
Setting up the problem.....	93
Exercise:.....	94
BIFURCATION/OUTLIER ANALYSIS	95
Problem description	95
Setting up the problem.....	95
Viewer.....	96
DYNA Stats	96
ROBUSTNESS OF METAL FORMING (OPTIONAL).....	98
Problem description	98
Setting up the problem.....	99
Viewer.....	99
DYNA Stats	99

Thursday, May 02, 2019

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

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

- Reliability
- Outlier Analysis
- Reliability-based design optimization
- Robust design

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.

The GUI allows easy navigation amongst examples by simply using the *New* or *Open* options in the File menu (menu bar, top left). *New* allows the definition of new projects while *Open* lists the existing projects. *Save* (*ctrl+S*) allows the user to save the current project with any changes that have been made in the GUI.

1.1 Software tools

The following tools are used in this tutorial:

- LS-OPT together with LS-OPTui and Viewer (Requires Ver. 6.0 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)
- Perl (for user-defined problems)

Please note that this problem set was defined using LS-OPT Version 6.0 and LS-DYNA R11.0. The simulation results are approximate values and may differ if other versions of LS-DYNA are used.

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:

- **xxxx.lsopt** Command input file for LS-OPT. To be created or modified by the user. The tool LS-OPTui is used to create/edit the file.

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)
job_log	The simulation run/extraction log is saved in that file in the local run directory.

1.6 Execution of LS-OPT

Graphical User Interface

Type: **lsoptui xxx.lsopt**

Batch mode

Type: **lsopt xxx.lsopt**

where **xxx.lsopt** is the name of the command file.

Remarks:

- A typical restart only requires the selection of **Normal Run** in the menu bar 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** option under **Tools**.
- LS-OPTui features a **Repair** option under **Tools** that enables the user to make changes in the design data and rerun the individual steps. Several Repair options are also available by right-clicking on individual steps such as *Optimization* or *Sampling*.

1.7 Using LS-OPTui for post-processing

Type **viewer xxxx.lsopt** in the working directory or select the **Viewer** button from the GUI.

1.8 Result interpretation

1. Simulation Plots

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

Plot simulation results without dimension restriction. Each dimension is visualized on a vertical axis and each data point is shown as a polyline connecting the respective values on the vertical axis.

d. Self-Organizing Map

Plot analysis results on a matrix with a specific number of cells with similar results grouped together in the same or nearby cells.

e. Histories

Show time history curves of simulations as well as file-based curves.

f. Statistical Tools

Interactive tools for histograms, mean, standard deviation and probability of exceeding constraints.

2. Metamodel Plots

a. Surface

- 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. 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 95% 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**

Plot Pareto Optimal solution variations in a matrix with similar solutions grouped together.

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. Then LS-PrePost can be selected to post-process a selected design.

1. **Viewing printed results:** Use **View 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

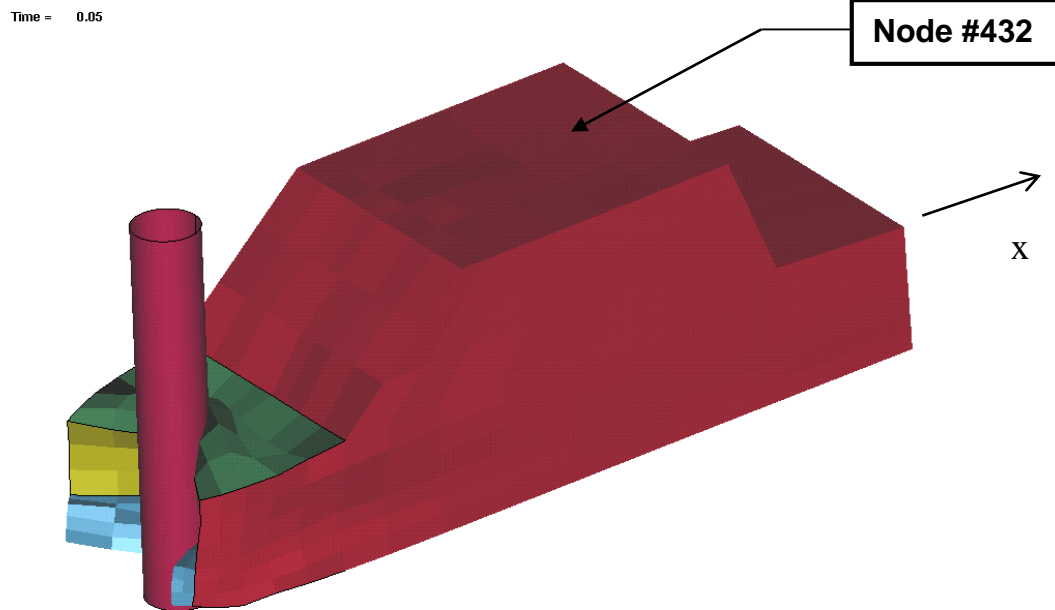
The principal reference documents are:

1. **LS-OPT User's Manual, Version 6.0**, *Livermore Software Technology Corporation, Livermore, CA, 2018*. (access by selecting “Help” in the GUI)
2. **Class Presentation notes**, *Livermore Software Technology Corporation, Livermore, CA, 2018*.

SIMPLE OPTIMIZATION AND VIEWING RESULTS

Problem description

The problem consists of a simplified vehicle moving at a constant velocity and crashing into a pole. The first figure shows the deformed vehicle after 50ms and the second one shows the part names and numbers.



Objectives of this example

- Formulating the optimization problem
- Going through the Viewer functionalities
- Interpreting the results.

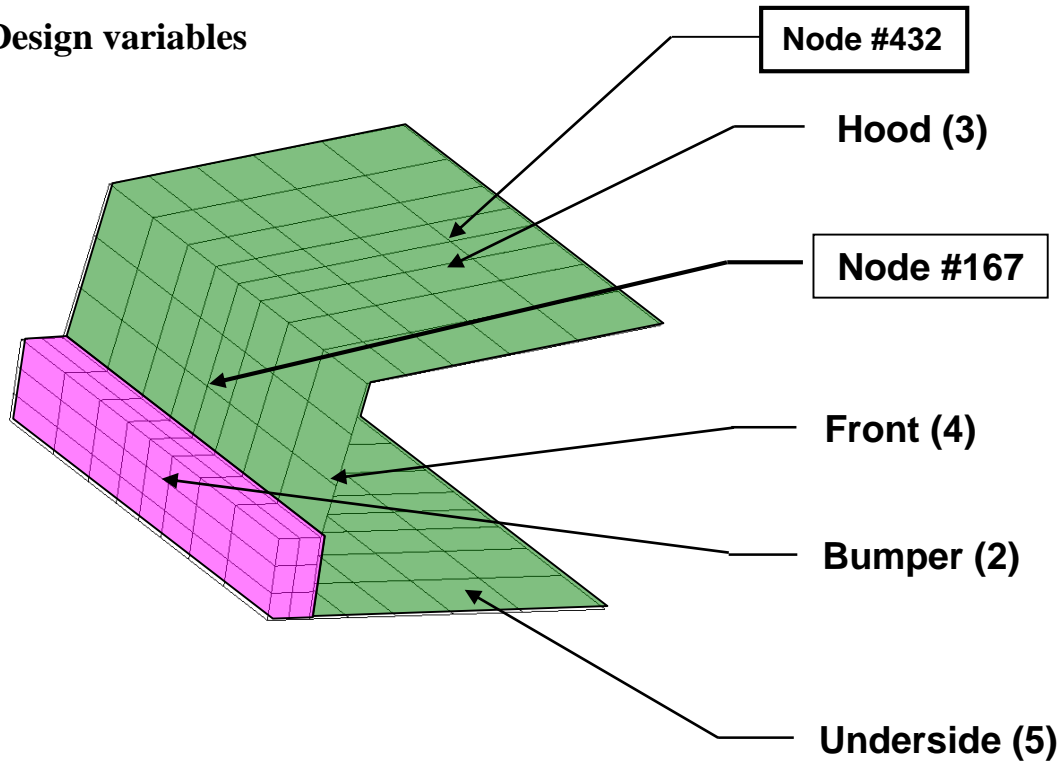
Design criteria

The quantities of interest to define the design criteria are the following:

- Head injury criterion (HIC) of a selected point (15ms) on the roof (node 432)
- Component Mass of the structural components (bumper, front, hood and underside)
- Intrusion computed using the relative motion of two points (nodes 167 and 432).

Units are in *mm* and *sec*.

Design variables



Some of the parts are assigned the same thicknesses (same variable). Thus the design variables are the following gauges:

- Hood, front and underside (**t_{hood}**)
- Bumper (**t_{bumper}**)

Note: Parts 3, 4 and 5 are grouped under a single design variable **t_{hood}** while part 2 is identified as **t_{bumper}**.

Post-processing study using the Viewer

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

Starting file: **simple.start.lsopt**

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
simple.correct.lsopt	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 **simple.start.lsopt** using the LS-OPT GUI.
2. Select View file→Other 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 the GUI flowchart to observe the flow of information, starting with setup and ending with the problem solution.

4. Inspect each component in the GUI flowchart to see the input specification fields. Confirm that you will be running a Single Iteration metamodel-based optimization with 20 design points.
5. Now, formulate the design problem using the **Optimization** component of the flowchart 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.
6. This option (single iteration) runs one iteration.
7. Double click on the LS-DYNA stage box "1" and select the LS-DYNA command that is available on your machine.
8. Run the example by selecting **Normal Run**.

Exercises:

Open the Viewer and select the following plot options:

1. **Scatter plots:**
 - 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 4 infeasible designs in the first iteration.
 - 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 entity (or color).
 - 1.5 Select "No entity" on the z -axis in order to view the sampling scheme arrangement of the points (Space Filling).
2. **Parallel Coordinates Plot:**
 - 2.1 Display the Variables, Constraints and Objectives.
 - 2.2 Find a feasible point with the lowest HIC value. Verify that `tbumper` = 2.3, `thood` = 2.0, `Intrusion` = 541.6, `HIC` = 196.2 (approximately).
 - 2.3 Find a feasible point with the lowest `Intrusion` value. Verify that `tbumper` = 4.9, `thood` = 4.9, `Intrusion` = 454.6 and `HIC` = 351.6 (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` = 3.2, `Intrusion` = 503.7 and `HIC` = 292.1.
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 “Only selected” under the Options tab).
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. Modify the transparency. 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 572.8.
6. **Metamodel Accuracy**
 - 6.1 Study and compare the PRESS values for HIC, Mass, Disp2 and Disp1.
 - 6.2 Display the PRESS Statistics. Explain the difference between the PRESS Statistics display and the ordinary predicted response value display on the Accuracy plot.
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. *Hint:* Select Neutral as color
 - 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 + 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 plotting one plot, then add a new plot and use the vertical split function.
 - 10.2 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 file" selection on the menu bar.

SETTING UP AN OPTIMIZATION RUN FROM SCRATCH

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

The files in the directory have the following meanings:

singlestage.correct.lsopt	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

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 for a single iteration

1. Confirm 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 **tbumper** parameter is labeled as follows:

```
*SECTION_SHELL
2,2,0.,0.,0.,0.,0
&tbumper,&tbumper,&tbumper,&tbumper
```

thood is defined in a similar fashion.

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

3. Open LS-OPT by clicking the lsoptui executable. Set the working directory as DESIGN_OPTIMIZATION/SIMPLE/SINGLESTAGE.
4. Enter a file name, e.g. singlestage.setup and press the “Create” button. A file named singlestage.setup.lsopt will be created in the working directory and the LS-OPT GUI will open with a problem definition template.
5. Using the LS-OPT GUI, define the required optimization problem by visiting the various flowchart components; the changes will be reflected in the .lsopt file when you save.

Step	Remark
Task and strategy selection (●●●): Choose the task as Metamodel-based Optimization with single iteration strategy.	
Stage(s): 1. Specify the dyna executable ls971_single (or any other name that you have given to the executable) and the input file main.k 2. Define the responses associated with the LS-DYNA stage. These are the x -displacement of node 432 (Disp2) and node 167 (Disp1), the HIC response evaluated at node 432, maximum x -acceleration at node 167 (use SAE filter, 60 Hz) and the combined Mass of parts 2, 3, 4 and 5.	The input file variables will be automatically detected and added as constants under Setup .
Variable Setup: Switch the constants to continuous variables under Setup . Define the size of the design space as [1,5] for each of the thickness variables.	
Sampling: The metamodel is set to RBF network and the point selection scheme is set to Space Filling. The number of points is 10.	These selections are defaulted. The samples are selected in a space with dimensions equal to the number of <i>Active Variables</i> under Sampling .

Composites: The Intrusion constraint consists of the difference between two of the responses (Disp2 and Disp1). Add a composite from the “Add” menu to define it.	A standard or expression composite is used to define the intrusion.
Optimization setup: 1. Select the objective function. 2. Select the constraint and set the bounds	The “Strict” option, available as an advanced option, can be ignored for the constraint. This is only used in special applications.
Uncheck "Do verification run" from the task menu (or delete the “Verification” component of the flowchart)	
Save the file by any name in the directory DESIGN_OPTIMIZATION/SIMPLE/ SINGLESTAGE.	
Run the project (Normal Run)	

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	2.1	.78
Disp2	-736.7	0.6	.99
HIC	68.1	40.2	.97

- 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 blue 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 (color entity) as *Feasibility*.
 - 3.3 How many infeasible designs are there in the design set?
4. **Verification run:** Set "Do verification run" (*task menu*) or add a verification run using '+'. Select **Run** (Normal 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 table in spread sheet format of the computed results of both the starting and optimum points can be obtained by selecting both (use + and click near the iterations on the graph).
 - Explore the different options "A", "M", "P" and "F" of the table spread sheet.

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.
<i>Tbumper</i>	3		3.4	
<i>Thood</i>	1		1.7	
Mass	0.41	0.41	0.58	0.58
Intrusion	575.7	571.8	550.1	550.1
HIC	68	34.6	160	154

Max. Constr. violation	25.7	21.8	0.1	0.1
------------------------	------	------	-----	-----

5. **Design Sensitivities:**

5.1 Study the final design sensitivities in **View file→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	-3.03	-0.38
HIC	+15.6	+1.7

Add a constraint and repair the optimization run

Directory: **DESIGN_OPTIMIZATION/SIMPLE/SINGLESTAGE** (same as previous)

Starting file: Use the file created in the previous section.

The purpose of the example is to add a constraint to the design without rerunning the simulations. You should have a database of 10 runs generated previously. The run will be done in the same project directory.

Setup:

1. Add a Mass upper bound constraint of 0.5
 - 1.1 Repair the optimum using the Optimize repair option (right-click the Optimization box). Make sure that the first iteration is repaired and not the verification iteration 2 (change the iteration number in the iteration box, on the right side of the task button).
 - 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 Clean from iteration 2 (under Tools) and run a metamodel-based optimization to produce a *new* verification run. Tabulate the results:

	Start		Optimum	
	Comp.	Approx.	Comp.	Approx.
<i>Tbumper</i>	3		1.37	
<i>Thood</i>	1		1.83	
Mass	0.41	0.41	0.5005	0.5005
Intrusion	575.7	571.8	547.10	556.08
HIC	68	34.6	226.7	220.94
Max. Constr. violation	25.7	21.8	0.0005	6.08

FINDING A CONVERGED SOLUTION

Directory: **DESIGN_OPTIMIZATION/SIMPLE/ITERATE**

Starting file: **iterate.start.lsopt**

Design Formulation

Minimize

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

subject to

Mass < 0.5

Intrusion (50ms) < 550mm

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

Setting up the problem

1. In the **Task selection** menu, select the SRSM strategy.
2. Use the **Sampling** and/or **Metamodel Building** page to confirm that the linear polynomial metamodel and *D*-Optimality sampling criterion have been selected. *Select 5 points per iteration.*
3. Select the Hybrid ASA (ASA with switch to LFOP) as core **optimizer algorithm**.
4. For **Termination criteria**, select a tolerance of 0.001 to be satisfied by the design **and** objective changes.
5. Select 10 iterations and run the problem.
(*Don't forget to set your LS-DYNA command before running the problem.*)

Exercise

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 10 points for a Single Iteration run (run in the previous example: Add a constraint and repair the optimization run).

	<i>Single Iteration</i> (10 simulations)		<i>SRSM</i> (51 simulations)	
	Comp.	Approx.	Comp.	Approx.
<i>tbumper</i>	1.37		1.58	
<i>thood</i>	1.83		1.77	
Mass	0.500 5	0.500 5	0.500 05	0.500 5
Intrusion	547.1 0	556.0 8	550	550
HIC	226.7	220.9	239	240
Max. Constr. violation	0.000 5	6.08	0.000 05	0.000 5

2. Accuracy.

- 2.1 Study the **Optimization History Plot**. Comment on the accuracy of the HIC response (on the **Value** plot of HIC).
 - 2.2 What is the RMS error trend of HIC response?
 - 2.3 What is the R^2 trend of HIC response?
 - 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 Iteration and iterative solutions as far as utility and accuracy is concerned.

DISCRETE OPTIMIZATION

Single iteration

Directory: DESIGN_OPTIMIZATION/SIMPLE/DISCRETE

Starting file: **discrete.start.lsopt**

Design Formulation

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. 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 “Continuous” to “Discrete variable” and “Adding new values” **1, 2, 3, 4, 5** to define the set of possible values. Use **1** as a starting variable value.
2. Set the *Sampling Type* for the variable thood to **Discrete**.
3. Make sure that the only constraint is Intrusion < 550.
4. On the **Sampling** page, select 20 points.
5. Set the optimizer to Hybrid ASA (with switch to LFOP).
6. Conduct a Single Iteration run.

(Don't forget to set your LS-DYNA command before running the problem).

Exercise:

1. Compare the predicted results of the continuous (first example: SIMPLE OPTIMIZATION AND VIEWING RESULTS/VIEWER) and discrete analyses.

	Continuous	Discrete
tbumper	3.52	3.20
thood	1.57	2
HIC	132.72	196.2
Mass	0.57	0.65
Intrusion	550.06	538.02

2. Study a tbumper-thood plot of the surface and points to get an impression of the *discrete space filling* point selection scheme.
3. In the metamodel surface plot, select HIC with the Constraints option to view the optimal design in the feasible region.

USER-DEFINED OPTIMIZATION PROBLEM

Directory: **DESIGN_OPTIMIZATION/USER_DEFINED**

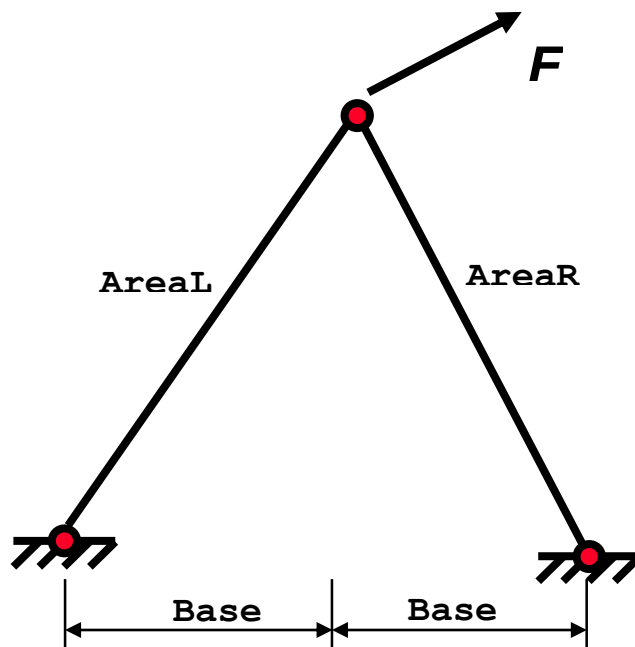
Special topics: LS-OPT parameter type, Neural nets, Pareto optimality, dependent variables

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: $F_x = +24.8\text{kN}$, $F_y = 198.4\text{kN}$.

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



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. Open the LS-OPT GUI, set the working directory, select names for the file, sampling and stage, and click Create.
2. *Stage Setup*: Select the **User-defined** solver option under stage setup. Enter the command as **perl**. Browse for the input file name: **2bar**.
3. *Setup*: Select the variable lower and upper bounds Base:[0.1,1.6], AreaL:[0.2,4], AreaR:[0.2,4]. Change the variables starting value to Base: 0.8, AreaL: 2, AreaR: 2.
4. *Stage Responses*: Enter the responses using FILE response option. The solver dumps the results into individual files: **Weight**, **StressL**, **StressR**, therefore the FILE response option can be used to read the values by defining the filename in the response dialog. USERDEFINED response option can also be used to read values from the files. In this case, the command must write the value in the file to standard output, e.g. **cat Weight** (Linux) or **type Weight** (Windows).
5. *Optimization*: Minimize the Weight. Bound the stresses from above and below as [-1;1] in both bars.

Exercises:

1. *Strategy*: Set the **Strategy** (under Task selection) to **SRSM**.
2. *Run*: Perform a baseline 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	2.56

3. Run 10 iterations using a linear approximation and the *D*-Optimality criterion for point selection and tabulate the optimal variables and responses:

	Predicted	Computed
Base	0.18	–
AreaL	1.7	–
AreaR	0.33	–
StressL	1	0.99
StressR	1	1
Weight	1.03	1.02

Reducing the number of variables by constraining the bar areas

Starting file: `userdef.constrained.start.lsopt`

Directory:

`DESIGN_OPTIMIZATION/USER_DEFINED/
CONSTRAINED_VARS_LINEAR`

Setup:

1. Choose the SRSM strategy.
2. Remove the range values so that the problem starts running from the full design space.
3. Change **AreaR** to a **Dependent Variable** with definition **AreaR = AreaL/2.0**. (Just type in **AreaL/2.0** in the definition box).
4. Save the modified data under any name.
5. Run 5 iterations.

Exercises:

1. Tabulate the optimal variables.

	Computed	Predicted
AreaL	1.45	
AreaR	0.73	
Base	0.42	
StressL	0.96	1.0
StressR	1.04	0.96
Weight	1.18	1.19

2. Do you think the solution has converged?

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 9.5. 3. Importing user-defined analysis results.

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. Specify a project directory, file, sampling and stage name and click Create.
3. *Task selection menu*: Choose Metamodel-based Optimization task Single Iteration strategy.
4. *Sampling*: Browse for the "AnalysisResults.csv" file using the "Import - Analysis Results" option under Sampling Features. Verify that the selected metamodel is RBF network.
5. *Variables and Responses setup*. Check the variables (under setup) and responses (under stage(s)).

6. Adjust the variable bounds to [1, 5].
7. *Menu bar.* Click Tools→Repair→Import Results. You can also right-click on Sampling and then Repair→Import Results.
8. View the Summary Report.
9. Optimization:
 - a. Define HIC as the objective.
 - b. Use a Composite-Expression to define the constraint:
$$\text{Intrusion} = \text{Disp1} - \text{Disp2} < 550\text{mm.}$$
10. In the Task selection menu, uncheck "Do verification run" (or simply delete the Verification box). Run the project (Normal 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? (340)
 - b. What are the HIC and Intrusion values for the design with the lowest Max. Acceleration? (412, 484)
2. Using the Parallel Coordinate Plot, compare the best simulated design (i.e. from the imported table) with the predicted optimum.

	Simulation	Predicted Optimum
tbumper	3 . 69	3 . 9
thood	1 . 69	1 . 6
HIC	1 68	1 66
Mass	0 . 61	0 . 60
Intrusion	548 . 11	550

DIRECT OPTIMIZATION

Directory: **DESIGN_OPTIMIZATION/DIRECT/SIMPLE**

Starting file: **direct.start.lsopt**

The example is the same as the small car pole crash (SIMPLE OPTIMIZATION AND VIEWING RESULTS)

Design Formulation

Minimize

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

subject to

Mass < 0.5

Intrusion (50ms) < 550mm

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

Setting up the problem

1. Open **direct.start.lsopt** in the GUI and change the task to Direct Simulation→Optimization.
2. Verify that the Objectives and Constraints are set correctly.
3. In the **Optimization→Algorithms** page, set the population size of the Genetic Algorithm (GA) to 20 and the number of generations to 20. The changes will be reflected in Sampling and Termination Criteria. Set max repeat optimum/generation to 0.5.
4. Save the input to a new file name.
5. Run the direct optimization.

(Don't forget to set your LS-DYNA command before running the problem).

Exercise

1. Compare the results of metamodel-based (iterative) and direct methods.

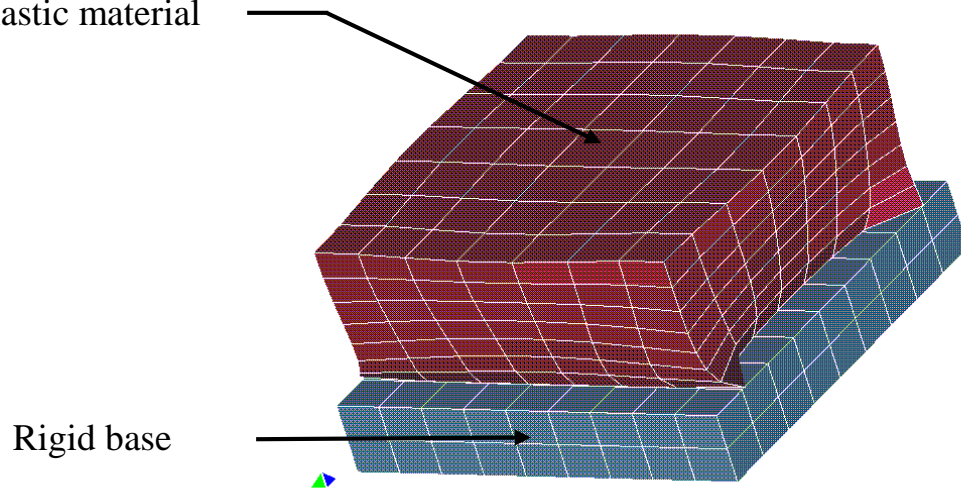
	<i>SRSM</i> (51 simulations)		<i>Direct</i> (400 simulations)	
	Comp.	Approx.	Comp.	Approx.
<i>Tbumper</i>	1.58		1.13	
<i>Thood</i>	1.77		1.75	
Mass	0.5	0.5	0.46	
Intrusion	550	550	549.5	
HIC	239	240	174	
Max. Constr. violation	0	0	0	

MATERIAL PARAMETER IDENTIFICATION

Directory: **PARAMETER_IDENTIFICATION**

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

Elastoplastic material

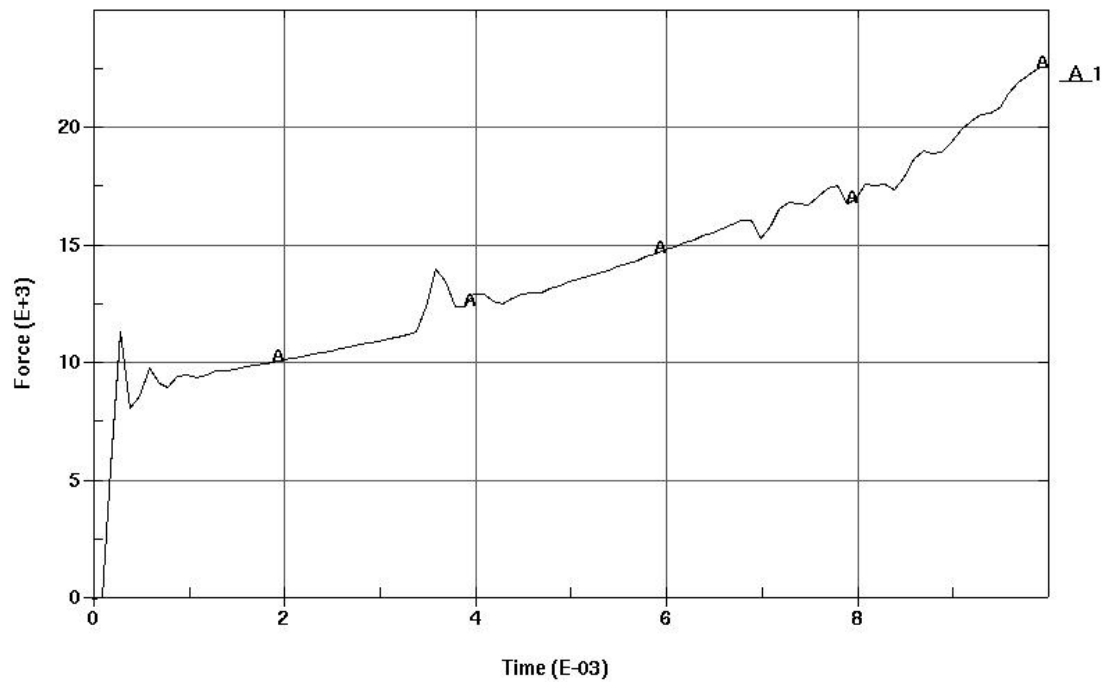


Problem description

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 history-based Mean Squared Error composite function.
 - using a point-based Mean Squared Error composite function.
- How to construct crossplots.
- How to use multiple simulation models in the same optimization problem (multi-case).

Ordinate-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. The example must be set up from scratch.
2. Select the SRSM strategy.
3. Specify **foam1.k** as the input file.
4. Use the starting values and variable bounds as tabulated below (YMod is the Young's modulus and Yield is the yield stress).

Variable	Starting Value	LB	UB
YMod	700,000	5e5	2e6
Yield	1,500	5e2	2e3

5. Go to the "Histories" panel under Stage:
 - 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*.
 6. Add an ordinate-based Curve Matching composite called **MSE**. Select the algorithm as Mean Square Error.
 7. Using “*add new file history*”, assign **Test1** as the target curve. Assign **F_vs_d** as the computed curve.
 8. Select **MSE** as the objective.
 9. Run the example with an iteration limit of 3.
- (Don't forget to set your *LS-DYNA* command before running the problem).

Exercises:

1. View **Optimization**→**History** and the **MSE** composite (multi-objective).
2. View the optimal curve matching by selecting **Histories** under **Simulations** and then selecting **F_vs_d** and **Test1**.
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. In the Viewer, also view **Optimization**→**Variables** for the 95% 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	95% Confidence Interval	
		Lower	Upper
YMod	2e+006	-16e6	20e6
Yield	1018	913	1123

Ordinate-based MSE: multiple cases

Directory:

PARAMETER_IDENTIFICATION/MSE_HISTORY/MULTICASE

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

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 a stage Stage2 with input file foam2.k under the same sampling. This can be done by cloning the existing stage and modifying the new stage.
2. Stage2 Histories tab:
 - a. Change the name to Disp2, the z-displacement at node 288
 - b. Change the name to Force2, the z-reaction slave force at interface 1
 - c. Add the crossplot of these values as F2_vs_d2 (-Disp2 vs. Force2)
 - d. Add Test2 as an imported file history from Test2.txt
3. Add a mean square error curve matching composite MSE2 using Test2 and F2_vs_d2
4. Add MSE2 as an objective
5. Run the example with an iteration limit of 4.

(Don't forget to set your LS-DYNA command of your two stages before running the problem).

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. These can be viewed together with the histories of Stage 1 using the “Multi” option.
4. Note down the optimal values and confidence intervals for the two parameters (**View→Summary Report**, scroll to the bottom):

Parameter name	Value	95% Confidence Interval	
		Lower	Upper
YMod	1.0e6	7e5	1.37e6
Yield	1018	912	1124

5. Compare the confidence intervals to those of the single case.
6. Study the force vs displacement curves for the two cases. Is it okay to use these together (are these for the same material)?

Point-based Mean Squared Error (optional)

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. Open LS-OPT GUI with any name for the file, sampling and stage.
3. *Task selection menu*: Select Metamodel-based Optimization with the SRSM strategy.
4. *Sampling or Metamodel Building*: Use the default settings - linear polynomial approximations.
5. *Stage setup*: Define the single precision LS-DYNA solver ls971_single with foam1.k as the input file.
6. *Setup*: Use the starting values and variable bounds as tabulated below.

Variable	Starting Value	LB	UB
YMod	700,000	5e5	2e6
Yield	1,500	5e2	2e3

7. Stage histories and responses: Extract the Z-slave force history **Force** from **RCFORC**, with interface ID = 1. Use response expressions to compute the forces, e.g. **Force(0.002)**.
8. *Composite*: Add an MSE composite called **MSE** from the table above. (The composite to be selected is the “standard” type in which Target values can be defined, not a Composite-Expression. Assign targets for each point based on the Force-Displacement experimental results provided below.

Displacement	Force Resultant
0.36168	10162
0.72562	12964
1.0903	14840
1.4538	17672

9. *Optimization*: Select **MSE** as the objective function to minimize.

10. 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	2e6
Yield	1003
MSE	2e6

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. (Add Global Sensitivity and run).

3. Confidence Intervals:

3.1 Study the table of 95% 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). Click on a selected bar in the bar chart.

	Value	Lower bound	Upper bound
YMod	2e6	-3e7	3e7
Yield	1003	804	1202

3.2 Comment on the confidence levels of the individual parameters.

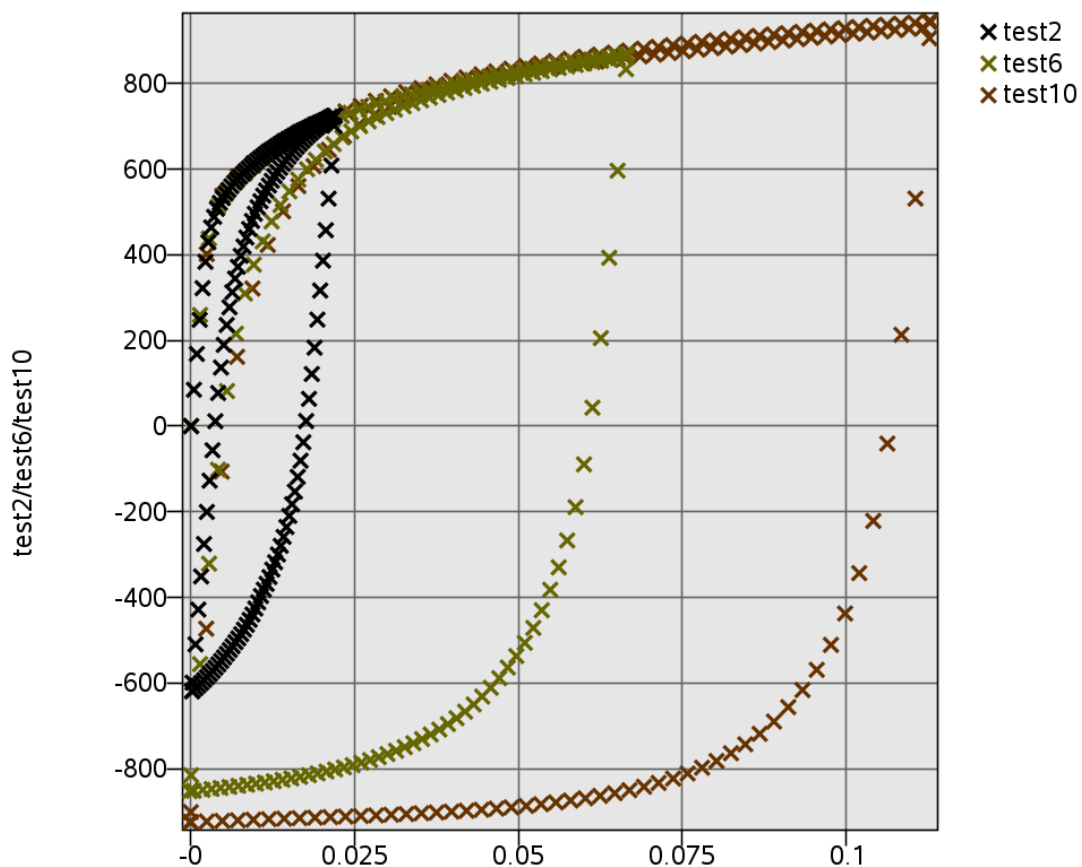
Parameter identification with hysteretic response: multiple cases

Directory:

PARAMETER_IDENTIFICATION/CURVE_MAPPING

The Bauschinger effect is significant for automotive sheet steels. The phenomenon is observed under cyclic loading which results in a hysteretic stress-strain curve. The nature of the hysteretic curve complicates the curve matching required to identify the material parameters and therefore an approach which is more sophisticated than the ordinate-based matching is required. For this purpose, a Curve Mapping algorithm is used.

The following example consists of three load cases, each representing a different cyclic loading range as illustrated in the stress-strain diagram in the figure below. The material is defined by 9 parameters.



Starting file: **mat125.calibration.start.lsopt**

Setup:

Modify the starting file as follows:

1. Histories tab of each stage: Study the definitions of the stress and strain histories. (*Don't forget to set your LS-DYNA command of your stages before running the problem*).
2. Select deletion of d3* files in all run directories (Stage→File Operations→Add→Delete). The d3plot and other database files will be deleted as a consequence.
3. Composites: Change each of the Composite Functions Residual2, Residual6 and Residual10 to *Curve Mapping* types instead of *Mean Square Error* types.
4. Sampling: Confirm that all the stages *share the same sampling or have identical samplings*.
5. Optimization: Construct an objective by adding the residual components together with equal weights.
6. Run the example with an iteration limit of 3.

Exercises:

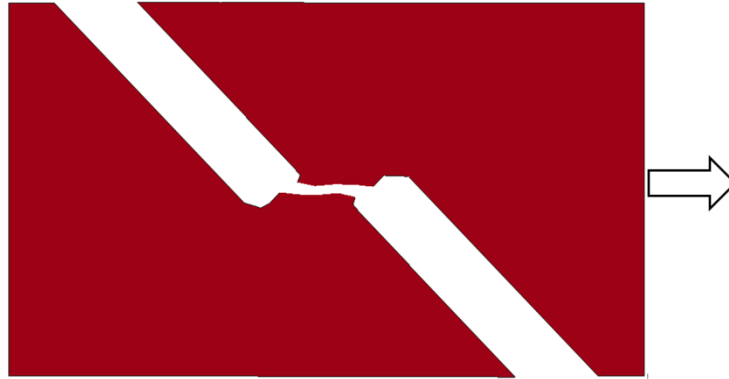
1. Study the Optimization History of the total residual. Also look at the individual residual components.
2. In the **Simulations**→**Histories** plot selection, select the *Multi* option to view all the stress-strain crossplots as well as test results simultaneously. Select the *Neutral* color option (4th column in the control panel, very last option) in order to distinguish between the load cases.
3. Under the **Options** tab in the control panel, select *Only Optimal* and compare the results of Iteration 1 and Iteration 4. *Hint: Select All iterations and use Iteration as color option*.
4. Add *Global Sensitivities* in the process flow and run the example again. *Global Sensitivities* option can be defined either through 'Add' menu or from the task menu. *Note: To prevent rerunning of the simulations, do not run a clean start!*

5. Select the **Sensitivity** option in the Viewer (under **Metamodel**). Choose the *Multi* option to display all the residuals. Which parameter appears to dominate the matching ability?

GISSMO FAILURE MODEL CALIBRATION (SHEAR LOAD CASE)

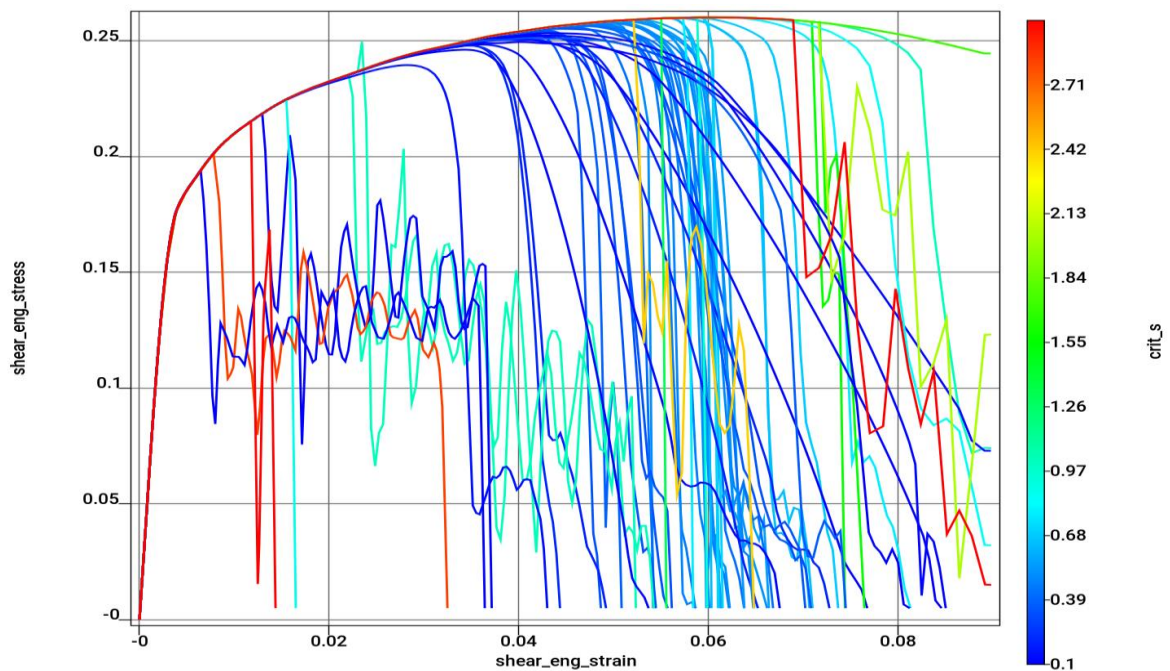
Directory:

PARAMETER_IDENTIFICATION/GISSMO/SHEAR



Problem description

Curve matching and parameter estimation of a failure model requires special consideration due to noisy nature of the curves resulting from element erosion due to failure and oscillations of the specimen after failure. Therefore, the curve matching algorithm needs to accommodate the noise and steep drop in the curves due to failure, as shown in the figure below.



In this example, Dynamic Time Warping (DTW) curve measure is used to match the noisy stress-strain curves of a specimen in shear load using GISSMO failure model and synthetic test data. Since the DTW measure requires the matching curves to have equal length, the computed stress curve is truncated based on the final stress value of the test data. Critical plastic strain and equivalent plastic strain to failure are considered as design parameters and rest of the material parameters are treated as constants, simply to reduce the problem size.

New Points illustrated by this example:

- How to do parameter identification for failure models.
- How to define the Scaled Dynamic Time Warping distance measure.
- How to use truncated curves in curve matching.

Starting file: **GISSMO_shear.start.lsopt**

Setup:

1. Open LS-DYNA input file in LS-PrePost and investigate the model, loading and boundary conditions.
2. Use GUI to open the start file **GISSMO_shear.start.lsopt** and investigate the optimization strategy, design variables, and histories defined the file.
3. Make sure stress-strain histories and stress vs strain crossplot is defined correctly.
4. Open the file **shear_exp.dat** containing experimental stress vs. strain data and note down the last stress value. This value is used along with computed stress curve to extract the corresponding time value.
5. Define a response to extract this time value using lookup function response and name this response as *shear_failtime*. In the lookup response definition, select *shear_eng_stress* as history and enter 0.05 (last stress value of experimental curve) in the value field. Hint: Enter a start time of 0.5 so that initial part of the curve is ignored during the lookup search.
6. Define a new crossplot history of stress vs strain with time '0' to '*shear_failtime*'. Name this history as *shear_sig_eps_trunc*. This crossplot is the truncated stress vs. strain history. Note: GUI does not accept response names in the time fields. However, you can define a

- number and replace it with the response name by editing the project file (*.lsopt file) using a text editor.
7. Define a curve matching response for truncated stress vs. strain history (*shear_sig_eps_trunc*) and experimental stress vs. strain (*shear_exp*) using Dynamic Time Warping algorithm. Use 100 interpolated points as points for regression.
 8. Select the DTW response as objective for minimization.
 9. Run a baseline run to verify the setup. After completion, open viewer and plot the truncated history and compare it with the full stress vs strain history to verify that the curve has been truncated correctly.
 10. Now run the example with iteration limit of 10.

(Don't forget to set the proper LS-DYNA command before running the problem).

Exercises:

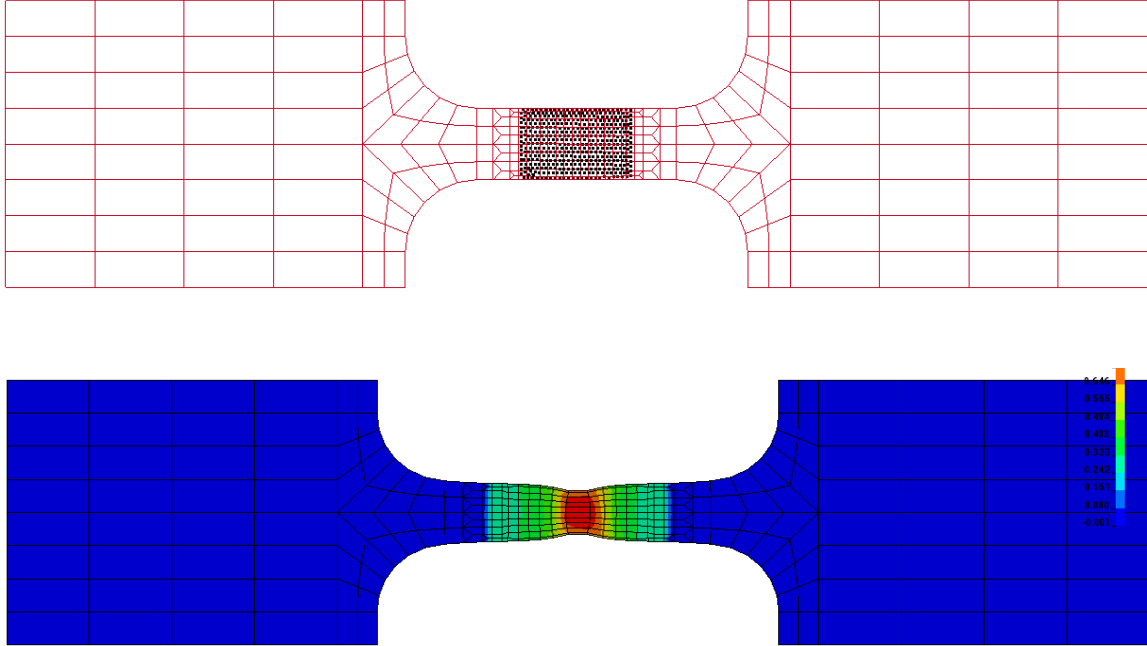
1. Study the Optimization History of residual (DTW response). Also look at the optimization history of the design variables. Did the optimization converge?
2. View the optimal curve matching by selecting **Histories** under **Simulations** and then selecting **shear_sig_eps_trunc** and **shear_exp**.
3. Using the same plot, under the **Options** tab in the control panel, select *Only Optimal* and compare the results of Iteration 1 and Iteration 10. *Hint: Select All iterations and use Iteration as color option.*
4. Confirm the optimal parameters and DTW residual:

Parameter/Response	Value
crit_s	0.443
eps_s1	1.602
eps_s2	1.046
DTW residual	0.014

FULL-FIELD CALIBRATION

Directory: **FULL_FIELD**

Special topics: Multi-point histories, distance measures



Problem description

The material parameters of a steel material yield curve must be determined from experimental results. The results are derived from optical results obtained from a Digital Image Correlation experimental setup. The full field experimental data (which includes displacements and strains) are available at a number of points near the center of the coupon as depicted in the figure above (463 black points). The picture at the top shows the FE mesh with the experimental points aligned. The bottom picture shows the ε_{xx} strain resulting from the DIC experiment (superimposed on the FE mesh).

The calibration problem is addressed by minimizing the residual tensile force vs. ε_{xx} at the experimental points. Four parameters: c , n , b and d , in the Hockett-Sherby formula:

$$f(\varepsilon_p) = d - be^{-c\varepsilon_p^n}$$

need to be calibrated. A pre-processor script (`yieldcurve.py`) is used in a pre-processor stage (called *YieldCurve*) to produce the yield curve in a `LOAD_CURVE` include file.

New Points illustrated by this example:

- How to define and preview a file multi-point history.
- How to define a multi-point history.
- How to do parameter identification using full-field experimental results.
- How to define the Scaled Dynamic Time Warping distance measure.
- How to interpret the experimental and computational fields and the difference between them.

Dynamic Time Warping

Directory:

PARAMETER_IDENTIFICATION/FULL_FIELD

Problem description:

The following files are provided:

- | | |
|-------------------------|---|
| ▪ yieldcurve.py | Yield curve pre-processor (python script) |
| ▪ Simulation.dyn | LS-DYNA keyword file |
| ▪ MFz_GOM.xml | DIC data output file (XML format) |

Setup:

1. Open the file `dic.start.lsopt`
2. Select the stage: **TensileTest**
3. **Experimental curve**. In the *Multihistories* tab, create a File Multihistory crossplot of force vs. strain (button at bottom right of dialog) as follows:
 - a. Provide a name for the file multihistory, e.g.
force_vs_strain_exp
 - b. The experiment was conducted using the ARAMIS DIC system. Select *ARAMIS* as the format or type of experimental data

- c. Select the `MFz_GOM.xml` file as the input data file provided by the ARAMIS DIC experiment.
 - d. Select the X-component as *SurfaceComponent.epsX*
 - e. Select the Y-component as *Force.DIM*
4. **Computational curve.** In the *Multihistories* tab, also create a **D3PLOT** multihistory as follows:
 - a. Select *Strain* → *M_surf_xx_strain*
 - b. To import the point coordinates (provided in the DIC data), select the name assigned to the file multihistory (e.g. **force_vs_strain_exp**).
 - c. Select the FE interpolation type as **Element**. This means that each experimental point will be mapped to a precise point within an element and the strain will be interpolated at that point.
 - d. First display the unaligned point set. Select "(no alignment)" and select "Preview in LSPP".
 - e. Alignment data (group name `align` in this example) already exist in the form of coordinates or IDs in the starting file. To view the points after alignment, select "Alignment → `align`". Preview again.
 - f. Name this multihistory **strain_comp**.
5. In the *Multihistories* tab, create a crossplot by name **force_vs_strain_comp**. This is done by selecting the previously defined **strain_comp** as the x -axis ($z(t)$) and the force **negForce1000** as the y -axis ($F(t)$) to create the crossplot $F(z)$.
6. Next, a response is created for finding the discrepancy (or distance) between the experimental field **force_vs_strain_exp** and the computational field **force_vs_strain_comp**.
 - a. In the *Responses* tab, select the *Curve Matching* option in the menu on the right.
 - b. Select *Multihistories* for the comparison.
 - c. Select the *Dynamic Time Warping* distance measure to quantify the distance between experiment and computation.
 - d. Select the target multihistory as the above created **force_vs_strain_exp**.
 - e. Select the computed multihistory as **force_vs_strain_comp**.
 - f. Select *using the regression points from the target curve*.
 - g. Provide a name for the distance measure, e.g. **Residual**.
7. Select **Residual** as the objective function.

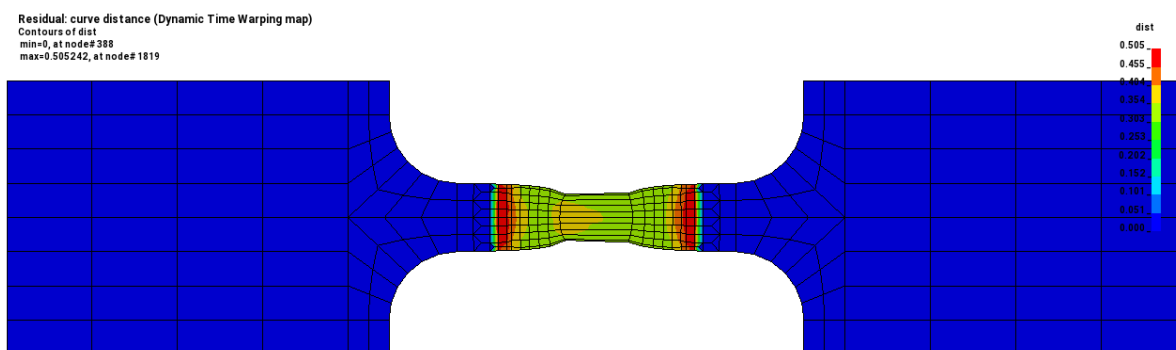
8. Conduct a baseline run to validate the setup (approximate running time: 6 min.).

(Don't forget to set the proper LS-DYNA command before running the problem).

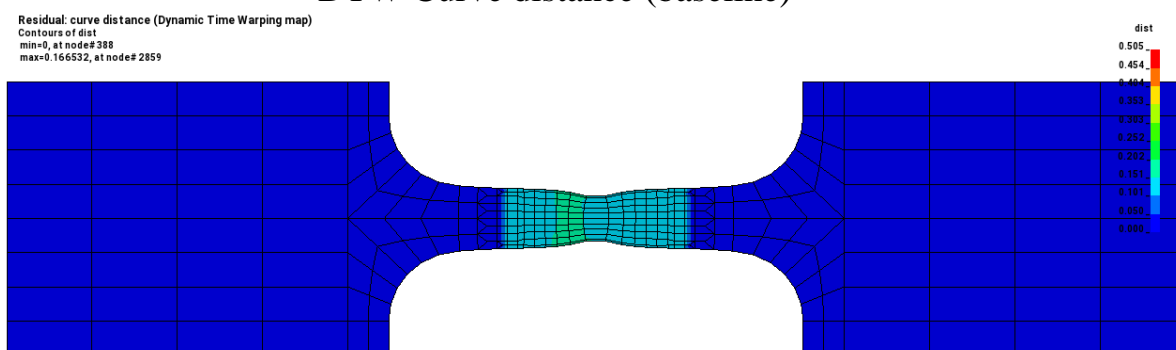
Exercises:

1. Select the **Scatter Plot** and then select the single point appearing on the plot. A table dialog window showing the baseline data will open.
2. Select **PP** (post-processor) to start LS-PrePost. Then select **LS-PrePost Fringe Plot→Residual→Target x** to display the contour plot of the experiment as superimposed on the FE mesh. Also select **Computed x** and **Difference x** . The x denotes that we are displaying the strain, which is the x -component of the crossplot (y represents the force). Make a note of the range of the **Difference x** [-0.23,+0.6].
3. View **PrePost Fringe Plot→Residual→Curve distance**. What does this represent?
4. Make a note of the baseline value of *Residual* [0.37].
5. Run an optimization with an iteration limit of 4 (i.e. select *Normal Run*).
6. Select the Optimization History to view the objective value and click on the last point (Iteration 4). Select the **Difference x** plot and note the strain range [-0.08,+0.2]. Then set the bounds to be the same as for the baseline ([-0.23,+0.6]). (In LSPP this is done by selecting **FriRang→User**, editing the *Min* and *Max* values and then selecting **Update** and **Done**). Compare the strain difference at the optimum to the baseline difference.
7. Compare the optimal distance value to the baseline distance (i.e. the value of the objective called *Residual*).

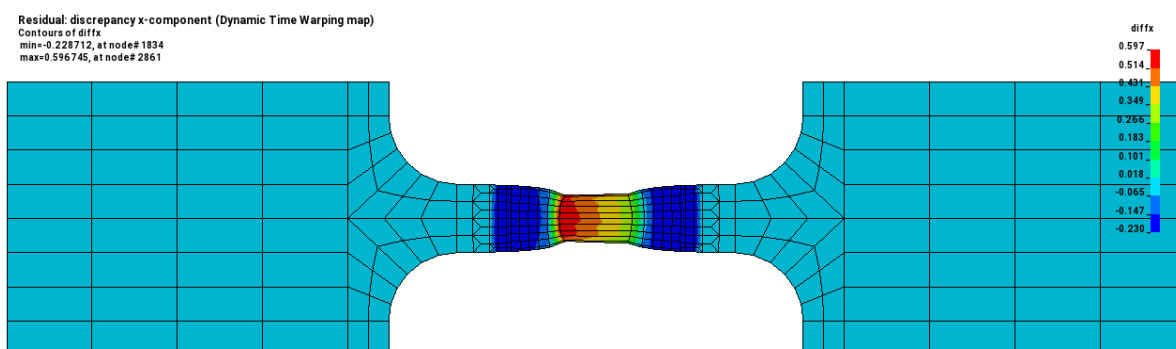
Baseline	Optimum (Iteration 4)
0.37	0.14



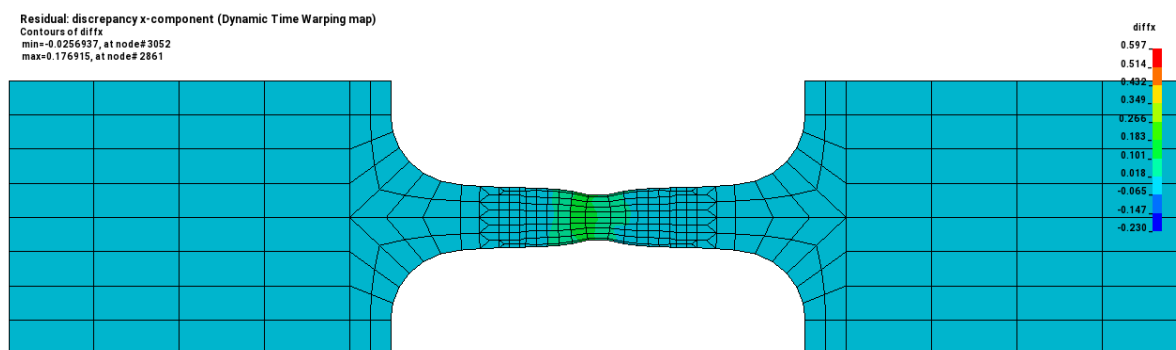
DTW Curve distance (baseline)



DTW Curve distance (iteration 4)



Strain difference (baseline)



Strain difference (iteration 4)

MODAL ANALYSIS AND TRACKING

Directory: **DESIGN_OPTIMIZATION/MODAL_ANALYSIS**

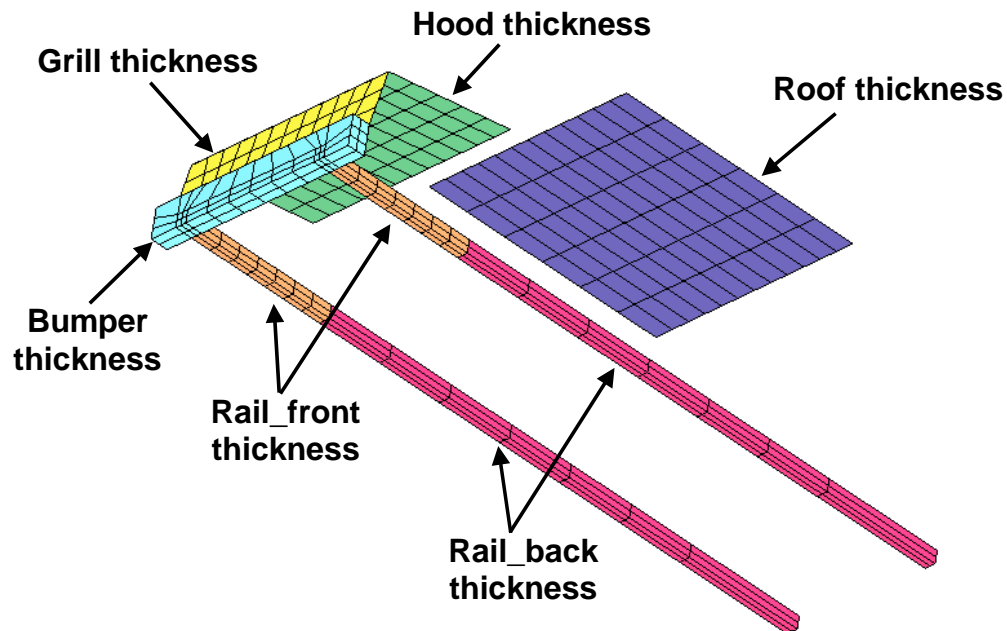
Special topics: Mode tracking

Problem description

This example illustrates the following new features:

- Running an LS-DYNA implicit (double precision) solver.
- 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 (\mathbf{x}) of the parts 2, 3, 4, 5, 6 & 7.

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

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

Other data:

- Input file: `car6_NVH.k`
- Solver: `ls971_double` (ls-dyna double precision)

Identifying the mode

Directory:

DESIGN_OPTIMIZATION/MODAL_ANALYSIS/DOE

Setup:

Creation of a new LS-Opt project:

1. Open LS-Opt and create a new project under any name.
2. In stage setup, import the input file.

Identifying the Torsional mode:

1. In the stage setup, define the double precision version of LS-DYNA (`ls971_double`). Add the mass response.
2. Set the bounds of the design variables in **Setup** [1, 6].
3. Select 1 as the "Baseline Mode Number" for the frequency extraction under the **Responses** tab of the stage definition. We have yet to identify the Torsional mode number, so "1" is as good a number as any.
4. Set up the design optimization problem as defined above.
5. Using the DOE Study task, run the baseline case. Save the file under any name.
6. 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. For the baseline run, identify a *pronounced* twisting mode with the lowest mode number.

DOE task:

1. Enter the previously identified mode number (mode 10) as the "Baseline Mode Number" for the frequency response.
2. Add the **New Mode Number** and **MAC** as two additional FREQUENCY responses in addition to Mass and Frequency. Visualization of the mode tracking is aided by selecting these responses.
3. Use a linear approximation with default settings. Save the work under any name.
4. Clean the existing results using the **Tools** menu. *This is important since responses have been added which changes the design problem.*
5. **Run** the project.

Exercise:

1. Which mode numbers are found when the twisting mode is tracked for the 10 different designs. (8, 10 and 11)
2. What is the lowest value of MAC found for the 10 different designs and for which mode (0.66; 8).
3. What is the highest value of MAC found for the 10 different designs and for which mode (1; 10).

Optimization

Directory: **DESIGN_OPTIMIZATION/MODAL_ANALYSIS/ITERATE**

Starting file: **frequency.iterate.start.lsopt**

Setup:

1. Verify that the strategy is SRSM and check the Sampling 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.

(Don't forget to set your LS-DYNA command of your stages before running the problem).

Exercise:

1. Study the optimization history of the objective and constraint.
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 MAC value?

	Value
tbumper	3 . 87
troof	1
trailb	1 . 2
trailf	1
tgrill	1
Mass	0 . 74
Frequency	1 . 7
Mode	11
MAC	0 . 99

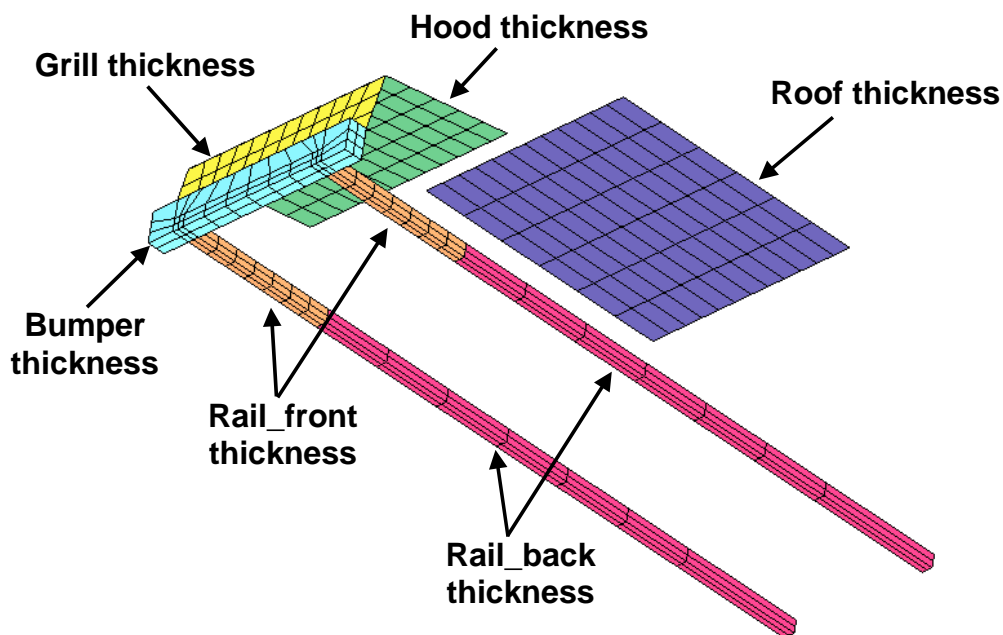
MULTIDISCIPLINARY OPTIMIZATION

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 ($\mathbf{x}_{\text{crash}}$)

subject to

HIC($\mathbf{x}_{\text{crash}}$) < 900

Intrusion($\mathbf{x}_{\text{crash}}$) < 500mm

Torsional mode frequency(\mathbf{x}_{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: First iteration with all variables

Directory: **DESIGN_OPTIMIZATION/MDO/ITERATE**

Starting file: **mdo.iterate.start.lsopt**

Setup:

1. Use the GUI to open the **mdo.iterate.start.lsopt** file.
2. Check the availability of parameters from the input files by clicking "Stage Matrix" and "Sampling Matrix" under **Setup**, "Active variables" under **Sampling** and "Parameters" under the **Stages**.
3. Use all variables, except **tgrill**. (Deselect **tgrill** for both cases).
Note: tgrill can also be set to a constant in the Setup dialog.
4. Constraints:
 - a. Set the frequency constraint as an *equality* constraint of 1.8 by setting both the lower and upper bounds to 1.8.
 - b. Add "Intrusion" constraint with an upper bound of 500.
 - c. Set the internal constraint scaling (check box in the Constraints tab). 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 significantly smaller numbers tend to be ignored.
5. Use '+' in the top menu bar to add the Global Sensitivities calculation.
6. Change the task selection to Metamodel-based optimization with SRSM strategy, set the termination criteria to one iteration and Run.

(Don't forget to set your LS-DYNA command of your stages before running the problem).

Exercise:Viewer: Using the *Global Sensitivities* selection, estimate the 3 most important variables of the CRASH discipline.

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

Iterative optimization using screened variables

Directory: **DESIGN_OPTIMIZATION/MDO/ITERATE**

Starting file: **mdo.iterate.start.lsopt**

Setup:

1. Identify the 3 most important variables of the crash discipline from the previous LS-OPT run of one iteration (trailb, troof, tbumper). Note that the LS-OPT input file of the previous run can be modified to continue optimization with the screened variables.
2. Sampling: Using the Sampling Matrix in Setup, select the 3 most important variables of the CRASH discipline (recorded in the previous section) as active variables for the crash sampling by deactivating the insignificant variables. Use all the variables for NVH.
3. Set the number of iterations to 5 (or more, time allowing) and continue the optimization process with the screened variables.

Exercise:

1. *Viewer:* Study the feasibility of the solution. Which response has the largest normalized constraint violation (violation divided by bound value)?
2. Use the 2D Interpolator to study the sensitivity of all functions to a variable which is only present in the NVH case. What do you see?

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

$$\text{HIC (15ms)} < 250$$

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: **direct.moo.start.lsopt**

Setting up the problem

1. Select Mass and Intrusion as objectives to define a two-objective optimization problem. De-select all other responses.
2. Set a HIC upper bound constraint of 250. De-select all other constraints.
3. Select the task as *Direct Simulation->Optimization* and check the box “Create Pareto Optimal Front”. The Pareto Optimal option can also be turned on from the objective tab under Optimization. *Note that the “Create Pareto Optimal Front” option appears only if multiple objectives are defined.*

4. On the **Optimization**→**Algorithms** page, set the population size to 20 and the number of generations to 20. Set the termination criterion to *Hypervolume change*.
5. Run.
(Don't forget to set your LS-DYNA command of your stage before running the problem).

Exercise

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

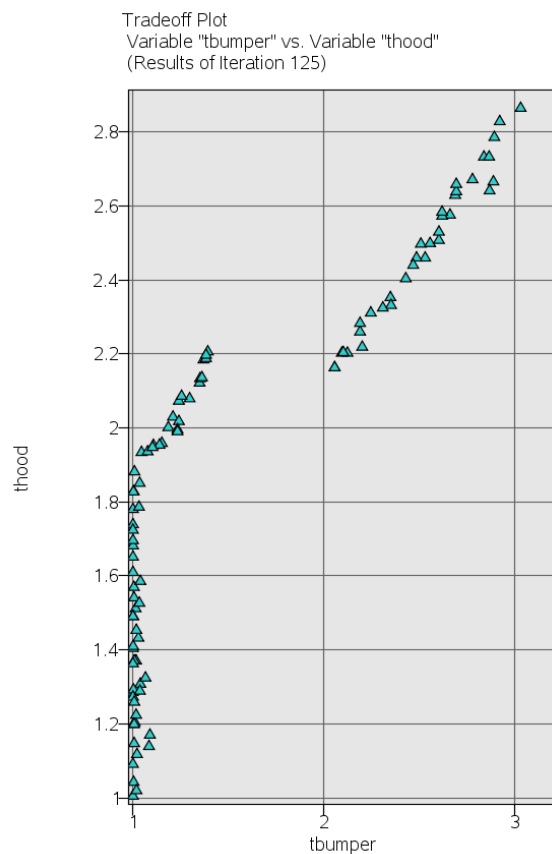
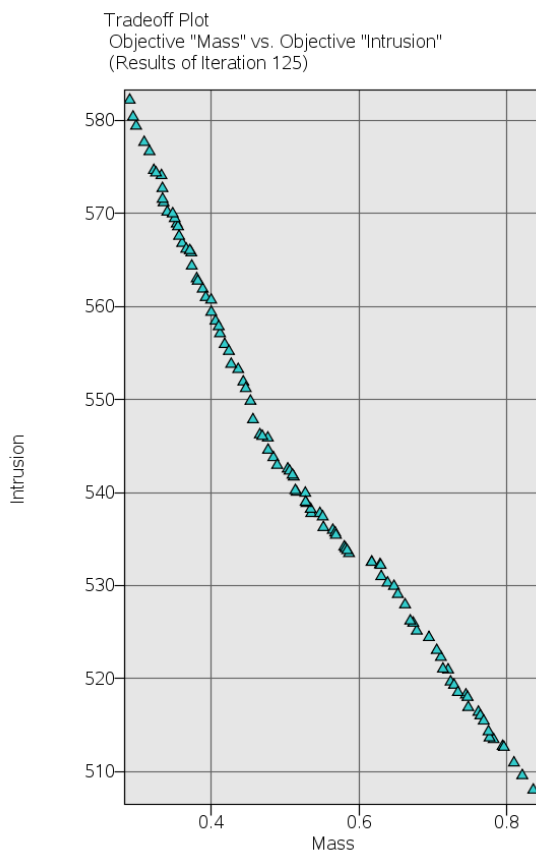
<i>Intrusion</i>	<i>Mass</i>
521	0.29

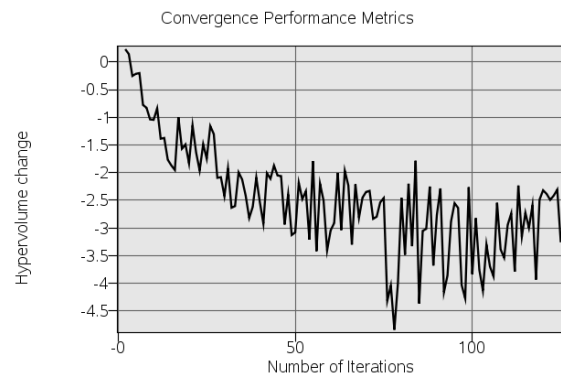
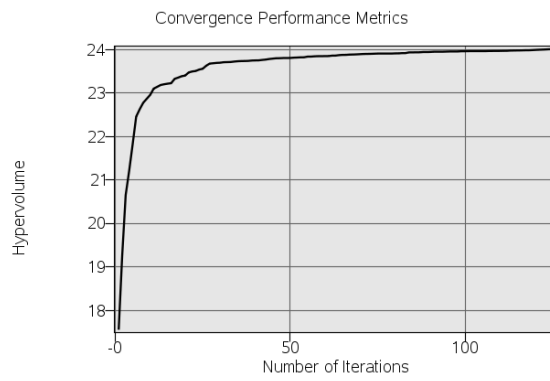
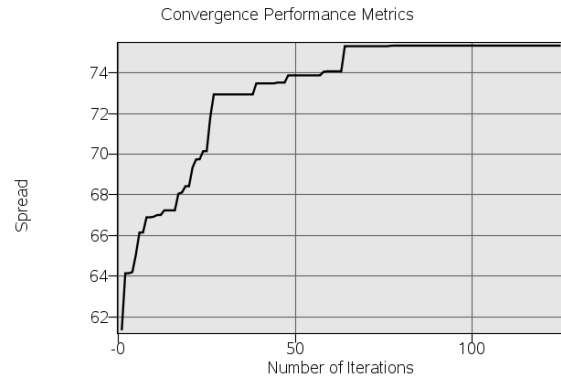
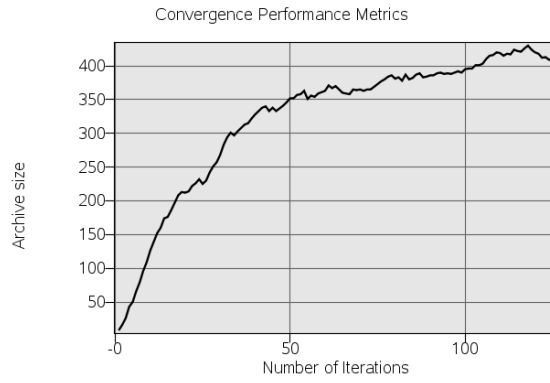
3. Add the Parallel Coordinate (Pareto) plot by splitting the screen vertically (and plot iteration 20 as well). Check "Select from active points" on the Parallel Coordinate plot and slide the upper bounds of the *Mass* and *Intrusion* functions down to about 0.65 and 560 resp. leaving a limited number of Pareto candidate points. What do you observe on the Tradeoff plot?
4. Add the SOM plot by splitting the screen again (plot iteration 20) 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 (iteration 20). 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}	$w_{Intrusion}$	<i>tbumper</i>	<i>thood</i>	<i>Mass</i>	<i>Intrusion</i>
0	1	2.5	2.46	.73	521.7
1	1	1	1.8	0.47	546.5
1	0	1.0	1.0	.29	583

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

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





Pole crash problem using Metamodel-Based optimization

Directory:

DESIGN_OPTIMIZATION/MULTIOBJECTIVE/SIMPLE/METAMODEL

Starting file: **metamodel.moo.start.lsopt**

Setting up the problem

1. *Optimization.* Select Mass and Intrusion as objectives. De-select all other. Set a HIC upper bound constraint of 250.
2. *Task Selection.* Select the task as *Metamodel-based Optimization*. Select *Sequential Optimization* strategy with Create Pareto Optimal Frontier option.
3. *Sampling.* Confirm that RBF and Space Filling are selected.
4. *Optimization→Algorithms.* Set the termination criterion to *Hypervolume change*.
5. *Termination criteria.* Set the iteration limit to 10. Set the response accuracy tolerance to 0.0001.
6. *Run.*

(Don't forget to set your LS-DYNA command of your stage before running the problem).

Exercise

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

<i>Intrusion</i>	<i>Mass</i>
514	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}	$w_{Intrusion}$	<i>tbumper</i>	<i>thood</i>	<i>Mass</i>	<i>Intrusion</i>
0	1	2.98	2.8	.82	514.5
1	1	1	1.97	.51	543.6
1	0	1.0	1.0	.29	580.17

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 Termination Criteria dialog) and compare the results again.

w_{Mass}	$w_{Intrusion}$	<i>tbumper</i>	<i>thood</i>	<i>Mass</i>	<i>Intrusion</i>
0	1	3.0	2.8	.83	509
1	1	1.1	1.9	.5	543
1	0	1.0	1.0	.29	582

9. Select 20 verification runs and extend the run (do not select a clean start). LS-OPT will simulate 20 Pareto designs from the last iteration.
 - a. Plot Intrusion vs. Mass using the Scatter Plot selection in the Viewer. Use "Max Constr. Violation" to color code the design points. This will provide an impression of the feasibility of the computed Pareto designs.
 - b. Tabulate the Utopian design of the simulation results.

<i>Intrusion</i>	<i>Mass</i>
508	0.29

MULTI-LEVEL OPTIMIZATION

This example demonstrates the approach for setting up a multi-level optimization in LS-OPT using nested optimization framework. An initial LS-OPT input file is treated as the outer loop of the setup with an LS-OPT stage consisting of another LS-OPT input file considered as the inner loop of the setup. The outer loop design variables are transferred as constants to the inner loop which could have its own design variables and an optimization setup. The optimum responses of the inner loop are extracted and defined as responses of the outer loop.

Two-level pole crash problem

Problem description

This example is a two level optimization of a simple pole crash analysis with both levels consisting of head injury criteria (HIC) and intrusion distance as the design objective and constraint, respectively. The inner loop optimization is based on thickness of a few selected parts whereas; the material parameters are considered as design variables of the outer loop optimization.

Directory:

DESIGN_OPTIMIZATION/MULTILEVEL/OuterMat_InnerThickness

Starting files: **Outer.start.lsopt**, **inner.start.lsopt**

Setting up the problem

Inner loop

1. Open the file **inner.start.lsopt** using the LS-OPT GUI.
2. Inspect the *Stage* component of the flowchart and make sure the LS-DYNA command and input file are defined.
3. Inspect the *Setup* component of the main GUI to assign the design variables of the inner loop optimization. Change the *Parameter* type of the thickness variables *tbumper* and *thood* to *Continuous* and assign a lower and upper bound of 1 and 5, respectively. The material variables SIGY and YM are the outer loop design variables which are transferred to the inner loop. Therefore, *Transfer variable* should be selected as the parameter type for the outer loop variables.

4. Assign a population size of 10 in the *Sampling* component.
5. Inspect the optimization component and confirm that HIC and intrusion distance are selected as objective and design constraint, respectively.
6. Inspect the termination criteria component and make sure you will be running four generations. Save the LS-OPT input file as **inner.correct.lsopt**.

Outer loop

1. Open the file **outer.start.lsopt** using the LS-OPT GUI.
2. *Stage Setup*: Select the **LS-OPT** solver option under stage setup. Enter the full path of the *lsopt* executable in the command section (default path can be selected by checking the option ‘*use default command*’), select **inner.correct.lsopt** as the input file. Add the solver input files of the lower level LS-OPT run using “Extra input files” option and click OK.
3. The variables defined in **inner.correct.lsopt** as transfer variables are now listed in the *Setup* component. Change the parameter type to *Continuous* and assign the following lower and upper bounds for the outer loop variables.

<i>Variable</i>	<i>Starting value</i>	<i>Lower bound</i>	<i>Upper bound</i>
SIGY	400	350	450
YM	200000	150000	250000

4. *Sampling*: Enter the 20 as the number of simulation points for the space filling sampling technique. Make sure *Radial Basis Function Network* is selected as the metamodel type.
5. *Stage Responses*: Define the outer loop responses using the LS-OPT option listed under response options. The LS-OPT response option lists all the responses, composites, objectives and constraints defined in **inner.correct.lsopt** file. Define the outer loop LS-OPT responses for head injury criteria and intrusion distance by selecting the corresponding component and make sure that the last iteration value is extracted. You can also add the inner loop LS-OPT variables *tbumper* and *thood* and inner loop responses for visualization.
6. *Optimization*: Select the LS-OPT response for *HIC* as the objective and LS-OPT response for *Intrusion* as the design constraints with a value of 550 as the upper bound.

7. Save the outer loop setup and use *Normal Run* option to run the two-level optimization example.

Exercise

1. Open the *Viewer* and inspect the simulation, metamodel and optimization history plots.
2. Report the optimum values of the objective, constraint and all the design variables. Note: You need to check the inner loop results of the verification run to obtain the optimum values of the inner loop variables. However, to avoid this step, you can define the inner loop variables as responses of the outer loop setup while setting up the example.

	<i>Start</i>	<i>Optimum</i>
<i>SIGY</i>	400	450
<i>YM</i>	200000	150000
<i>tbumper</i>	3	4.80
<i>thood</i>	1	1
HIC	67.28	47.25
Intrusion	287.65	307.21

3. Define the inner loop variables *tbumper* and *thood* as responses of the outer loop and extract the optimum values using the *Repair* tool.
4. Can metamodel-based optimization techniques be used in the inner loop instead of direct optimization? Comment on the simulation cost vs. metamodel accuracy associated with this approach.

RELIABILITY ANALYSIS

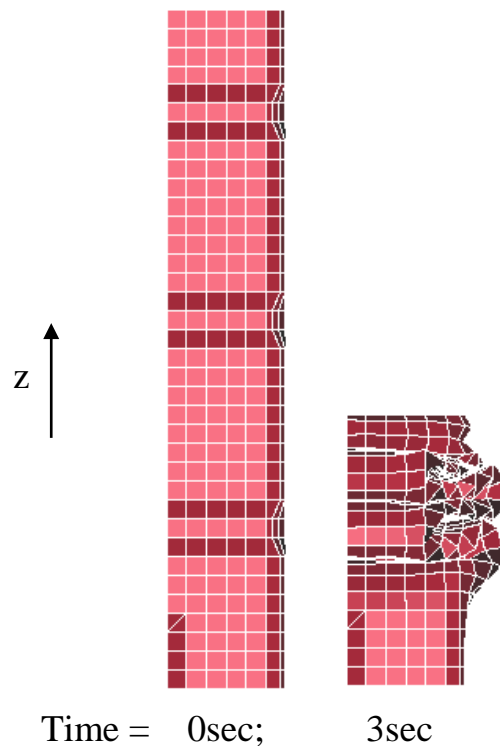
Reliability analysis is used to compute the probabilities of events.

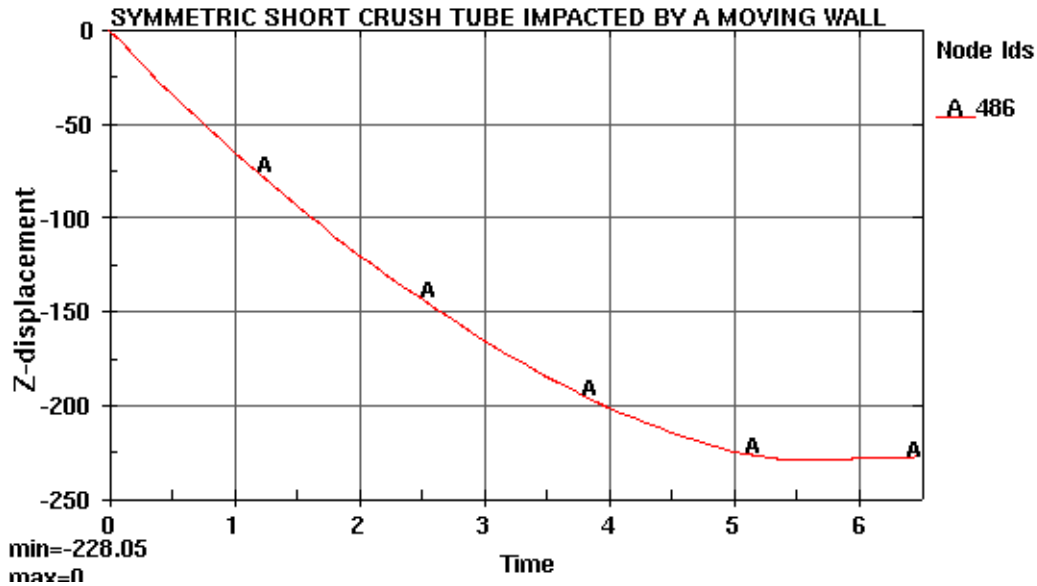
This example demonstrates:

- 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 `monte_carlo.start.lsopt` 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.

(Don't forget to set your LS-DYNA command of your stage before running the problem).

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 Statistics Summary and Probability of constraint violation options to:
 - a. Verify that the mean is -227 and standard deviation is 6.0
 - b. Verify that the probability of the response being larger than -220 is 0.1.
 - c. Verify that the probability of the response being less than -230 is 0.35.
 - d. Verify that the probability of the response being less than -235 is 0.05.
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?
5. Open Simulation Histories plot. Display the Histories, Mean and Standard deviation statistics of TOP_DISP_HIST.

DYNA Stats

(If not defined, open LS-Opt general settings and define the path through LS-PREPOST.) 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 `metamodel.start.lsopt` 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. Keep the default value of 2 standard deviations for the noise variable subregion size.
6. Verify that the metamodel is a quadratic surface.
7. Use full factorial design with 3 points per variable.
8. Constrain the displacement constraint to be more than -230 (a lower bound therefore).
9. Save your work under any name.
10. Run the Metamodel-based Monte Carlo Analysis.

(Don't forget to set your LS-DYNA command of your stage before running the problem).

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 Statistics Summary plot type.
 - c. Verify that the probability of the response being larger than -220 is 0.15 using the Probability of constraint violation 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

(If not defined, open LS-Opt general settings and define the path to your local LS-PREPOST executable.)

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 metamodel 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.

CLASSIFIER-BASED RELIABILITY ANALYSIS WITH SHAPE PARAMETRIZATION

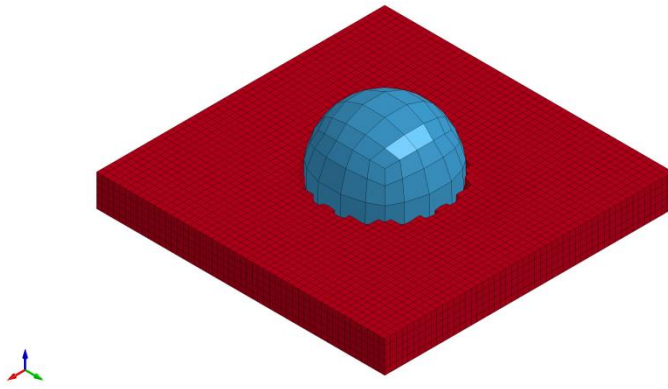
Reliability analysis is used to compute the probabilities of events.

This example demonstrates:

- Shape parametrization using LS-PrePost, and
- Variable bound selection for metamodel-based Monte Carlo Analysis.
- Binary response

Problem description

This example is a Monte Carlo analysis of a ball impacting a plate. The effect of variations in the ball radius and the plate thickness, yield strength and tangent modulus is investigated. The plate experiences failure upon exceeding an effective plastic strain of 0.2. The geometry is shown for a deformed failed state. The ball falls from a height of $(r+50)$ mm, r being its radius.



The distributions for the random variables are given below.

Variable	Distribution Type	Distribution Parameters
Ball radius (r)	Uniform	LB: 45 mm UB: 55 mm
Plate thickness (t_{plate})	Normal	Mean: 0.5 mm Standard dev: 0.05 mm

Setting up the problem

1. Open the file `cls.mc.plate.start.lsopt` in the directory `PROBABILISTIC/RELIABILITY/SHAPE_PARAMETRIZATION/REL_PLATE_BALL/CLASSIFIER`
2. Study the parametrization of the problem. Note that some of the parameters are defined in the LS-PrePost stage while the rest are defined in the LS-DYNA stage.
3. Change the task to Metamodel-based Monte Carlo Analysis.
4. Change `r` and `tplate` to be noise variables with the above distributions. The distributions are already defined as `dist_r` and `dist_tplate`.
5. Change the value of standard deviations for the noise variable subregion size to 4.
6. Define the maximum upper surface effective plastic strain of the plate (part 20001) as a D3PLOT response `eps_max`.
7. Study the input file for the “Post” stage. This stage determines whether there has been any failure due to the limit on the effective plastic strain and assigns a binary failure status response (failed = -1, safe = +1).
8. Add an SVC classifier `cls_fail` to classify -1 and +1 failure status.
9. Use 20 space filling designs.
10. Constrain the classifier `cls_fail` to be greater than 0.
11. Save your work under any name.
12. Run the Metamodel-based Monte Carlo Analysis task.

Viewer

1. How many infeasible points are there? Select an infeasible point and visualize the deformed shape in LS-PrePost.
2. Plot the relative frequency histogram of `eps_max` (probability histogram). How many of the 10 (default) bins represent failure (> 0.2)?
3. Find the probability of exceeding the limiting effective plastic strain of 0.2 using the response `eps_max`.
4. Find the probability of failure based on the classifier `cls_fail` (see Summary Report). Is this the same as the previous value? Why or why not?
5. What could be the motivation behind changing the value of standard deviations for the noise variable subregion size to 4?

RELIABILITY BASED DESIGN OPTIMIZATION (RBDO)

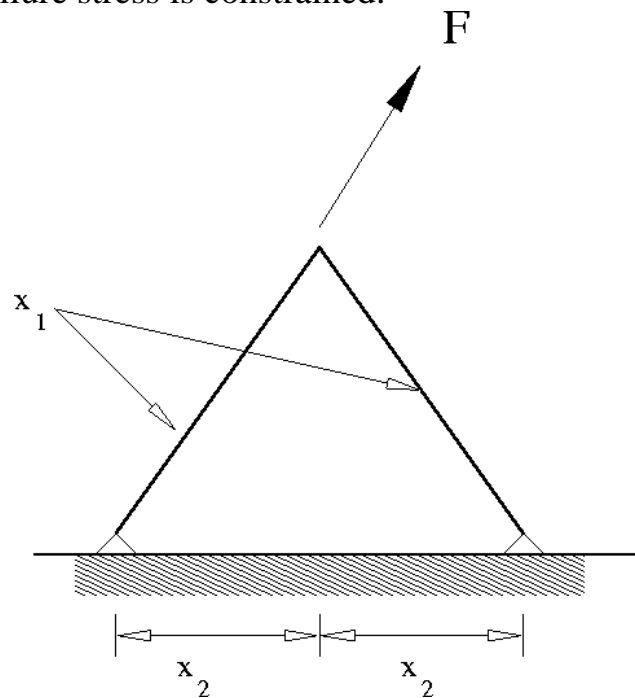
Reliability Based Design Optimization (RBDO) includes the variation of the design variables into the design problem.

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 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.



The two-bar truss problem.

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 `2bar.rbdo.start.lsopt` in the working 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. Make sure that the metamodel is a quadratic surface and that the experimental design is suitable (D-Optimal).
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 0.4.
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

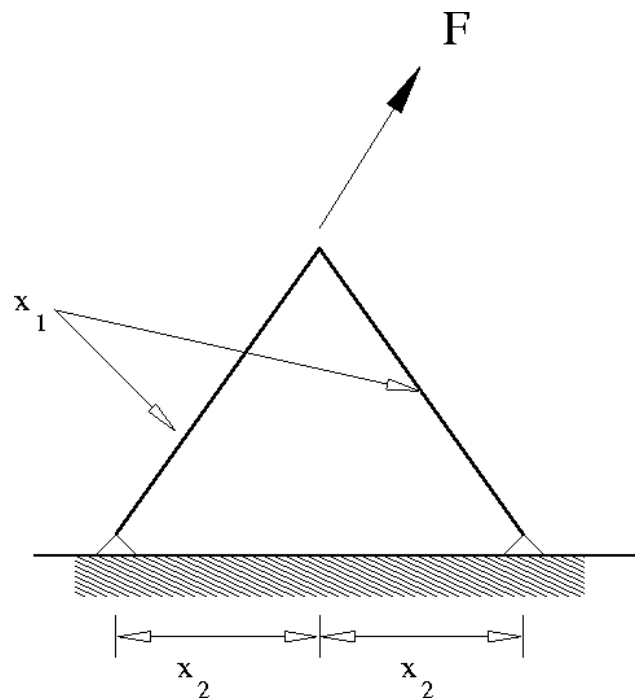
Robust parameter design selects designs insensitive to the variation of given parameters.

This example demonstrates:

- Robust Parameter Design

Problem description

This is the two-bar as considered previously: the 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.



The two-bar truss problem

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 `2bar.robust.start.lsopt` 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 composite 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. Use the `Metamodel` facility to investigate how the value of the stress standard deviation changes with the variables.

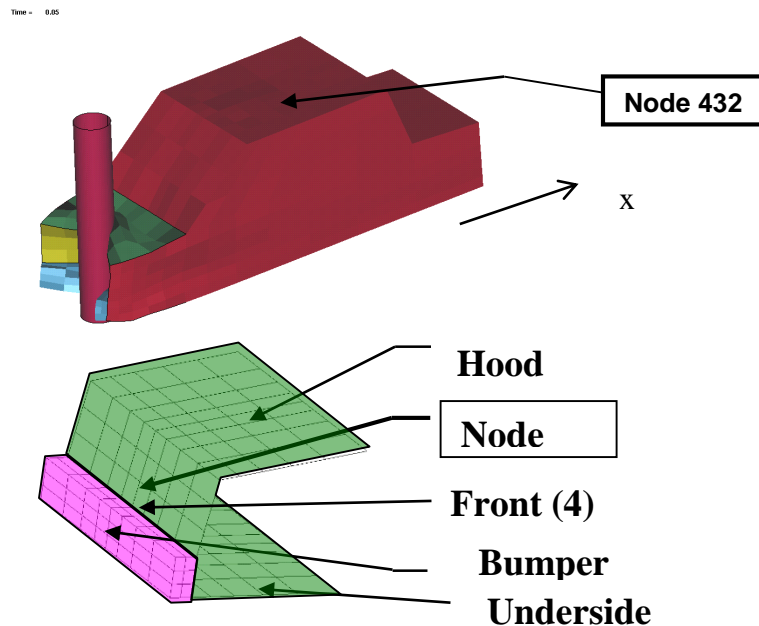
TOLERANCE OPTIMIZATION WITH IMPORTED METAMODEL

The problem illustrates the following features:

- User-defined experiment
- Metamodel import
- Monte Carlo analysis using imported metamodels
- Parametrization of probabilistic distribution parameters
- Extraction of probabilistic analysis results as responses
- Multilevel optimization
- Multi-objective optimization
- Tolerance optimization

Problem description

The problem consists 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.



The goal is to optimize two thickness parameters for the parts hood and bumper, as well as their associated tolerance values to attain a balance between the design objectives and the robustness of the optimum.

The underlying deterministic optimization problem without considering the tolerances or the effect of uncertainties is:

$$\begin{aligned}
& \min_{\mathbf{x}} \quad f_1 = \text{Mass}(\mathbf{x}) \\
& \text{s.t.} \\
& g_1(\mathbf{x}) = \text{Intrusion}(\mathbf{x}) - 550 \text{ mm} \leq 0 \text{ mm} \\
& g_2(\mathbf{x}) = \text{HIC}(\mathbf{x}) - 250 \leq 0 \\
& 1 \leq \mathbf{x} \leq 5
\end{aligned}$$

where \mathbf{x} is a vector of design variables, which are the thicknesses `thood` and `tbumper` of the selected parts. The solution of the above optimization may not be robust or reliable. To address that issue, tolerance limits are introduced. The thickness values must lie within the specified tolerance intervals. The nominal design variables are controlled so that the associated tolerance can be increased, thus making the design more robust with negligible probability/possibility of failure within the tolerance intervals.

This enhanced robustness may often come at the cost of other design objectives. Thus, the optimization formulation may consist of multiple competing objectives. In this example, the optimization is performed using two objectives. The nominal values of the mass are minimized while the relative tolerance is maximized. The final solution is a Pareto optimal front with a trade-off between the nominal mass and the relative tolerance.

The example is split into two parts:

1. Quantification of the uncertainty of a particular design within the tolerance limit. The design responses are calculated using pre-existing metamodels.
2. Optimization of the design and the associated tolerance.

Accurate global metamodels, previously constructed using LS-OPT, are already available for the required responses and there is no need for additional finite element simulations. The metamodel database is available as an XML file named *DesignFunctionsGlobal_PoleCrash*.

Imported metamodel-based Monte Carlo analysis with a fixed tolerance

In this example the goal is to determine the failure probability based acceptability (unacceptable if non-zero failure probability) of three specified design configurations, with nominal values of the hood and bumper thickness as follows:

- (1) `nominal_th = 1.9, nominal_tb = 3.0`
- (2) `nominal_th = 2.2, nominal_tb = 3.2`
- (3) `nominal_th = 1.9, nominal_tb = 3.0`

The design has uncertainties associated with the thickness of hood (`thood`) and the bumper (`tbumper`). Both the thickness values have Normal distributions with an equal standard deviation of 0.05.

A 2% tolerance limit (`rel_tol = 0.02`) is enforced for both the variables for the first two pairs of nominal thickness values. A 1% tolerance limit is used for (3).

The objective of this example is to learn:

- How to import a pre-constructed metamodel.
- How to use existing metamodels for Monte Carlo analysis without running simulations.
- How to parameterize an LS-OPT setup file using Transfer Variables.

Pre-setup Questions

1. How will you approach the setup if the failure probability was required at a single nominal design? Is there a difference in difficulty level?
2. Are the distributions in the three cases identical? What will be the distribution type?
3. Can you find all three probabilities without editing the distribution definitions? (Hint 1: what is the functionality of transfer variables?).

Hint 2:

mean values: `nominal_th` and `nominal_tb`,

lower bounds: `th_l = nominal_th*(1-rel_tol)`, `tb_l = nominal_tb*(1-rel_tol)`,

```
upper bounds: th_u = nominal_th*(1+rel_tol), tb_u =  
nominal_tb*(1+rel_tol).
```

Directory:

PROBABILISTIC/TOLERANCE/REL_IMPORT_METAMODEL

Setting up the problem

First, we will create a setup such that we do not need to edit the distribution for each of the three cases. We will parameterize the distribution so that we can simply change the nominal design and tolerance values to update the distribution and run LS-OPT again for the new configurations.

1. Start a new LS-OPT project with metamodel-based Monte Carlo analysis as the task type.
2. Specify the file *DesignFunctionsGlobal_PoleCrash* containing the pre-constructed metamodels using the *Import Metamodel* feature available in the *Build Metamodels* dialog.
3. Inspect the *Global Setup*. What are the variables in this example? Are `nominal_th` and `nominal_tb` variables or constants?
4. Set `tbumper` and `thood` as noise variables and assign appropriate distributions. (Note that the “&” operator can be used to parametrize a distribution parameter, constant and dependent variables can be used to define the distribution parameters.)
5. Inspect the response dialog.
6. Define the HIC and Intrusion (`Disp1-Disp2`) constraints.
7. Select Import Metamodels in the task dialog.
8. Run the metamodel-based MC analysis using 10000 MC samples.

Exercise:

1. Are the variable distributions same for any two of the three design configurations? If yes, then why? If no, which distribution parameters are varying?
2. Can the probabilities of failure at the three given sets of nominal designs be determined using a single LS-OPT run? Is any change required in the setup? If so, clear the earlier results and re-run LS-OPT.

- d. Failure probability likely to decrease or remain the same
 - e. Failure probability likely to increase or remain the same
7. And what will happen to the failure probabilities of design (3)?

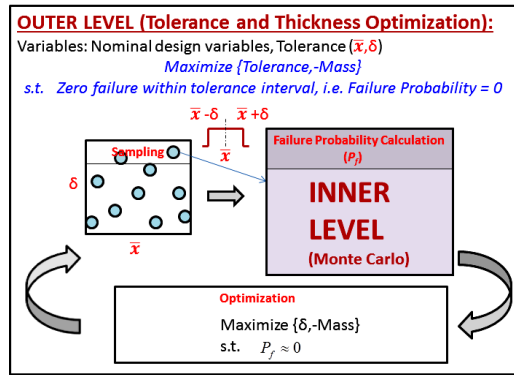
Multi-objective tolerance and mass optimization of vehicle

The deterministic optimization formulation presented earlier minimizes the mass of the car while satisfying the specified upper bounds on the intrusion and HIC, but the solution may not be robust or reliable. In this example tolerances are introduced into the problem and the nominal design variables are controlled so that the associated tolerance can be increased, thereby increasing the reliability.

For simplicity, both the thicknesses are assumed to have the same relative tolerance rel_tol . It should be noted the optimization variables are the *nominal values* for the thickness, referred to as $nominal_th$ and $nominal_tb$. Overall, the problem consists of three optimization variables - $nominal_th$, $nominal_tb$ and rel_tol . The tolerance-based optimization problem is:

$$\begin{aligned}
 & \max_{\bar{\mathbf{x}}, \delta} \quad \{rel_tol, -nominal_mass\} \\
 & s.t. \\
 & \quad Probability[Intrusion > 500] \leq 10^{-8} \quad \forall \mathbf{x} \in [\bar{\mathbf{x}}(1-rel_tol), \bar{\mathbf{x}}(1+rel_tol)] \\
 & \quad Probability[HIC > 250] \leq 10^{-8} \quad \forall \mathbf{x} \in [\bar{\mathbf{x}}(1-rel_tol), \bar{\mathbf{x}}(1+rel_tol)]
 \end{aligned}$$

where $\bar{\mathbf{x}}$ is the nominal design. The constraints on intrusion and HIC must be satisfied at all possible designs within the tolerance interval (i.e. $\forall \mathbf{x} \in [\bar{\mathbf{x}}(1-rel_tol), \bar{\mathbf{x}}(1+rel_tol)]$). The tolerance range is 0.01% to 10%.



To reduce the cost of the two-level optimization, the inner level Monte Carlo analysis is performed using imported pre-constructed global metamodels.

Directory:

PROBABILISTIC/TOLERANCE/MULTIOBJECTIVE/REL_DESIGN

Setting up the problem

1. Open the file `outer_tolerance.moo.start.lsopt`
2. Change the outer level task to direct optimization and specify 20 generations with population size 20.
3. Inspect the outer level global setup and ensure that `nominal_th`, `nominal_tb` and `rel_tol` are the optimization variables.
4. Open the inner level by clicking *Open* next to the outer level stage input file.
5. Inspect the outer and inner level setups. Ensure that the inner level can perform a metamodel-based MC analysis for any outer level design configuration with `thood` and `tbumber` as the noise variables and the distributions parametrized with respect to the outer level optimization variables.
6. Define the HIC and Intrusion upper bound exceeding probabilities, and the nominal values of mass, HIC and Intrusion as responses in the outer level (LS-OPT Statistics response type).
7. Define the objectives: Maximize `rel_tol`, Minimize `nominal_mass`

- Define the constraints: Probabilities of failure ≈ 0 (upper bound lower than the reliability resolution of inner level)

The left screenshot shows the 'Objectives' tab in the software interface. It has two sub-tabs: 'Objectives' and 'Constraints'. Under 'Objectives', there are two checkboxes: 'Maximize the Objective Function (instead of minimize)' which is unchecked, and 'Create Pareto Optimal Front (Multi-Objective Mode)' which is checked. Below these, there is a section 'Objective components:' with a table. The table has two columns: 'Response/Composite' and 'Value'. There are two rows: one for 'obj_tol' and one for 'nominal_mass', both with a value of 'x'.

The right screenshot shows the 'Constraints' tab. It has two sub-tabs: 'Objectives' and 'Constraints'. Under 'Constraints', there are two checkboxes: 'Constraint scaling (internal)' which is unchecked, and 'Show advanced options' which is checked. Below these, there is a section 'Optimization constraints:' with a table. The table has five columns: 'Response', 'Lower Bound', 'Strict', 'Upper Bound', and 'Strict'. There are two rows: one for 'Pf_u_intru' and one for 'Pf_u_hic'. For 'Pf_u_intru', the 'Lower Bound' is 'Set lower bound', the 'Upper Bound' is '0.00000001', and the 'Strict' checkbox is checked. For 'Pf_u_hic', the 'Lower Bound' is 'Set lower bound', the 'Upper Bound' is '0.00000001', and the 'Strict' checkbox is checked.

- Perform a Normal Run.

Exercise:

- Verify that the minimum mass design has mass 0.573 and tolerance 2.03%.
- Verify that the maximum tolerance design has mass 0.608 and 10% tolerance.
- Comment on the Pareto front, knowing that the optimal mass value was 0.455 based on the deterministic formulation (without considering tolerances).
- Comment on the shape of the Pareto front. Is there any particular region on the front that signifies a transition between two categories of design or behavior?
- Is the design robust from a mass perspective?

BIFURCATION/OUTLIER ANALYSIS

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

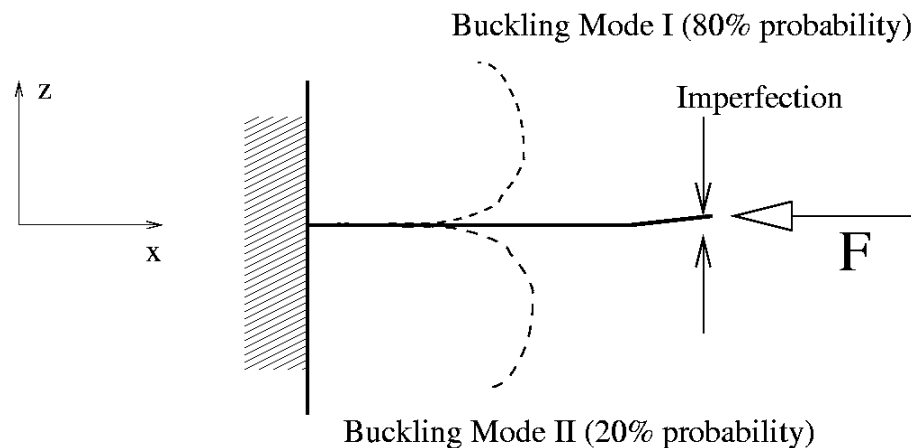
This example demonstrates:

- 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.



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 `outlier.lsopt` in the directory `PROBABILISTIC/OUTLIER_ANALYSIS`.

2. Run the Monte Carlo job.
(Don't forget to set your LS-DYNA command before running the problem).

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.
3. Using the history plot 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.

DYNA Stats

Use the `lsioptui 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.

ROBUSTNESS OF METAL FORMING (OPTIONAL)

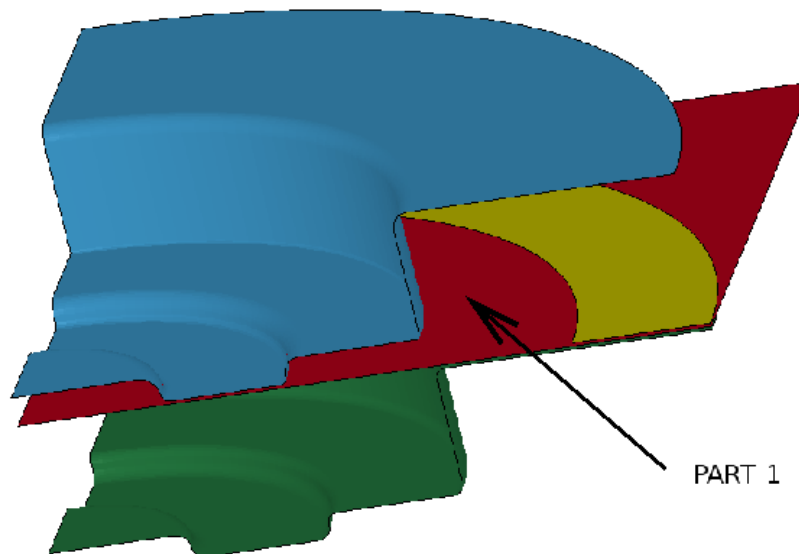
Metal forming requires the analysis of adaptive results at specific coordinates. The results can then be compared even though the node locations and numbers differ between FE models.

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_RELIABILITY`. If not, run the file `metal_MC.Isopt`.

(Don't forget to set your LS-DYNA command before running the problem).

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_fldc-eps1*”.