Morning Session (8:30 AM - 12:30 PM)

1. Introduction to HyperStudy
2. Exercises 1.1 and 1.2
3. BREAK (10 Minutes)
4. Design of Experiments (DOE)
5. Exercises 2.1, 2.2 and 2.3
6. Approximations
7. Exercises 3.1 and 3.2

Afternoon Session (1:30 - 4:30 PM)

8. Optimization
9. Exercises 4.1, 4.2 and (one of these 4.3, 4.4 or 4.5)
10. BREAK (10 Minutes)
11. Stochastic
12. Exercises 5.1 and 5.2
**Altair’s Vision**

**Today…**

- **Design (CAD)**
- **Virtual Test (CAE)**
- **Build**
- **Test**
- **Redesign**

**Altair…**

- **Concept Design Optimization**
- **Design (CAD)**
- **Virtual Test (CAE)**
- **Build**
- **Test**
- **Optimization**

**Optimization is Driver in CAE Driven Design Process**
Optimization with Altair HyperWorks

- Topology/Topography optimization: OptiStruct
- Shape/Size Optimization: OptiStruct/HyperStudy

Design Concept
Preliminary Design
Product Design
Prototype
Validation Testing

Parameter Studies: HyperStudy, HyperMorph
Robust/Reliable Design: HyperStudy
Chapter 1: Introduction to HyperStudy
Altair HyperStudy

- is a software to perform Design of Experiments (DOE), approximations, optimization, and stochastic studies.
- is applicable to study the different aspects of a design under various conditions, including non-linear behaviors.
- can be applied in the multi-disciplinary optimization of a design combining different analysis types.
- is integrated with HyperWorks through direct links to the models in HyperMesh (.hm), HyperForm (.hf), and MotionView (.mdl).
- models can be easily parameterized. Aside from the typical definition of solver input data as design variables, the shape of a finite element model can also be parameterized with ease through HyperMorph.
## Study Definition

<table>
<thead>
<tr>
<th>Study Type</th>
<th>DOE</th>
<th>Approximation</th>
<th>Optimization</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters Screening</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>System Performance Study</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>Response Surface Evaluation</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>Optimum Design</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>Variation Analysis</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>Robust Design</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
<tr>
<td>Reliability Design</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
<td>![thumbs-up]</td>
</tr>
</tbody>
</table>
DOE Studies

Design Space Sampling

Run DOE

Evaluate results

- Invoke solvers
- Batch submission
- Parallel job submission

- Factorial designs
- Plackett-Burman
- Box-Behnken
- Central-Composite
- Latin HyperCube
- Hammersley
- User defined
- External Matrix

- Effects plots
- ANOVA
Approximations

- Least square
- Moving least square
- HyperKriging
Optimization Studies

Multi-disciplinary Optimization

Define Design Variables

Run Optimization

Evaluate results

- Invoke solver directly
- From approximation
- Genetic Algorithm
- Sequential Quadratic Programming
- Adaptive Response Surface
- Method of Feasible Directions
- External optimizers

NVH Analysis

Size variables

Crash Analysis

Shape variables

- Objective function
- Design constraints
- Design variables
- Responses
Define input distributions

Select sampling method

Run stochastics

Evaluate results

- Normal
- Uniform
- Triangular
- Exponential
- Weibull

- Simple Random
- Latin HyperCube
- Hammersley

- Call solver directly
- Using response surface

- Probability Distribution Function (PDF)
- Cumulative Distribution Function (CDF)
- Histogram Distribution
- Ant-Hill plots
- Statistical moments
HyperStudy Process Flow

- Study Setup
  - Create studies
  - Create models
  - Create design variables
  - Do nominal run

- DOE Setup
  - Create DOE study
  - Controlled variables
  - Controlled interactions
  - Controlled allocations
  - Uncontrolled variables
  - Uncontrolled interactions
  - Uncontrolled allocations
  - Select responses
  - Write/execute runs
  - Extract responses

- Approximation
  - Create approximation
  - Input matrix
  - Validation matrix
  - Build approximation
  - Residuals

- Optimization Setup
  - Create optimization study
  - Define design variables
  - Constraints
  - Objectives

- Stochastic Setup
  - Create stochastic study
  - Define random variables
  - Define correlation
  - Select responses
  - Write/execute runs
  - Post processing
Preference file
Job Management
## Register Solver Script

<table>
<thead>
<tr>
<th>Label</th>
<th>Varname</th>
<th>Script Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADIOSS</td>
<td>radioss</td>
<td>C:/Altairwin64/hw10.0build60/hw solvers/bin/win64/radioss.bat</td>
</tr>
<tr>
<td>OptiStruct</td>
<td>os</td>
<td>C:/Altairwin64/hw10.0build60/hw solvers/bin/win64/optistruct.bat</td>
</tr>
<tr>
<td>Templex</td>
<td>templex</td>
<td>C:/Altairwin64/hw10.0build60/hwbin/win64/templex.exe</td>
</tr>
<tr>
<td>HyperXtrude</td>
<td>fx</td>
<td>C:/Altairwin64/hw10.0build60/hwbin/win64/fx.exe</td>
</tr>
<tr>
<td>MotionSolve - standalone</td>
<td>ms</td>
<td>C:/Altairwin64/hw10.0build60/hw solvers/bin/win64/motionsolve.bat</td>
</tr>
<tr>
<td>TCL</td>
<td>tcl</td>
<td>C:/Altairwin64/hw10.0build60/hwbin/tcl/4.13/win64/bin/tclsh84.exe</td>
</tr>
</tbody>
</table>

**Display the Readers Solver Script dialog box.**
The process of selecting design variables / factors / random variables for Optimization / DOE / Stochastic studies can be summarized as:

Parametrization of a FEA model

Models in HyperMesh, MotionView, or HyperForm → Parameterized Model using Create Template GUI → Input file for HyperStudy → DOE → Approximation → Optimization → Stochastics → Direct Response Selection for Automated Post-Processing
Direct linking to Hypermesh-MotionView-HyperForm provides HyperStudy direct access to simulation models and to the features such as thickness, concentrated masses, shape changes which are used as the design variables in DOE, optimization or stochastic studies.

When HyperStudy is launched in a standalone mode, the design variables need to be identified using a common method as HyperStudy interacts with solvers of different physics.

HyperStudy uses TEMPLEX to parameterize the solver input deck. The solver input deck is taken and “parameter” statements are added, which in turn is linked to the property such as thickness, concentrated masses. This file with “parameter” statements is called as “parameterized input deck”. This deck forms the input file for HyperStudy.
Parameterized Model using Create Template GUI

Models in HyperMesh, MotionView, or HyperForm

Input file for HyperStudy (.hm, .mdl, .hf)

HyperStudy/HM-MV-HF link
HyperMesh-HyperStudy Link

Shapes in HyperMesh

Model Parameterization
Exercise 1.1: Create Design Variables Through HyperMesh
Parameterized Model using Create Template GUI

Parameterized File for HyperStudy (.tpl)
A parameterized solver input deck is required when HyperStudy is launched in standalone mode.
**What is Templex?**

*Templex* is a general-purpose text and numeric processor. Templex generates output text based on guidelines defined in a template file.

The input files can be created using any text editor or word processor that can save a file in ASCII format.

*Templex* solver can be used to process the template file and the output can be sent to a monitor, to another file, HyperGraph or to other output streams.

*Templex* is built into HyperStudy as an expression builder for creating responses for Objectives and Constraints.

*Templex* can also be used as a standalone solver on mathematical expressions to perform Design of Experiments (DOE) / Optimization / Stochastic studies.
Important Templex Rules

• Statements, expressions, and variables must be placed between braces, {}, so that Templex will evaluate them. Otherwise they are treated as plain text and not processed.

• After Templex processes a parameterized template file, the plain text is copied to the specified output stream verbatim.

• When Templex encounters an expression within braces, it evaluates the expression and sends the resulting value to the output stream. If the item between the braces is a Templex statement, Templex carries out the statement instructions. If the Templex statement generates output, the resulting value is sent to the output stream too.

• Variable names can be up to eighty characters long. They consist only of letters, numbers, and underscores; no other characters are allowed. Variable names must always begin with a letter and are case sensitive.

• In addition to scalars Templex supports one- and two- dimensional arrays. In HyperStudy to define the responses the results from the analysis runs are represented as one-dimensional arrays “VECTORS”. The index starts with 0 as in C language.

• The template file must have at least one parameter statement to be able to read into the HyperStudy.
The parameterized input deck is a **Templex template, *.tpl, file.**

To create the *.tpl file, the following are added to the analysis input file:

1. Templex parameter statements,
2. Field variable calls, and
3. External-file-include statements
1. Templex parameter statements are used for individual field changes and shape vector (DESVAR) nodal perturbations. All parameter statements are placed at the top of the *.tpl file.

The templex parameter function has the following format:

```
{parameter(varname, label, nominal, minimum, maximum)}
```

Example:

```
{parameter(TH1, "Thicknes 1", 1.6, 0.5, 3.5)}
```
2. Field variable statements must be inserted into the analysis deck for each field to be optimized, such as component thickness, density, etc.

The formats for field variable statements is as follows:

{variable,%field_width.decimal_places}

‘Variable’ is the same as in the corresponding parameter statement.

The ‘field width’ value depends on the solver being used so that the proper spacing is maintained in the analysis deck. Each solver will have its own field width: LS-DYNA is 10, OptiStruct is 8, and HyperForm is 8. The ‘decimal_places’ value is a user preference.

Example

{TH1, %8.5f}
Original LS-DYNA Deck

$HMNAME   PROPS    1beam
1         71.0              20.0       0.0             0
1.6       1.6       1.6     1.6       0.0

Parameterized LS-DYNA Deck (.tpl)

$HMNAME   PROPS    1beam
1         71.0              20.0       0.0             0
{TH1,%10.5f}{TH1,%10.5f}{TH1,%10.5f}{TH1,%10.5f}0.0

3. If the design variables in optimization are shape vectors, further editing of the
dock is necessary such as external-file-include statements.
HyperStudy offers a simple GUI to parameterize input decks.

Example: Nastran Input Deck

HyperStudy offers a simple GUI to parameterize input decks.
Example: Nastran Input Deck
Exercise 1.2: Create Design Variables using the Create Template GUI
Direct Solvers & Results Access

- Use of HyperWorks result readers: no configuration needed
- Supported Solvers
  - ABAQUS
  - ADAMS
  - ANSYS
  - DADS
  - LS-DYNA
  - MADYMO
  - MARC
  - NASTRAN
  - PAM-CRASH
  - RADIOSS
  - SIMPACK
- HyperWorks Solvers
  - OptiStruct
  - MotionSolve
  - HyperForm
- General Result Formats
  - MS Excel (.csv file)
  - Any XY data
- File Parser
- Additional interfaces
  - Import Reader Language
  - External Readers Programmed in C
Interfacing with Solvers

By default, HyperStudy has 4 solvers: OptiStruct, Templex, HyperForm and HyperXtrude. Additional, solvers may be added through the

- Register Solver Script GUI or
- Preference file located in <install_dir>/hw/prefinc directory.
Interfacing with Solvers through the GUI
Interfacing with Solvers through the preference file

```
*BeginStudyDefaults()

*BeginSolverDefaults()

*RegisterSolverScript(os, "OptiStruct", "C:/Altair/hw8.0/optistruct/bin/WIN32/optistruct.bat")
*RegisterSolverScript(templex, "Templex", "C:/Altair/hw8.0/hw/bin/WIN32/templex.exe")
*RegisterSolverScript(hf, "HyperForm", "C:/Altair/hw5.0/hm/bin/WIN32/hfisol.bat")
*RegisterSolverScript(hs, "HyperStudy", "C:/User/HyperStudy/HyperStudy/SolverScript/HSolver.bat")
*RegisterSolverScript(ansys, "Ansys", "C:/User/HyperStudy/AnsysDir/ansys.bat")
*RegisterSolverScript(fluent, "Fluent", "C:/User/HyperStudy/FluentDir/Fluent.bat")

*EndSolverDefaults()
```
Integration with Excel

![Excel Integration Diagram](image-url)
HyperStudy/HyperMorph Coupling

• Morphing: automatic mesh parameterization
• Morphing shapes used as design variables in HyperStudy and OptiStruct.
• Mesh based, no CAD data needed
Shape design variables necessary for DOE/Optimization/Stochastic studies can be easily handled using HyperMorph module within HyperMesh.

The finite element model is morphed to a given shape and then saved as “shape”. Multiple “shapes” can be saved in HyperMorph. These saved “shapes” are then used by HyperStudy as design variables.

Domains

| global domains | 1D domains |
| local domains | 2D domains |
| global and local | 3D domains |
| edge domains | auto functions |
| general domains |

Morph Volumes
Morph to Geometry
Handles

- Generated automatically
- Can be added by hand
- Global
- Local
- Biasing allows for C1 continuity
Shape Variable Generation using HyperMorph
Shape Variable Generation using HyperMorph
Example

Design Variable: Boss Radius
Exporting Shape Variables

Select Nodes
Exporting Shape Variables
Chapter 2: Design of Experiments (DOE)
Design of Experiments (DOE)

Input file for HyperStudy

Direct Response Selection for Automated Post-Processing
What is DOE?

Design of Experiments (DOE) can be defined as a series of tests in which purposeful changes are made to the input variables of a process or system so that the reasons for changes in the output responses can be identified and observed.

Objectives of DOE Study

• To determine which factors are most influential on the responses.

• To determine where to set the influential controlled input variables so that:
  The response is close to the desired nominal value.
  Variability in output response is small.
  The effects of the uncontrolled variables are minimized.

• To construct an approximate model that can be used as a surrogate model for the actual computationally intensive solver.
• A **Factor** is an input parameter (or design variable) of the system. Factors can be controlled or uncontrolled.

• A **Level** is a discrete (or continuous) value of the factor.

• A factor can be either **Discrete** i.e., slow (-) or fast (+) (Ex. a variety of seed, type of paint etc.) or **Continuous** (Ex. temperature, a sinusoidal input for frequency response analysis etc.)

• **Controlled factors** are design variables that can be realistically controlled in the production (real world) environment. Examples include gauge thickness of sheet steel, shape of a support bracket, and mold temperature.

• **Uncontrolled factors (Noise)** are variables that cannot be realistically controlled in the production (real world) environment, but can be controlled in the lab. Examples include ambient temperature and occupant seating positioning.
Main Effects are the influence of the factors on the responses.

Interaction Effect is the interdependence among the factors. Due to this interdependence, a difference in the response occurs when the factors are changed simultaneously as against what would have occurred when the factors were changed individually.
Key DOE concepts are illustrated by considering an airbag design. Here we want to study how the **Speed of Bag Inflation** $(S)$, **Size of Vent** $(V)$ and the **Size of Bag** $(B)$, influence the HIC of an air bag and also find out how these factors interact with each other.

<table>
<thead>
<tr>
<th>FACTOR</th>
<th>LABEL</th>
<th>LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Design Variable)</td>
<td></td>
<td>(-)</td>
</tr>
<tr>
<td>Speed of Bag Inflation</td>
<td>S</td>
<td>Slow</td>
</tr>
<tr>
<td>Size of Vent</td>
<td>V</td>
<td>Small</td>
</tr>
<tr>
<td>Size of Bag</td>
<td>B</td>
<td>Small</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>(+)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fast</td>
</tr>
<tr>
<td></td>
<td>Large</td>
</tr>
</tbody>
</table>
Initially, consider the simplified problem where the vent size (V) is constant and consequently the are only two factors, S and B. For this, a simple full factorial DOE matrix is given as:

\[
\begin{array}{cccc}
\text{ROW} & S & B & S \times B \\
1 & - & - & + \\
2 & + & - & - \\
3 & - & + & - \\
4 & + & + & + \\
\text{Divisor} & 2 & 2 & 2 \\
\end{array}
\]

The following DOE study will result in a polynomial expression that relates HIC to the two factors: Speed of the air bag and Size of the Bag.

**Example:** \( HIC = a0 + a1 \times S + a2 \times B + a3 \times S \times B \)
DOE Study Types - Applications

• DOE Types Available in HyperStudy
  • Full Factorial
  • Fractional Factorial
  • Central Composite
  • BoxBehnken
  • Plackett Burman
  • Latin Hypercube
  • Hammersley
  • Run Matrix
  • User Defined
• DOE for Screening
• DOE for Factorial Studies
• DOE for Response Surface (RSM) evaluation
**Objective**

- A simple DOE study (ex. two level design with no interactions) will provide a global understanding of the complete system i.e give the magnitude and direction of effects
- This initial screening exercise will allow parameters which do not influence the system to be discarded thus reduce the number of factors and runs.
- Lower precision

**Types**

- Fractional Factorial
- Plackett-Burman
- D-Optimal
DOE for Factorial Study

Objective

- Fewer factors
- Main effects and some factor interaction effects
- Linear model

Types

- Full Factorial
- Fractional factorial
- D-Optimal
Objective

- Fewer factors
- Model of relationships
- Accurate prediction
- Optimization

Types

- Box-Behnken
- Central composites
- D-Optimal
DOE Process

1. Assign factors (dv’s) from .tpl, .hm,.mdl, .hf file.
2. Perform nominal run to create responses for DOE study.
3. Select the **DOE type** for controlled and/or uncontrolled factors.
4. Divide the factors into controlled and uncontrolled if needed.
5. Export the solver input files for the specified runs as required by the DOE matrix.
6. Solve the above exported files.
7. Extract the responses for the above solved files.
8. Study the main effects, interaction effects, sensitivity index.
Exercise 2.1: DOE Study of a Rail Joint using OptiStruct
Exercise 2.2: DOE Study of a Cantilever Beam using Templex

### Design Variables
- **Width:** $20 < b < 40$
- **Height:** $30 < h < 90$
- **Length:** $50 < L < 100$

**Design Space:** *All beam elements*

### Responses:
\[
\sigma_{\text{max}}(L, b, h) = \frac{M_{c}}{I}
\]
\[
U_{\text{max}}(L, b, h) = \frac{pL^{3}}{3EI}
\]
\[
Vol(L, b, h) = Lbh
\]
Exercise 2.3: DOE with MotionSolve Data Mining
Chapter 3: Approximations
Approximation

Models in HyperMesh, MotionView, or HyperForm

Parameterized Model using Create Template GUI

Input file for HyperStudy

DOE → Approximation → Optimization → Stochastics

Direct Response Selection for Automated Post-Processing
Approximations are surrogate models that represent the actual responses.

Why the need for approximations?
- Some simulations are computationally expensive which makes it impractical to rely on them exclusively for design studies. Use of approximations in such cases lead to substantial savings of computational resources.
- Optimization can fall into local minimum or maximum when the responses are nonlinear. Using approximate responses, the user can avoid this issue.

The challenge:
- When using approximations, the issue of a trade off between accuracy and efficiency is ever present.
- The question is how approximate the representation of the design space can be while remaining accurate enough.
Approximations Definitions

• **Regression** is the polynomial expression that relates the response of interest to the factors that were varied. It is only as good as the levels used when performing the study. For example, a two-level parameter only has a linear relationship in the regression. Higher order polynomials can be introduced by using more levels. Note that using more levels results in more runs.

• **Linear Regression model**
  • \( F(X) = a_0 + a_1X_1 + a_2X_2 + \text{(error)} \)

• **Interaction Regression Model**
  • \( F(X) = a_0 + a_1X_1 + a_2X_2 + a_3X_1X_2 + \text{(error)} \)

• **Quadratic Regression Model (2nd order)**
  • \( F(X) = a_0 + a_1X_1 + a_2X_2 + a_3X_1X_2 + a_4X_1^2 + a_5X_2^2 + \text{(error)} \)
**Anova** (Analysis of Variance) is a representation of the contributing percent of each design variable for the selected response.
Approximation Types

• Least Squares Regression
• Moving Least Squares Method
• HyperKriging
Approximation Types

Least Squares Regression (LSR)

LSR attempts to minimize the sum of the squares of the differences (residuals) between responses generated by the approximation and the corresponding simulation results.

HyperStudy allows the creation of least squares regressions for any polynomial order.

Moving Least Squares Method (MLSM)

MLSM is a generalization of a conventional weighted least squares model building. The main difference is that the weights, associated with the individual DOE sampling points, do not remain constant but are functions of the normalized distance from a DOE sampling point to a point \( x \) where the approximation model is evaluated.

HyperStudy provides the choice of first, second, and third order functions for the Moving Least Square fit.
HyperKriging

An approximation function, built using Kriging, has the property that it agrees with the original response function at the DOE points, i.e. this approach produces an interpolation model. This makes the technique suitable for modeling highly nonlinear response data that does not contain numerical noise. HyperKriging is computationally more expensive than the LSR and MLSM method.
Approximations Process

1. Select approximation types for responses
2. Import the run matrix
3. Build approximation models for each response
4. Check the residuals
5. Do trade-off studies and view ANOVA results
Exercise 3.1: Approximation Study of a Rail Joint using OptiStruct
Exercise 3.2: Approximation Study of a Cantilever Beam using Templex
Chapter 4: Optimization
Optimization

Models in HyperMesh, MotionView, or HyperForm

Parameterized Model using Create Template GUI

Input file for HyperStudy

DOE
Approximation

Direct Response Selection for Automated Post-Processing

Optimization

Stochastics
Optimization Problem Formulation

Objective: \[ \text{min } f(x) \]
\[ \text{min cost ($)} \]

Constraints: \[ g(x) \leq 0.0 \]
\[ \sigma < \sigma_{\text{allowable}} \]

Design Space: \[ \text{lower } x_i \leq x_i \leq \text{upper } x_i \]
\[ 2.5 \text{ mm} < \text{thickness} < 5.0 \text{ mm} \]
\[ \text{number of bolts } \in (20, 22, 24, 26, 28, 30) \]
**Design Variables**: System parameters that can be changed to improve the system performance.

- beam dimensions, material properties, diameter, number of bolts

**Objective Function**: System responses that are required to be minimized (maximized). These responses are functions of the design variables.

- mass, stress, displacement, frequency, pressure drop

**Constraint Functions**: System requirements that need to be satisfied for the design to be acceptable. These functions are also functions of the design variables.

- displacement, frequency, pressure drop, cost
**Optimization Definitions**

*Feasible Design*: Design that satisfies all the constraints.

*Infeasible Design*: Design that violates one or more constraints.

*Optimum Design*: Set of design variables along with the minimized (maximized) objective function that satisfy all the constraints.
Optimization methods can be classified in three categories:

1. Gradient Based: are effective when the sensitivities (derivatives) of the system responses w.r.t design variables can be computed easily and inexpensively.

2. Response Surface Based: are very general in that they can be used with any analysis code including non-linear analysis codes. Global optimization methods use higher order polynomials to approximate the original structural optimization problem over a wide range of design variables.

3. Exploratory Methods: are suitable for discrete problems such as finding the optimum number of cars to manufacture. These methods do not show the typical convergence of other optimization algorithms. Users typically select a maximum number of simulations to be evaluated. These algorithms are good in search on nonlinear domains however they are computationally expensive.
Gradient Based Methods

- Design sensitivity analysis (DSA) must be available
- Applied for linear static and dynamic problems
- Mostly integrated with FEA Solvers
- Not feasible for Non-Linear solvers

Flowchart:

- Simulation
  - Conv ?
  - Design Sensitivity Analysis (DSA)
  - Design Update
  - Optimum
Surface Approximation Methods

- Sequential Response Surface update
  - Linear step
  - Quadratic response surface
- Non-linear physics
- Experimental Analysis
- Wrap-Around Software
- HyperOpt is solver neutral
Exploratory Methods

- Gradients are not needed
- Suitable to nonlinear simulations
- Computationally expensive as it may require large number of analysis
- Convergence not guaranteed

1. Create a number of designs (population)
2. Simulate all and pick the best one
3. Conv? Or # of simulations > Max # of sim?
4. Population Update
5. Optimum
HyperStudy offers four optimization engines: Adaptive Response Surface, Method of Feasible Directions, Sequential Quadratic Programming, Genetic Algorithm. It also allows for User-defined Method.

With user-defined method, user’s may interface their optimization code with HyperStudy to perform optimization.

Also, the approximations can be used as surrogate model to perform optimization.
Adaptive Response Surface Method (HyperOpt)

In this approach, the objective and constraint functions are approximated in terms of design variables \( x \) using a second order polynomial.

The polynomial coefficients are determined using a least squares fit of the functions on to the previous design points (actual nonlinear analysis results).

In general, more designs are available than are required for an exact least squares fit, making the system over-determined. HyperOpt uses a very efficient algorithm to estimate a response surface to be closer to certain designs of interest. HyperOpt also uses move limits to make the optimization algorithm robust.
The method of feasible directions is one of the earliest methods for solving constrained optimization problems.

The fundamental principle behind this method is to move from one feasible design to an improved feasible design. Hence, the objective function must be reduced and the constraints at the new design point should not be violated.
Sequential quadratic programming (SQP) is a method for solving constrained optimization problems.

The fundamental principle behind this method is to create a quadratic approximation of the Lagrangian and to solve that quadratic problem to define the search direction $s$.

The constraints are linearized during the search.

This quadratic problem can be solved by a variety of methods. The solution of the problem yields the search direction along which the next design that improves the objective function and does not violate the constraint can be found.
A genetic algorithm is a machine learning technique modeled after the evolutionary process theory.

Genetic algorithms differ from conventional optimization techniques in that they work on a population of designs. These designs are then evaluated for their fitness, which is a measure of how good a particular design is. Following Darwin’s principle of survival of the fittest, designs with higher fitness values have a higher probability of being selected for mating purposes to produce the next generation of candidate solutions.

In addition to reproduction, selected individual designs go through crossover and mutation. The designs that result from this process (the children) become members of the next generation of candidate solutions.

This process is repeated for many generations in order to artificially force the evolution of a population of designs that yield a solution to a given problem.

Genetic algorithms do not show the typical convergence of other optimization algorithms. Users typically select a maximum number of iterations (generations) to be evaluated. A number of solver runs is executed in each generation, with each run representing a member of the population.
Ordinary response surface methods are based on DOE studies. Response surface is fitted (quadratic polynomial) with a fixed number of designs analyzed, consequently it is not sufficient to approximate and optimize highly non-linear functions.
Optimization Process

1. Assign factors (dv’s) from .tpl, .hm,.mdl,.hf file
2. Perform nominal run to create responses.
3. Select the Optimization Algorithm
4. Select the Design Variables for optimization study
5. Define Objective function and optimization constraints
6. Launch Optimization
7. Postprocess the optimization results
Optimization Problem Example

- A cantilever beam is modeled with 1D beam elements and loaded with force $F=2400$ N. Width and height of cross-section are optimized to minimize weight such that stresses do not exceed yield. Further the height $h$ should not be larger than twice the width $b$. 
Optimization Problem Example

- **Objective**
  - Weight: $\min m(b,h)$

- **Design Variables**
  - Width: $b^L < b < b^U$, $20 < b < 40$
  - Height: $h^L < h < h^U$, $30 < h < 90$

- **Design Region:** All beam elements

- **Design Constraints:**
  - $\sigma(b,h) \leq \sigma_{\text{max}}$, with $\sigma_{\text{max}} = 160$ MPa
  - $\tau(b,h) \leq \tau_{\text{max}}$, with $\tau_{\text{max}} = 60$ MPa
  - $h \leq 2b$
Optimization Problem Example

Mathematical Design Space

Beam height, h (mm)

Beam width, b (mm)
HyperStudy Case Studies and Examples
Case Studies Examples

- Optimization studies are increasing in popularity. It provides added value to the baseline study, it helps the engineer in understanding the physics of the problem, it allows exploration in the design space.

- The technology has been applied to full body crash. An example consists of an objective to minimize mass using footwell intrusion displacements as constraints and several material thickness values as design variables.

- The technique has been applied to headform assessments. Initially to identify the worst attitude of impact. Once this is established the rib thickness of the crashbox is determined.

- Four case studies are presented. Two relating to structural crashworthiness, two relating to occupant safety.

- In addition fours examples are given. Two of these examples relate to computational fluid dynamics and two of them relate to fatigue.
Case Study 1: Rail Optimization

- A box rail of length $l_x=800$mm is clamped at the right side and is impacted by a moving stone wall of mass $1,000$kg and initial velocity $2$m/s from the left side.

- The rail is required to absorb the maximum internal energy within an analysis time of $t_s=30$ms. The displacement of the stone wall ($x$-direction) should be less than $50$mm. The problem is typical of the designs encountered in car bumpers and crash boxes.

- A size optimization is performed with six design variables consisting of variable thickness patches. The initial thickness of all design variables was $2$mm and could vary between $0.2$mm and $3$mm.
Case Study 1: Rail Optimization

- The HyperOpt optimization (Sequential Optimization) is compared against local approximation and DOE approaches. Although structural behaviour is similar, HyperOpt maximizes the internal energy.

- Design iteration history is presented showing the variation of the objective with the constraint.
Case Study 1: Rail Optimization

Problem Setup

max Internal Energy
$\Delta l_x = 50\text{mm}$

Moving stone wall

$V_0 = 2\text{m/s}$
$m = 1000\text{kg}$

Fixed end
### Case Study 1: Rail Optimization

#### Size Optimization

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Ord. RS</th>
<th>Seq. RS</th>
<th>Local app.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV1 [mm]</td>
<td>2.00</td>
<td>1.70</td>
<td>2.02</td>
<td>2.25</td>
</tr>
<tr>
<td>DV2 [mm]</td>
<td>2.00</td>
<td>1.92</td>
<td>1.57</td>
<td>2.95</td>
</tr>
<tr>
<td>DV3 [mm]</td>
<td>2.00</td>
<td>0.20</td>
<td>1.56</td>
<td>2.11</td>
</tr>
<tr>
<td>DV4 [mm]</td>
<td>2.00</td>
<td>1.95</td>
<td>1.65</td>
<td>1.94</td>
</tr>
<tr>
<td>DV5 [mm]</td>
<td>2.00</td>
<td>2.63</td>
<td>3.00</td>
<td>2.28</td>
</tr>
<tr>
<td>DV6 [mm]</td>
<td>2.00</td>
<td>3.00</td>
<td>2.48</td>
<td>2.28</td>
</tr>
<tr>
<td>$E_{\text{max}}$ [Nm]</td>
<td>1010</td>
<td>1177</td>
<td>1254</td>
<td>1170</td>
</tr>
<tr>
<td>$\Delta l_x$ [mm]</td>
<td>52.08</td>
<td>49.95</td>
<td>49.88</td>
<td>49.85</td>
</tr>
</tbody>
</table>
Case Study 2: Roof Crush

- The US standard FMVSS216 requires a certification which simulates the roll over of a vehicle.
- The certification test consists of moving an undeformable plate into an A-pillar at a constant velocity and the load / displacement response monitored.
- The objective is to minimize the mass of the design and still ensure that the applied load meets the certification requirement. The design variables consist of five material thicknesses in the A-pillar and the roof frame.
Case Study 2: Roof Crush

Objective: minimize mass
Constraint: max $F(d) > F_{\text{limit}}$
Design variables: Thicknesses of A-Pillar

Diagram: A car with a plate applying force $F$.
Case Study 2: Roof Crush

Variation of the Normalized Constraint with Time

Force Data

- Initial Design
- Final Design (iteration 13)
- Lower Bound

Normalized Force

Time

0 10 20 30 40 50 60 70 80 90 100 110
Airbag impactor test requires the HIC not be exceed 1000. The impactor should not displace more than 260mm. The design variables are vent orifice area, fabric permeability, ignition time and mass flow rate of the gas.

- The test set-up consists of allocating an initial velocity to the impactor.
- After twenty-two iterations the HIC has been considerably reduced.
Case Study 3: Airbag

Objective: minimize head injury criterion (HIC)

\[
HIC = \max_{t_2-t_1 \leq 36ms} \left( \frac{1}{t_2-t_1} \int_{t_1}^{t_2} \ddot{d}_{\text{impactor}} dt \right)^{2.5} (t_2 - t_1)
\]

Constraint: Deflection \( d_{\text{max}} < 260\text{mm} \)

Design Variables: Ignition time
                Fabric permeability
                Mass flow rate
                Vent orifice area
Case Study 3: Airbag

![Graph showing deflection over time with iterations marked]

Deflection

Time [s]

0 0.01 0.02 0.03 0.04 0.05 0.06 0.07

0 0 50 100 150 200 250 300

Iteration 1

Iteration 22
Case Study 3: Airbag

Objective with Time

![Graph showing acceleration over time with two iterations, Iteration 1: HIC = 870, max a = 9.1 g and Iteration 22: HIC = 605, max a = 8.0 g.]}
Case Study 3: Airbag

Variation of the Objective / Constraint with Design Iteration History
Case Study 4: Knee Bolster Design

• This study considers the occupant behavior to determine the initial package space for a knee bolster. The model consists of the lower extremities of the dummy.

• The dummy is loaded by an acceleration pulse $a_x(t)$ resulting from the crash test in accordance with FMVSS208.

• The objective is to minimize the femur load with a limit on the knee penetration into the knee bolster. The design variables consist of geometric and stiffness parameters.
Case Study 4: Knee Bolster Design

Substitute Model
Optimization Problem

Objective: Minimize Femur Load:

\[ F_{\text{femur}}(\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_z, \Delta \alpha) \Rightarrow \min \]

Constraint: Knee Penetration:

\[ \delta(\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_z, \Delta \alpha) \leq 0.07 \, \text{m} \]

Design Variables: Geometric and Stiffness Parameters

\[ (\Delta \theta, \Delta k_x, \Delta l_x, \Delta l_z, \Delta \alpha) \]
Case Study 4: Knee Bolster Design

Results
# Design Variable Results

## Case Study 4: Knee Bolster Design

### Design Variable Results

<table>
<thead>
<tr>
<th>DESIGN VARIABLE</th>
<th>LOWER BOUND</th>
<th>UPPER BOUND</th>
<th>OPTIMAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta \psi^\circ$</td>
<td>-10.00</td>
<td>10.00</td>
<td>-10.00</td>
</tr>
<tr>
<td>$\Delta k_x$</td>
<td>-0.10</td>
<td>0.07</td>
<td>-0.10</td>
</tr>
<tr>
<td>$\Delta l_x$</td>
<td>-0.05</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>$\Delta l_z$</td>
<td>-0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$\Delta \alpha^\circ$</td>
<td>-5.00</td>
<td>5.00</td>
<td>-4.65</td>
</tr>
</tbody>
</table>
Case Study 5: Exhaust System Design

- **Objective:** Uniform flow velocity in section $S$
  - Average velocity
  - Local deviation:
  - Uniformity-Index:
- **Constraint:** Min. pressure drop between inflow and outflow

6 Shape Variables
Results

Case Study 5: Exhaust System Design

\[ \gamma \pm 12.0\% \]

\[ \Delta p \; -16.0\% \]

Flow Velocity in Section S

Initial

Optimized

Section S
Case Study 6: Notch Design

Loads & Boundary Conditions

 statically loaded with: $\sigma_s = 100 \text{ N/mm}^2$

 dynamically loaded with: $\sigma_s = 200 \text{ N/mm}^2$

Stresses

 statically loaded with: $471 \text{ N/mm}^2$

 dynamically loaded with: $234 \text{ N/mm}^2$

Damage

$D_{\text{max}} = 0.13$
Case Study 6: Notch Design

Design Variables for Shape Optimization

5 handles defining 5 design variables

Shape 1
Shape 2
Shape 3
Shape 4
Shape 5

Courtesy of MAGNA STEYR
Case Study 6: Notch Design

- Optimization Problem
  - Minimize maximum stress at notch edge
  - Both load cases considered
  - Shapes scaled +/- 1
- Damage deteriorated!

Stresses

Design Variable History

Stress Based Optimization

Damage

Courtesy of MAGNA STEYR

Altair Proprietary and Confidential Information

Copyright © 2008 Altair Engineering, Inc. All rights reserved.
Case Study 6: Notch Design

- HyperStudy identified dynamic load case to be sensitive to damage
- New contour different to stress optimized contour
- Damage significantly reduced

Damage Based Optimization

D_{\text{max}} = 0.0002
• Optimization commences once a clean baseline analysis has been performed (eg. negligible hourglassing and sliding energies)

• Care with shape variables: large perturbations can result in a reduced time step size

• Mass scaling used with shape optimization should be used with caution. Particularly if the objective is to minimize mass

• For size optimization, numerical contact can be effected by changing the thickness. Workaround by keeping the contact thickness constant

• Technology can be simultaneously applied to multiple loadcases (eg. central / oblique impact attitudes)
GENERAL DISCUSSION

• Changes in design variables can result in a completely different system response (eg. components not in contact may now contact)

• Original analysis duration may extend to capture the systems response optimization

• Computational time can be a barrier with each iteration requiring a complete explicit analysis

• Convergence is quickly achieved

• Optimized design can be conveniently restarted with new design variables or further optimization starting at a new position in the response surface
Exercise 4.1: Shape and Size Optimization of a Rail Joint
Exercise 4.2: Size Optimization Study using RADIOSS
Exercise 4.3: Optimization Study Using an Excel Spreadsheet
Exercise 4.4: Shape Optimization Study using HyperForm
Exercise 4.5: Shape Optimization Study using ABAQUS
Chapter 5: Stochastic
Models in HyperMesh, MotionView, or HyperForm

Parameterized Model using Create Template GUI

Input file for HyperStudy

DOE ➔ Approximation ➔ Optimization ➔ Stochastics

Direct Response Selection for Automated Post-Processing
Uncertainty in Design

- Physical uncertainty
  - Loads
  - Boundary and Initial condition
  - Material properties
  - Geometry

- Numerical simulation uncertainty
  - Conceptual modeling
  - Mathematical modeling

- Manufacturing
  - Sheet metal thickness
  - Welds
  - Random design (controlled) variables

- Loads
  - Direction
  - Magnitude
  - Random noise (uncontrolled) variables

- Material data
  - Elastic properties
  - Failure
  - Random noise or design variables
Minimizing variations in performance caused by variations in design variables
Robustness Definition

- The design should be invariant against small changes of the parameters which are inevitable in the normal design process.

- Scattering of design parameters due to real world conditions in manufacturing and boundary conditions should not reduce the performance of the product.

- Robustness is the degree to which a system is insensitive to effects that are not considered in the design.

- Robust statistical procedures has to be designed to reduce the sensitivity of the parameter estimates to unreliability in the assumptions of the model.
Type I Robust Design
(Flexible Specifications)

Given
Model f(x)

Find
A range of control factors x, \( \Delta x \) (top-level specifications)

Satisfy
System constraints
Goals
  Bring the mean \( \mu_y \) on target
  Minimize the variance \( \sigma_y \)

Minimize
Deviation Functions
Robustness Definition

Non-robust Design

Robust Design
Deterministic Optimization Approach

- Find acceptable solution to the design problem
- Design variables are continuous or discrete quantities
- Environmental conditions are given as deterministic load cases
- Objective
  \[ f(x) \rightarrow \min \]
- Constraints
  \[ g_j(x) \leq 0 \]
- Design space
Deterministic Optimization Approach

- Deterministic optimization problem formulation does not incorporate uncertainty of design variables.

- Optimal solutions for systems exhibiting highly non-linear responses can be misleading.

- Surrogate models (response surfaces) built using certain approximation techniques tend to smooth the noisy behavior.

- Most optimization algorithms push the constraints to the bounds in search of optimal solution.
Since the objective of any robust design problem is to bring the mean on target and minimize the variance, robust design problems should be modeled as bi-objective problem.

The two objectives are:

- Minimize the variance of performance
- Bring the mean on target or optimizing the “mean of performance”
Robust Design Optimization Formulation

- **Objective**: 
  \[ \sigma(f(x,r)) \Rightarrow \min \]

- **Constraints**: 
  \[ \mu(x,r) - f_{\text{Target}} \leq 0 \]
  \[ \mu_j(x,r) - g_{j_{\text{Target}}} \leq 0 \]
  \[ x^L \leq x \leq x^U \]

- **Design variables**: 
  \[ r = [r_i(\mu_i,\sigma_i)] \]

- **Random variables**: 

![Graph showing Objective: Deviation and Constraint: Mean](image)

Copyright © 2008 Altair Engineering, Inc. All rights reserved.
Exercise 5.1: Stochastic Study of a Rail Joint
Exercise 5.2: Stochastic Study of a Plate Model
REFERENCES


