LSTC
Livermore Software
Technology Corp.

LS-OPT® Training Class

OPTIMIZATION THEORY

NIELEN STANDER   TUSHAR GOEL

June 2010

Contents

Contents

Contents

Contents

Overview

Design Improvement and Optimization

- Best multi-criterion designs
  - Multiple objectives
  - Multiple constraints
- Parameters
  - Continuous
  - Discrete (underlying continuous, e.g. off-the-shelf plate thickness)
  - Integer (e.g. material types, binary)
- Multiple cases/disciplines

Overview

Reliability and Robustness

- Reliability:
  - Constrain probability of failure
- Robust Design:
  - Minimize Standard Deviation of response
  - Consistent product performance
- Reliability-based Design Optimization (RBDO)
  - Incorporates Reliability and Robustness into design improvement
- Identify sources of uncertainty in the FE models: Outlier Analysis

OPTIMIZATION FUNDAMENTALS

Gradient Computation

Formulation and types

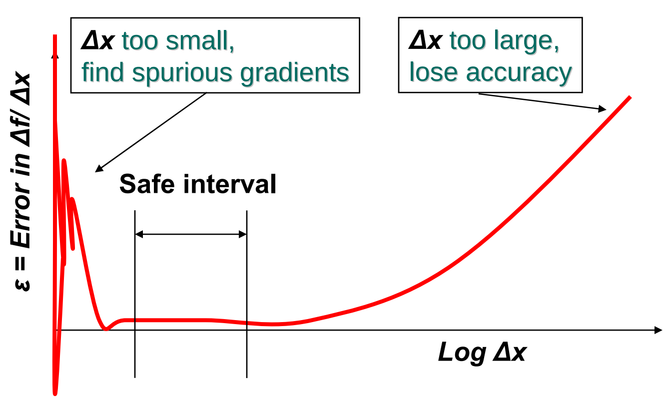
- Gradient based **Optimization** and certain **Reliability** algorithms require gradient computation
- Types
  - Analytical: Derivatives are formulated explicitly and implemented into the code. Complicated.
  - Numerical: Design is perturbed and (n+1) analyses are simulated.

$$\frac{\mathrm{d}f(x,y)}{\mathrm{d}x} \approx \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x} = \frac{\Delta f}{\Delta x}; \qquad \varepsilon = \frac{\mathrm{d}f}{\mathrm{d}x} - \frac{\Delta f}{\Delta x}$$

- Simple but error prone.

Gradient Computation

Numerical gradients: accuracy

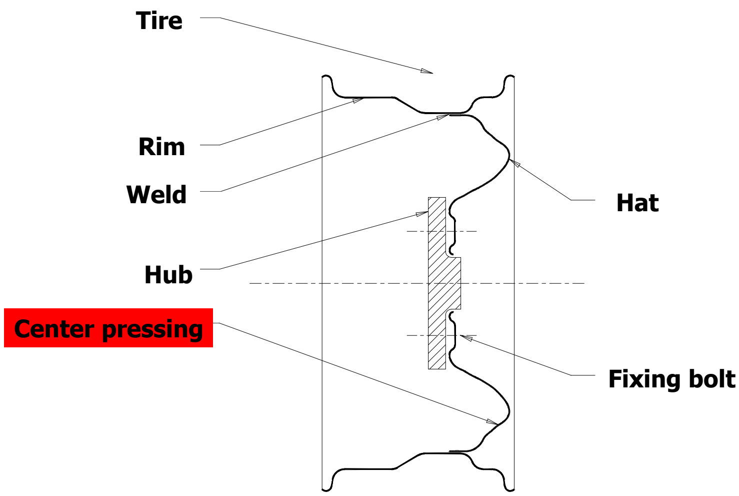


Gradient Computation
Causes of spurious derivatives

Spurious derivatives computed using small intervals are due to:

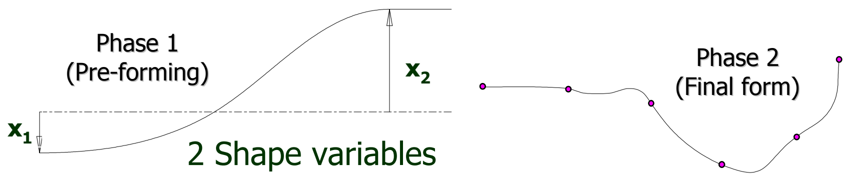
- Highly nonlinear structural behavior. Especially in crash analysis.
- Adaptive mesh refinement. Different designs have different meshes.
- Numerical Round-off error. LS-DYNA uses single precision computation.

Response Noise
Wheel: Stamping of Center Pressing (1996)

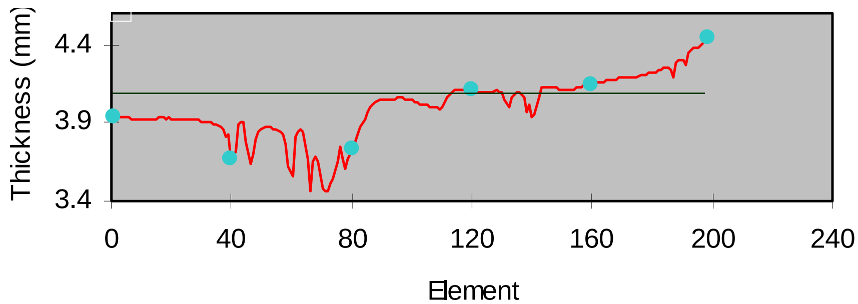


Response Noise

Wheel: Final thickness distribution



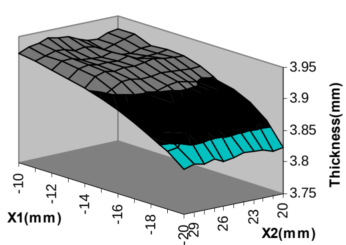
2 Shape variables



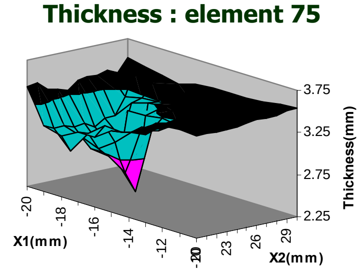
Response Noise: Stamping Problem

Final thickness a function of pre-forming die shape

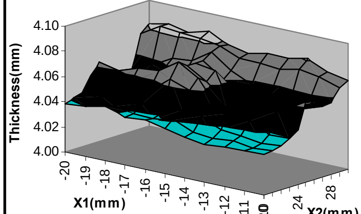
Thickness : element 19



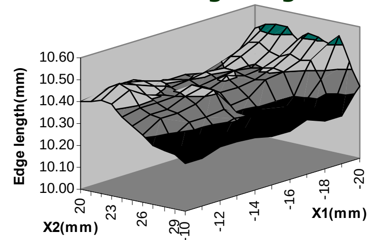
Thickness : element 75



Thickness : element 109



Weld edge length

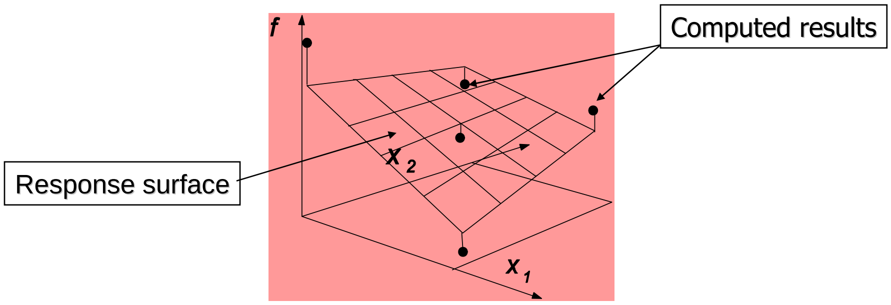


Approximations

- Sometimes referred to as metamodels
- Local Approximations
  - Design Sensitivities (also called Gradients)
    - Numerical. Perturb the design. Uses *n+1 simulations*
    - Analytical. Incorporated in the analysis code
- Midrange Approximations
  - Uses a region of interest in the design space to construct the approximations
  - Approximations can be simple polynomials, e.g. linear
  - Used in iterative methods, e.g. Sequential Response Surface Method in LS-OPT
- Global Approximations
  - Use the full design space
    - Neural Networks
    - Radial Basis Function Networks
    - **Response Surfaces** (especially higher order polynomials)

Response Surface Methodology
How does it work?

Design surfaces (*f*) are fitted through points in the design space (results from simulations) to construct an approximate optimization problem

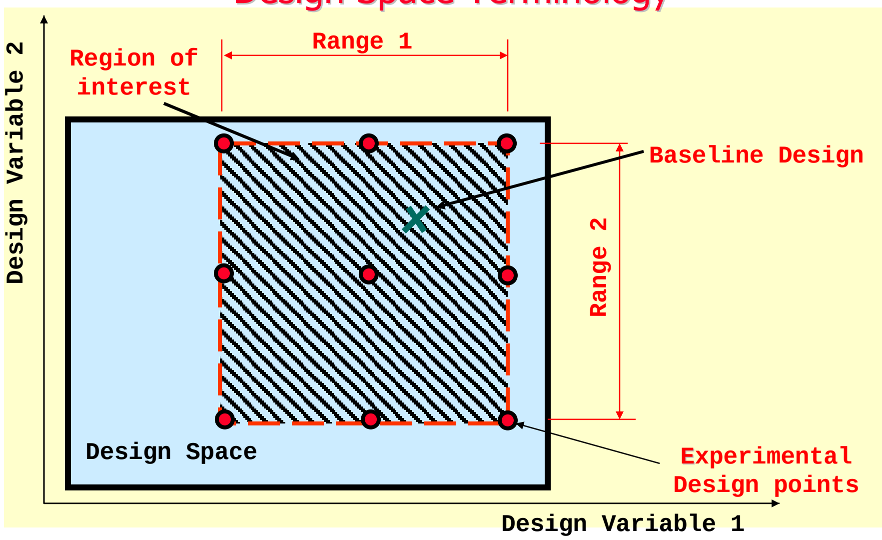


The idea is to find the surfaces with the best predictive capability

Response Surface Methodology

- Creates **metamodel** based on **polynomial approximations**
- Does not require **analytical sensitivity analysis (analytical derivatives).**
- Smoothes the design response, hence stabilizes **numerical sensitivities**
- Avoids selection of outlier design points by averaging the design response
- Accurate design surfaces in a sub-region allow for **inexpensive exploration of the design space**
  - Response Surface optimization
  - Reliability

Design Space Terminology



Response Surface Methodology
Least squares

$$y = \eta(x).$$

The exact relationship is approximated as

$$\eta(x) \approx f(x).$$

The approximating function f is:

$$f(x) = \sum_{i=1}^{L} a_i \phi_i(x)$$

where L is the number of basis functions ϕ_i used to approximate the model.

19

Response Surface Methodology
Least squares

Sum of the square error:

$$\sum_{p=1}^{P}\left\{\left[\mathbf{y}_p(\mathbf{x}) - \mathbf{f}_p(\mathbf{x})\right]^2\right\} = \sum_{p=1}^{P}\left\{\left[\mathbf{y}_p(\mathbf{x}) - \sum_{i=1}^{L}\mathbf{a}_i\varphi_i(\mathbf{x}_p)\right]^2\right\}.$$

P: number of experimental points

$\boldsymbol{y}_p$ is the exact functional response at the experimental point $\boldsymbol{x}_p$.

20

Response Surface Methodology
Least squares

The solution:

$$a = (X^T X)^{-1} X^T y$$

where X is the matrix

$$X = [X_{ui}] = [\phi_i(x_u)].$$

Choose appropriate basis functions, e.g.

$$\phi = [1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_1 x_n, \ldots, x_n^2]^T$$

Response Surface Methodology
Approximation models

$$1 \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \begin{bmatrix} x_1^2 & x_1 x_2 & \ldots & x_1 x_n \\ x_2 x_1 & x_2^2 & \ldots & x_2 x_n \\ \vdots & \vdots & \ldots & \vdots \\ x_n x_1 & x_n x_2 & \ldots & x_n^2 \end{bmatrix}$$

Linear **Quadratic**

Response Surface Methodology

1st vs. 2nd order approximations

- **First order approximations**
  - The most basic approximation
  - Inexpensive. Cost ~ *n*
  - Cycling (oscillation) can occur when used in sequential method for optimization. Successfully addressed by adaptive optimization algorithm (SRSM)
- **Second order approximations**
  - More expensive. Full Quadratic: Cost ~ *n*-squared
  - More accurate. Good for trade-off studies

Linear Approximation is recommended in many cases, e.g.

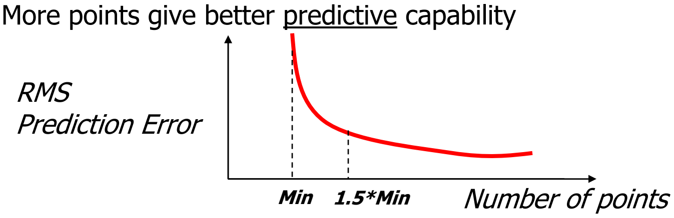
Sequential approximations for Optimization, Reliability

Response Surface Methodology
Factors influencing accuracy

- Size of the region of interest.

  The smaller the size, the more accurate the surface
- Number and distribution of experimental points.

  More points give better predictive capability



- Order and nature of the approximating function.

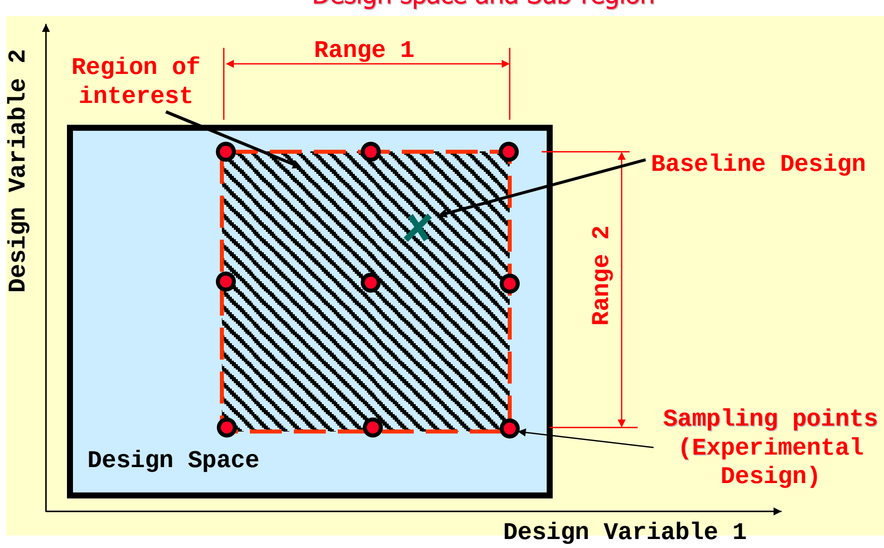
  Higher order is more accurate, but overfitting can occur

Response Surface Methodology
Factors influencing accuracy

- *Overfitting.* Prediction Error increases due to overfitting (the addition of more terms to the approximation model).
- *Noise.* Reduction of the size of the region of interest will improve accuracy up to a point where only the noise dominates.

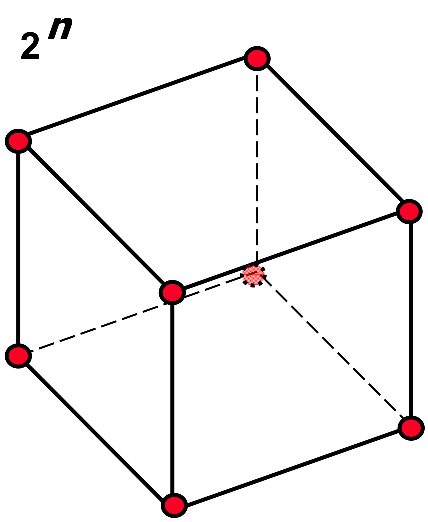
Experimental Design

Design space and Sub-region



Experimental Design (Point Selection)

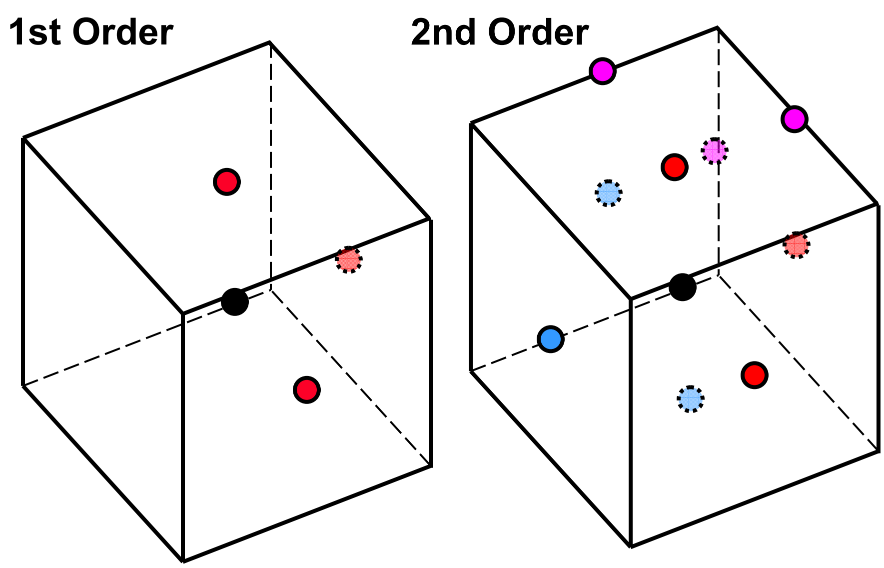
Factorial (n=3)



27

Experimental Design

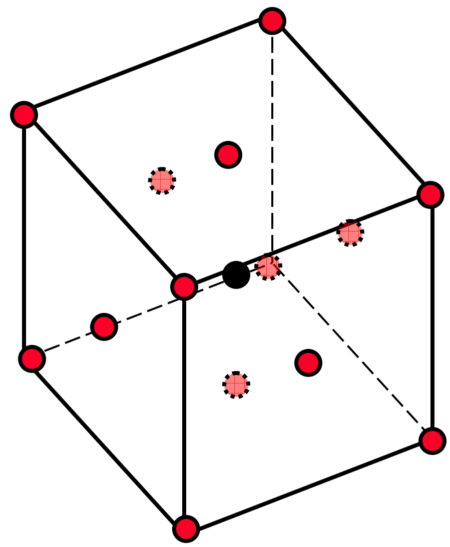
Koshal (n=3)



28

Experimental Design

Central composite design (*n*=3)



Experimental Design

D-optimal design

Uses a subset of all possible design points as a basis to solve (using genetic algorithm)

$$\max |\boldsymbol{X}^T \boldsymbol{X}|$$

where X is the matrix

$$X = [X_{ui}] = [\phi_i(x_u)]$$

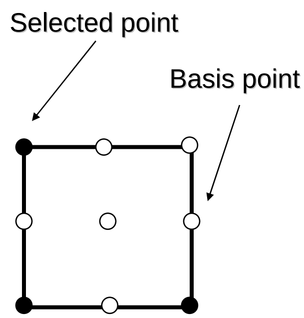
Experimental Design
D-optimal design

- Find the 'best' distribution of a fixed number of sampling points
- The 'basis experiment' is used as a superset from which the *D*-optimal points are selected.
- Oversampling improves the predictive capability of the response surface. 50% is used as a thumb rule.
- Previous points can be used and new points added *D*-optimally (augmented *D*-optimal).
- Irregular design spaces (e.g. bounded by nonlinear constraints) can be used. In this case the basis set is irregularly distributed. (see Reasonable Design Space)

Experimental Design
D-optimal design: Basis points

- Default subset (basis experiment) taken from factorial design
  - Linear: $m^n; \quad m = 11,9,7,5,3,2$
  - Quadratic $m^n; \quad m = 11,9,7,5,3$

  as n increases
- Space Filling used for large n
- Can choose discrete points when using discrete variables.



Experimental Design
D-optimal augmented points

- The D-optimality criterion for the additional points is:

Add rows to X:

$$X_a(x_p) = \begin{bmatrix} X \\ A(x_p) \end{bmatrix}$$

and solving

$$\max |X_a^T X_a| = \max |X^T X + A^T A|.$$

for x_p.

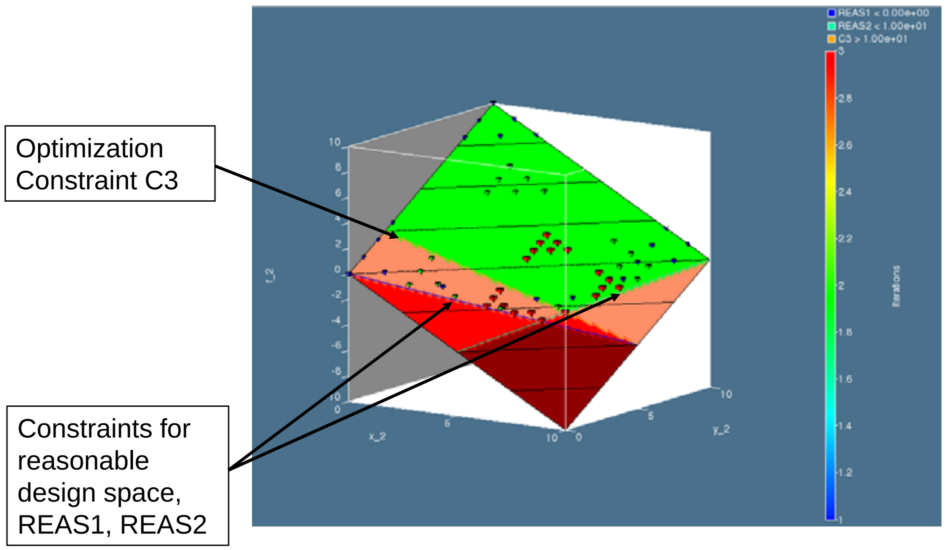
- E.g. starting point + Augmented D-Optimal

Reasonable Design Space

- Some designs may not be analyzable, e.g. incompatible geometries can be created
- Causes the solver to terminate with an error or give nonsensical results
- Can be prevented by specifying a reasonable design space.
- A flag can be set for any constraint

Reasonable Design Space

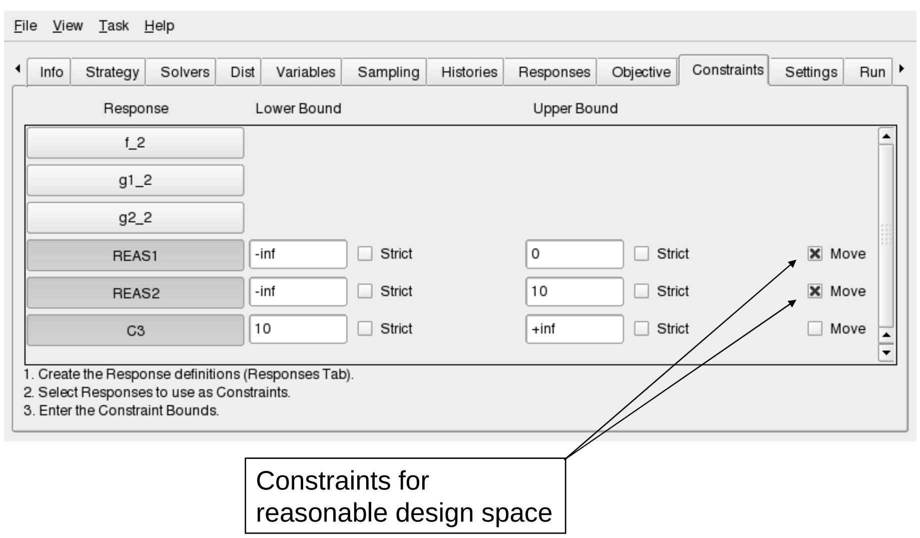
LS-OPT display



35

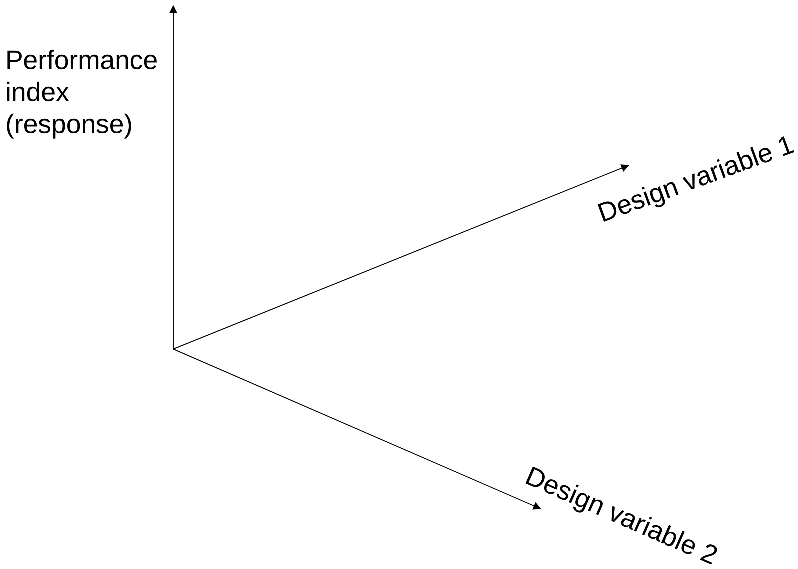
Reasonable Design Space

Selection in LS-OPT



36

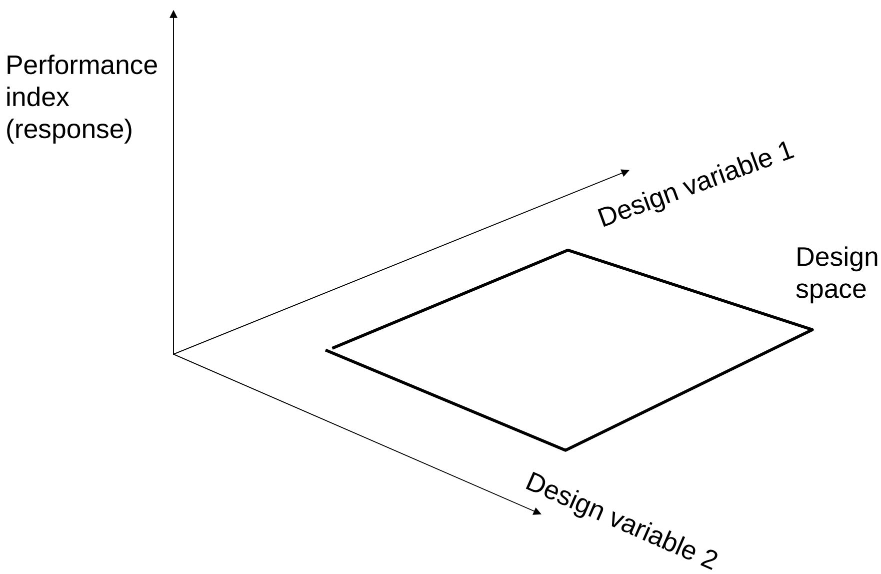
Design Cycle



37

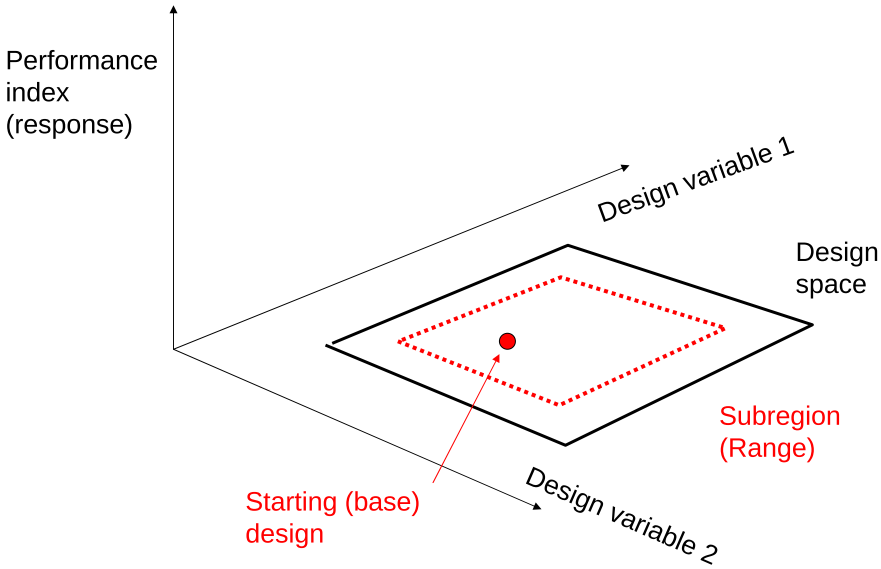
Design Cycle

Design space

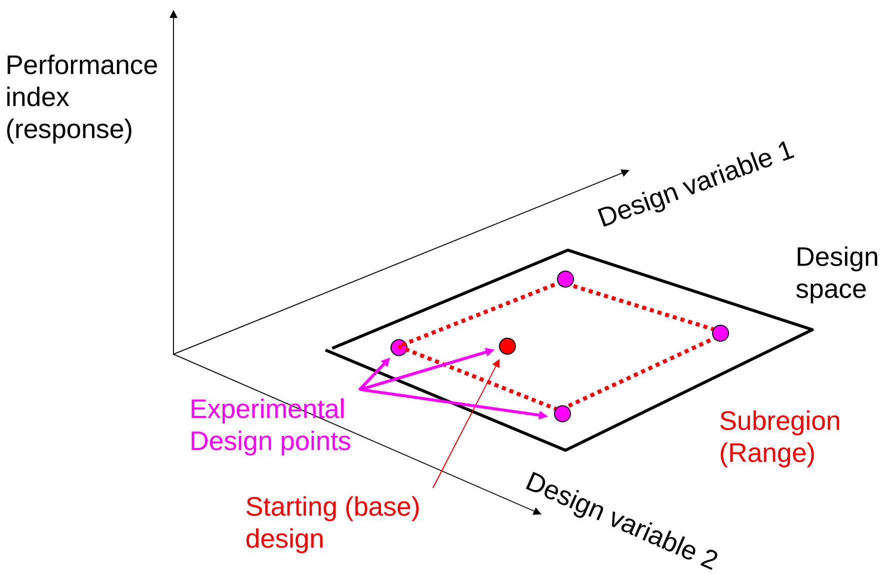


38

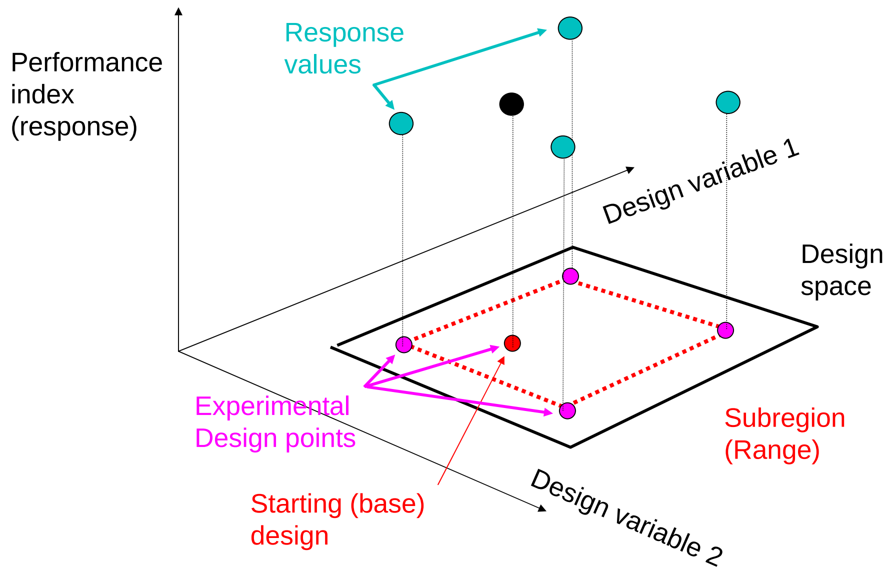
Design Cycle
Sub-region



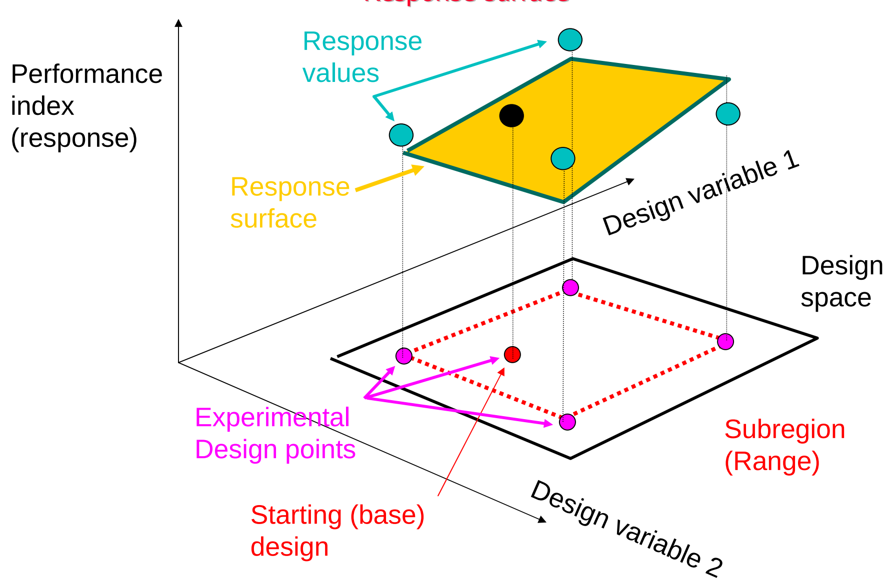
Design Cycle
Experimental design



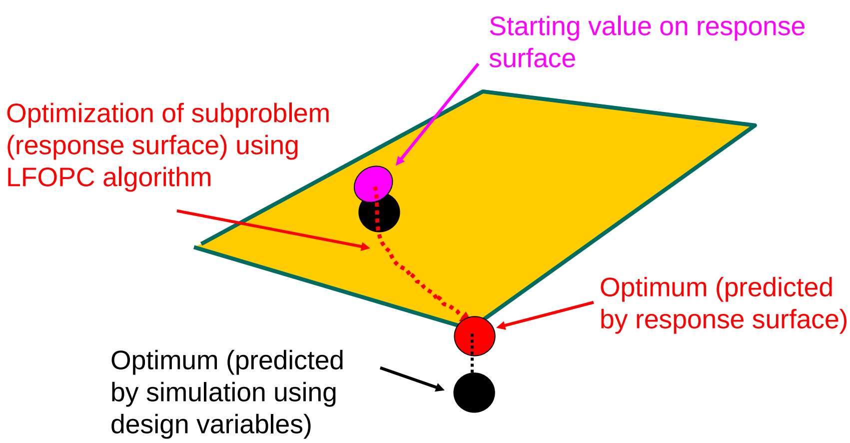
Design Cycle
Simulation results



Design Cycle
Response surface



Design Cycle
Optimization



Variable Screening

The $100(1-\alpha)\%$ confidence interval for the regression coefficients $b_j, j = 0, 1, ..., L$ is determined by

$$b_j - \frac{\Delta b_j}{2} \leq \beta_j \leq b_j + \frac{\Delta b_j}{2}$$

where

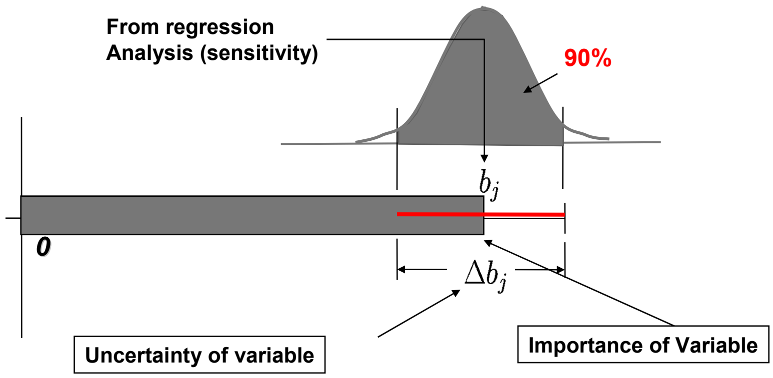
$$\Delta b_j = 2t_{\alpha/2,P-L}\sqrt{\hat{\sigma}^2 C_{jj}}$$

and $\hat{\sigma}^2$ is an unbiased estimator of the variance σ^2 given by

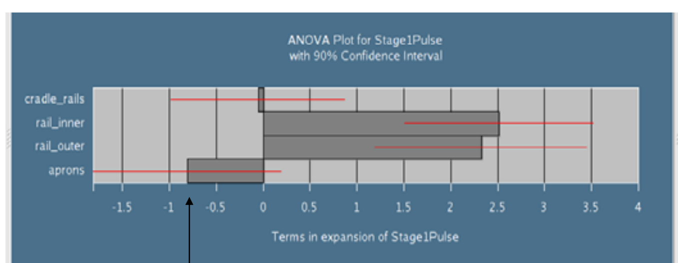
$$\hat{\sigma}^2 = \frac{\varepsilon^2}{P-L} = \frac{\Sigma_{i=1}^{P}(y_i - \hat{y}_i)^2}{P-L}.$$

C_{jj} is the diagonal element of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ corresponding to b_j and $t_{\alpha/2,P-L}$ is Student's t-Distribution.

Variable Screening



Variable Screening: Sensitivities Chart



Normalized Sensitivity

Error bar: 90% Confidence Interval

Note: Values are normalized with respect to design space

Variable Screening: Example

Crash model

30 000 elements

Displacement = 552mm

Stage1Pulse = 14.34g

Stage2Pulse = 17.57g

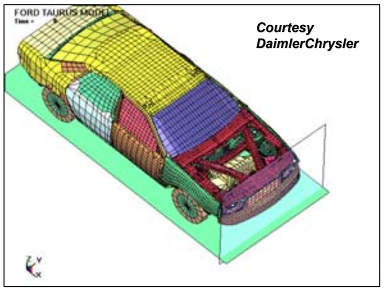
Stage3Pulse = 20.76g

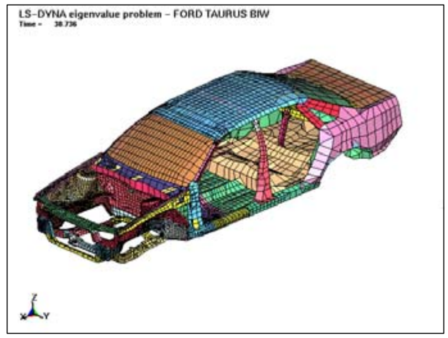
BIW model

18 000 elements

Torsional mode 1

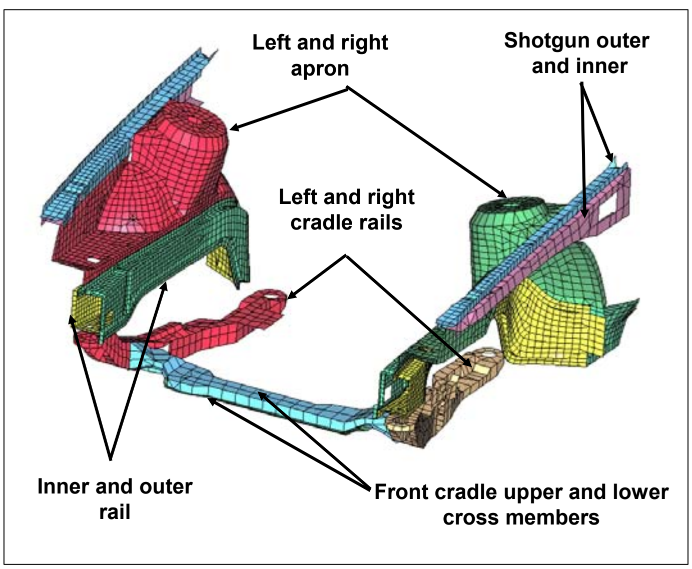
Frequency = 38.7Hz





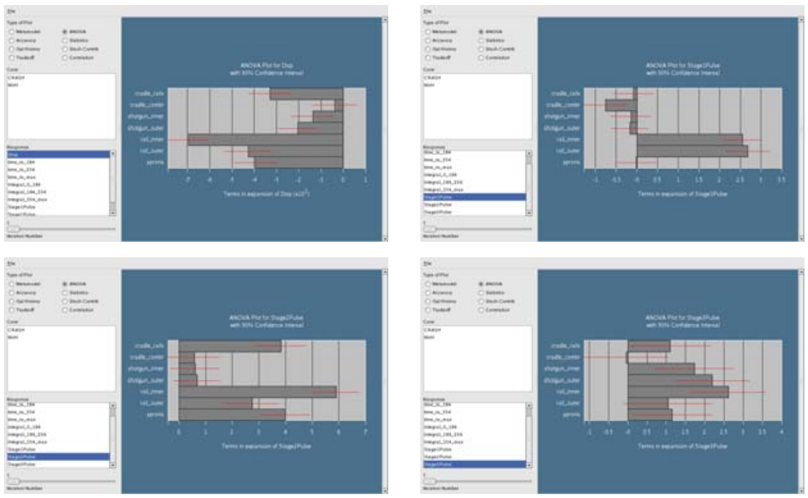
47

Variable Screening: Parameters

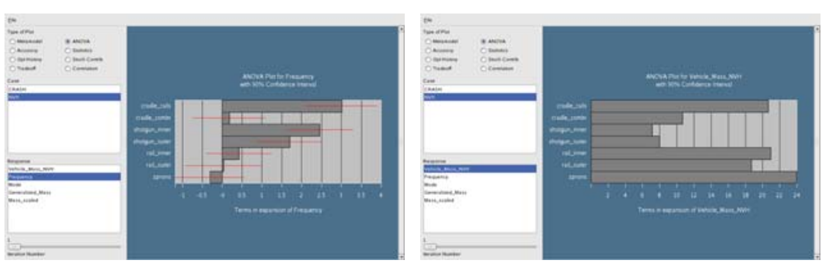


48

Variable Screening: ANOVA



Variable Screening: ANOVA



META-MODELING

Metamodeling

What is a metamodel ?

An *approximation* to the design response, usually a simple function of the design variables. Is used instead of actual simulations during exploration hence also called *surrogate*.

Metamodel Types in LS-OPT

Response Surface Methodology (RSM)

- Polynomial-based
- Typically regional approximation (especially linear)
- Linear regression

Feedforward Neural Networks (FF)

- Simulation of a biological network, sigmoid basis function
- Global approximation
- Nonlinear Regression: more expensive

Radial Basis Function Networks (RBF)

- Bell curve type basis functions in a linear system
- Global approximation
- Linear Regression (assuming constant spread and center)

User-defined

- Dynamically linked (`.so, .dll`)

Metamodeling
Motivation

Why Neural Nets / RBFN's

- Model for any number of simulation runs
  - Different polynomial orders require discrete numbers of runs (e.g. 10var: L=11+, Q=66+)
- Local refinement
  - Refine regionally, but maintain global relevance
- High accuracy (with enough points)
- Regression (smoothing) vs. Interpolation
  - Smoothing required to quantify noise

Metamodel Applications

Variable screening

Optimization

- Sequential construction/updating
- Pareto-optimal front using GA

Outlier Analysis

- Locate sources of response noise

Reliability Estimation

- Monte-Carlo simulations
- Robust design

Radial Basis Function Networks

Network construction

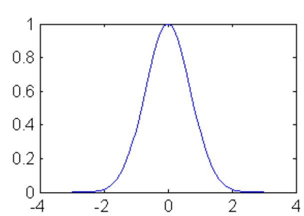
- Linear output layer:

$$Y(\mathbf{x}, a) = a_0 + \sum_{h=1}^{H} a_h \cdot f(\rho_h)$$

- Hidden layer:

$$f(\rho) = e^{-\rho}$$

$$\rho_h = r_{h0} \sum_{k=1}^{K} (x_k - X_{hk})^2$$



- Center:

$$\boldsymbol{X}_h = (X_{h1}, \ldots, X_{hk})$$

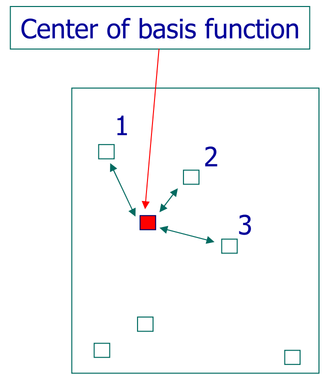
Radial Basis Function Networks

$$r_{h0} = 1/2\sigma_h^{\ 2}; \quad \sigma_h = Cd_{hm}$$

d_{hm} is the mean

of the distances to

the m closest points



"Spread" coefficient

$$C \approx 1.5$$

57

Radial Basis Function Networks

Mean Squared Error:

predicted
computed

$$MSE = \sum_{i}^{P} (\hat{y}_i - y_i)^2 / P,$$

Requires linear regression to solve for coefficients

$$a_0, \ldots, a_H$$

if $\sigma_h, \boldsymbol{X}_h$ is constant.

$\boldsymbol{X}_h$ typically centered on the design point

58

Feedforward Neural Networks

Network construction

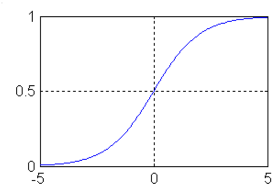
Linear output layer:

$$Y(\mathbf{x},\mathbf{W}) = W_0 + \sum_{h=1}^{H} W_h \cdot f\left[W_{h0} + \sum_{k=1}^{K} W_{hk} x_k \right]$$

Hidden layer (sigmoid):

$$f(x) = \frac{1}{1+e^{-x}}$$

⇨



Feedforward Neural Networks

Mean Squared Error:

predicted
computed

$$MSE = \sum_{i}^{P} (\hat{y}_i - y_i)^2 / P,$$

Requires nonlinear regression to solve for coefficients

$$W_h, W_{h0}, W_{hk}$$

RPROP, Levenberg-Marquardt, BFGS

Feedforward Neural Networks

Regularization

Minimize:

$$F = MSE + \alpha \sum_{m=1}^{M} W_m^{\;2}$$

- Causes the network to have smaller weights
- Response will be smoother
- Aids numerical robustness of the non-linear regression
- α cannot be too big, otherwise increasing modeling error

Feedforward Neural Networks

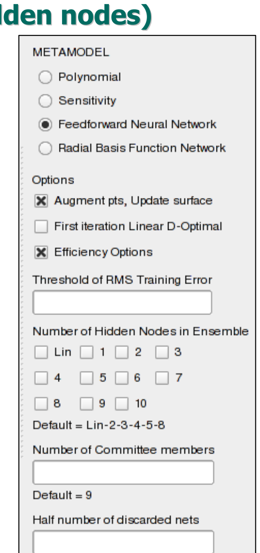
Finding a suitable topology (number of hidden nodes)

- Construct *ensemble* of architectures (different numbers of hidden nodes)
  - Single layer architectures (0 – 5 hidden nodes)
  - Select the "best" net using Min. Generalized Cross Validation (GCV)

υ = Effective number of model parameters

$$\frac{MSE}{(1-\upsilon / P)^2}$$

  - Leave-one-out is too expensive



Feedforward Neural Networks

Variability of FF

- Neural nets have natural variability due to
  - Local behavior of the FF training algorithms
  - Uncertainty (noise) in the training data
- Variability is induced by random initial weights in the regression procedure
- Sequential Response Surface Method: Recommended to use Linear surface (D-optimal sampling) in iteration 1 (default in GUI)

Feedforward Neural Networks

Variability (contd.)

- Committees (families of nets) are used to average the result. (default = 9)
- Nets with highest and lowest training MSE are discarded (default = 2x2) trying to avoid over/underfitting
- Committees dramatically affect the cost of computation

Metamodels: Efficiency

Example

12 variables
305 points
31 responses
2.6GHz AMD Opteron

FF Neural Net (NN9) (9 committee members: preferred accuracy setting)	**FF Neural Net (NN1)** (1 committee member: typically lower accuracy option)	**RBF**
minutes	minutes	minutes
220	22	3

Metamodels: Experimental Design

NN + RBFN

- Space Filling
- Simulated Annealing to locate new points
- Max. Min. distance between
  - new points
  - new points + fixed points
- New points bounded by sub-region
- $1.5(n+1)$ points per iteration: relatively sparse!

Response Surface Method

- Use D-Optimality (GA)
- $1.5(n+1)$ points per iteration (for linear)
- No updating (do not consider previous points)

Metamodels: Summary

Global approximation

- Can apply global trade-off study or robustness analysis after optimization run
- Maintains local smoothness, filters noise

Accuracy

- Nonlinear method. Will develop curvature as soon as enough points are available. Faster convergence.
- RBF more accurate than NN in many cases because of cross-validation.
- NN appears to be more accurate for smooth problems.

User decision

- Choice of NN/RBFN architectures is automated
- Independent of number of points chosen (default same as for linear)
- No initial range specification required.

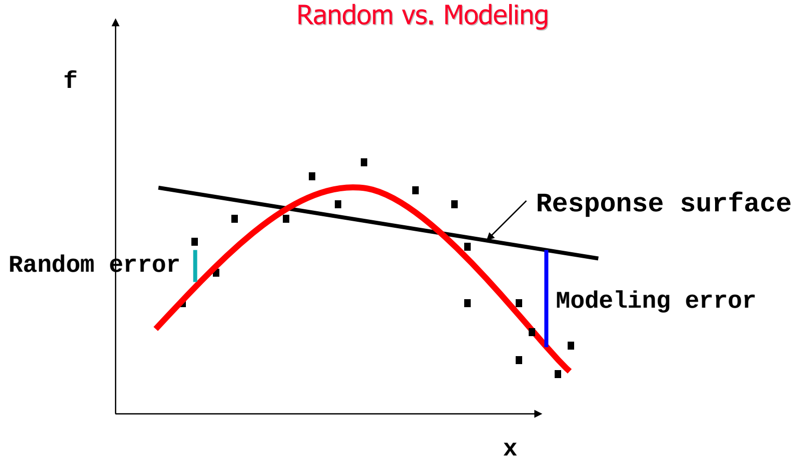
A regression method

- NN/RBFN Smoothes (as opposed to e.g. Kriging, which interpolates)
- Committees allow the extraction of point-wise variance information
- High variance an indication of sparsity (NN's only)

Metamodels: Summary

Response Surface Methodology (RSM)	Feedforward Neural Networks (FF)	Radial Basis Function Networks (RBF)
Polynomial basis functions	Simulation of a biological network. Sigmoid basis fns.	Local Gaussian or multi-quadric basis functions
Regional approximation	Global approximation	Global approximation
Linear regression. Accuracy is limited by order of polynomial.	Nonlinear regression. High accuracy. Robustness requires committee (e.g. NN9)	Linear regression within nonlinear loop. Cross-validation for high accuracy
Very fast	Very slow	Fast

Error Analysis
Random vs. Modeling



Error Analysis
Parameters in LS-OPT output

- RMS error
- Average error
- Maximum error
- PRESS error
- R^2 indicator

Error Analysis
Root mean square error and Maximum error

$$\varepsilon_{\mathrm{RMS}} = \sqrt{\frac{1}{P}\sum_{i=1}^{P}(y_i - \hat{y}_i)^2}.$$

$$\varepsilon_{\max} = \max\left|y_i - \hat{y}_i\right|.$$

Error Analysis
R^2

The coefficient of determination R^2 is defined as:

$$R^2 = \frac{\sum_{i=1}^{P}(\hat{y}_i - \bar{y}_i)^2}{\sum_{i=1}^{P}(y_i - \bar{y}_i)^2}$$

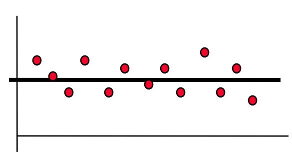
P: number of design points
$\hat{y}$: predicted response
$\bar{y}_i$: mean of the responses
y_i: the actual response

Error Analysis
Significance of R^2

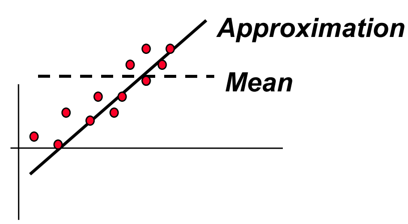
- A fraction of the variation in the data explained by the model
- A measure of the ability of the response surface to quantify the variability of the design response

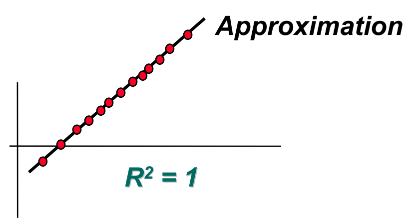
Error Analysis
R^2

$R^2 \longrightarrow 0$



$0 << R^2 < 1$





Accuracy of Response Surface
Noise detection

RMS	R^2	
Small	**~ 1**	High variability detection: low noise, good fit
Small	**~ 0**	Low noise/good fit, small gradient
Large	**~ 1**	High variability detection with noise
Large	**~ 0**	Lack of fit, perhaps accompanied by noise. Must shrink the move limits.

Error Analysis
Remarks: R^2 *and RMS*

Note: Too few points implies that

RMS → 0 and R^2 → 1

- The parameters reveal nothing about the predictive capability of the curves.
- 50% oversampling is the default for *D*-optimal experimental design in LS-OPT.

Error Analysis
Prediction errors

PRESS (PREdiction Sum of Squares):

Estimates the predictive capability of the reponse surface. Also known as **Leave-one-out** (LOO)

1. Remove one point from the least squares calculation. Fit a surface to the remaining points and predict the error at the chosen point.
2. Sum the square of errors.

An alternative formulation is done without the outer loop:

Error Analysis
Prediction errors

$$\text{PRESS} = \sum_{i=1}^{P} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2$$

h_{ii} are the diagonal terms of

$$H = X \left(X^T X \right)^{-1} X^T.$$

H is the "hat" matrix, the matrix which maps the observed responses to the fitted responses, i.e.

$$\hat{y} = Hy$$

Error Analysis
Prediction errors

Square root form is presented in the output:

$$\mathrm{SPRESS} = \sqrt{\frac{1}{P}\sum_{i=1}^{P}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2}.$$

R-squared indicator for prediction

For the purpose of *prediction* accuracy the $R^2_{\text{prediction}}$ indicator is used:

$$R^2_{\text{prediction}} = 1 - \frac{\mathrm{PRESS}}{S_{yy}}$$

where

$$S_{yy} = y^T y - \frac{\left(\sum_{i=1}^{P} y_i\right)^2}{P}$$

$R^2_{\text{prediction}}$ represents the ability of the model to detect the variability in predicting new responses

OPTIMIZATION

Design Formulation
Entities

Design variables

Design parameters which can be changed e.g. size or shape

$$\boldsymbol{x} = \{x_1, x_2, x_3, ..., x_n\}$$

Design objectives

A measure of goodness of the design, e.g. cost, weight, lifetime. Can involve more than one function $f_i(x)$.

$$\min p[f_i(\boldsymbol{x})] \quad ; \quad i = 1, 2, 3, ..., N$$

Design constraints

Limits on the design, e.g. strength, intrusion, deceleration

$$L_j \leq g_j(\boldsymbol{x}) \leq U_j \quad ; \quad j = 1, 2, 3, ..., m$$

Mathematical Optimization
Constrained minimization

$$\min f(x)$$

subject to

$$g_j(x) \le 0 \;;\; j = 1,2,\ldots,m$$

and

$$h_k(x) = 0 \;;\; k = 1,2,\ldots,l$$

f: cost or objective function
g: inequality constraint function
h: equality constraint function
x: design variables (parameters)

Mathematical Optimization
Equality constraints

$$h_k(x) = 0 \;\sim\; 0 \le h_k(x) \le 0.$$

allows simplification to

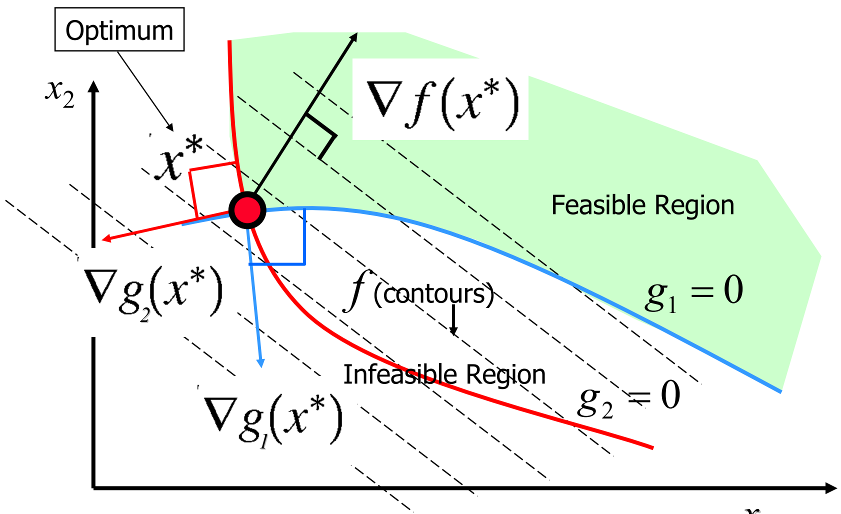
$$\min f(x)$$

subject to

$$g_j(x) \le 0 \;;\; j = 1,2,\ldots,m$$

Mathematical Optimization

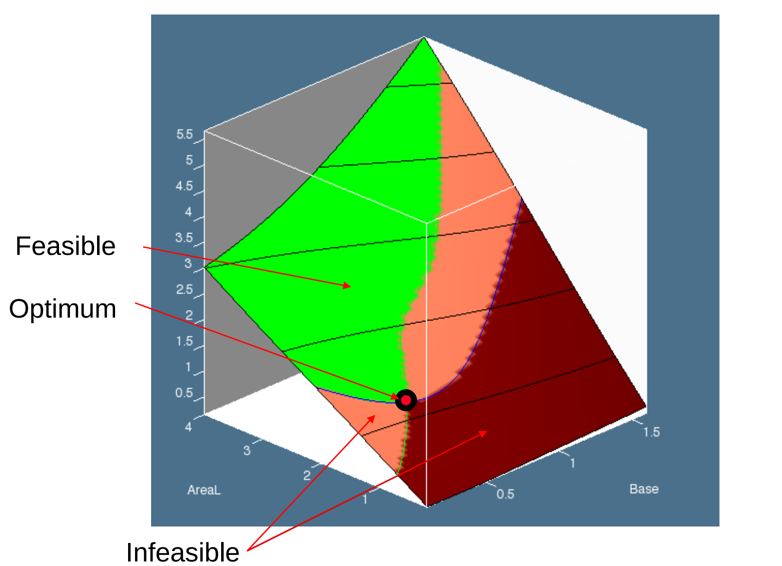
Example: inequality constrained problem in 2D



85

Mathematical Optimization

Example: inequality constrained problem (LS-OPT display)



86

Optimality Criteria

Karush-Kuhn-Tucker conditions for constrained optimization:

$$\nabla f(x^*) + \lambda^T \nabla g(x^*) = \mathbf{0}$$

$$\lambda^T g(x^*) = 0$$

Feasibility: $g(x^*) \leq \mathbf{0}$

Lagrange Multipliers: $\lambda \geq \mathbf{0}.$

Unconstrained optimization: $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$

Optimization Algorithms in LSOPT

There are three core-optimization algorithms

- LFOPC is a gradient based optimizer. Multiple starting points are used to avoid local optima.
- Genetic Algorithm (GA) is a population based global optimizer that emulates nature
- Adaptive Simulated Annealing (ASA) is a probabilistic optimizer that simulates metallurgical process
- Two hybrid algorithms Hybrid ASA and Hybrid GA are also available.
  - In the hybrid approach, a ASA or GA run is followed by a single LFOPC run. The idea is to find a good starting point using global optimizers and then switch to local optimizer to speed convergence.

Optimization Algorithm – LFOPC

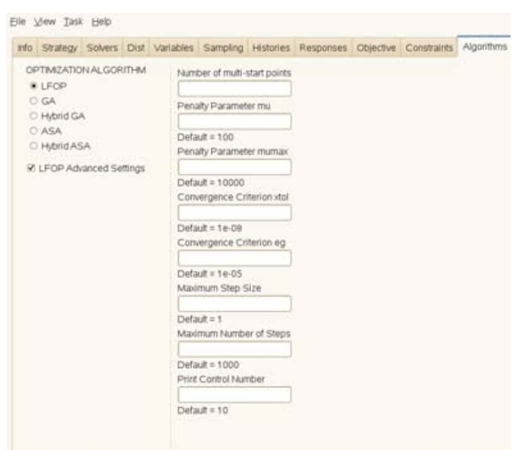
Leap-Frog method for Constrained Optimization (LFOPC)

- Gradient method
- Generates a dynamic trajectory path
- Does not use any line searches
- Penalty formulation for constraints

Optimization Algorithm LFOPC

Adjustable Parameters

- Initial penalty value
- Maximum penalty value
- Gradient of the Lagrangian function (tolerance)
- Convergence tolerance on the step movement
- Maximum number of steps per phase



Not necessary to adjust

Feasibility Handling: LFOPC

Standard internal formulation :

Phase I :

Min. e (max. violation) ← e = *Slack variable*

subject to

$g_j(\mathbf{x}) \le e \quad ; \quad j = 1,\ldots,p$ ← Slack constraints (e >0 if feasibility not possible)

$g_j(\mathbf{x}) \le 0 \quad ; \quad j = p+1,\ldots,m$ ← Strict constraints (must be satisfied)

$e \ge 0$

Phase II (if $e = 0$, otherwise stop) :

Min. $f(\mathbf{x})$

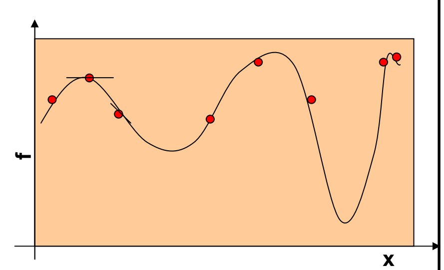
subject to

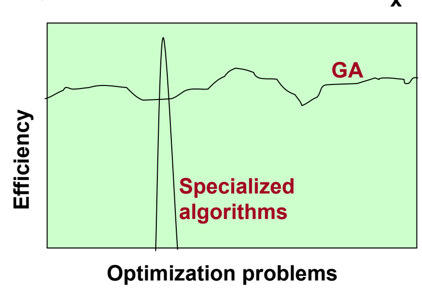
$g_j(\mathbf{x}) \le 0 \quad ; \quad j = 1,\ldots,m$

Note: e ***is automatic, internal***

Optimization Algorithm – Genetic Algorithm (GA)

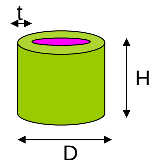
- GA was developed by John Holland in 1965
- Inspired by nature – "Nature does not waste resources", GA emulates Darwin's "Survival of the fittest" principle
- Specific GA features are
  - Population based stochastic optimizer
  - Robust global optimization method
  - Does not require gradient of function
  - Works with any function evaluator
  - Can be easily used on parallel architecture of machines
- Requires a large number of function evaluations





GA Terminology

Gene – each design variable (x)
Chromosome – group of design variables
Individual – each design point
Fitness – how good is the individual?
Population – group of individuals



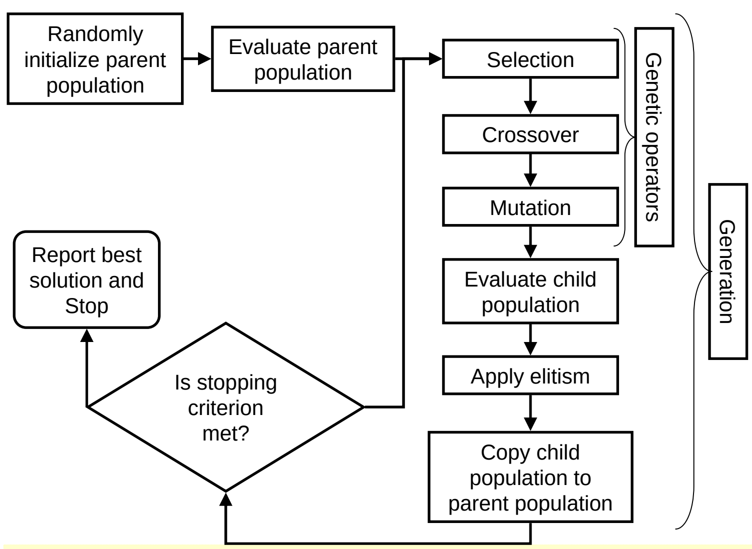
Individual
Chromosome [4.00,2.80,19.0] or [010011110010011] Fitness 25.8 (sum of variables)

Genetic operators – drive the search

- Selection – select the high fitness individuals – exploit the info.
- Crossover – parents create children - explore the design space
- Mutation – sudden random changes in chromosomes

Generation – each cycle of genetic operations

Flowchart of a Simple Genetic Algorithm



Simple GA caters to single objective optimization problems

Optimization Algorithm – Adaptive Simulated Annealing (ASA)

- This global stochastic optimization method simulates annealing process – starts at a high temperature and slow cooling would allow to achieve the lowest energy state
- Objective function is defined as the energy function E
- Points are accepted using a Metropolis criterion

$$A(E, E', T) = \min\{1, \exp(-(E' - E)/T)\}$$

- Temperature is periodically updated and search terminates when the temperature has fallen substantially
- Conventional SA updates the temperature as $T_p^{(k+1)} = T_p^0 / \log(k)$
- Ingber modified sampling to focus in the fast varying parameters such that faster cooling rates were feasible

$$T_p^{(k+1)} = T_p^k \exp(-ck^{1/n})$$

- Periodic re-annealing was also used to update the sensitivities

Discrete Optimization

- Discrete variables can have only distinct values, e.g. { 1.0, 2.0, 2.5, 4.5 }
- In most cases too expensive to evaluate all possible designs, e.g. 30 design variables with 5 possible values result in 10^{21} possible designs
- Discrete and continuous variables can be used together.
- User decides the sampling type to be continuous or discrete
- The optimization solution using LFOPC is a three stage procedure:
  - Find continuous optimum (using LFOPC)
  - Freeze continuous variables and do discrete optimization using Genetic Algorithm
  - Freeze new discrete variables and do continuous optimization
- Optimization using adaptive simulated annealing or genetic algorithm is a single stage procedure.
- Sequential strategy: Uses SRSM with special modifications to the region of interest for discrete variables

Normalization of Constraints

Required by user

$$g_1(x) - U_1 \leq e$$

$$g_2(x) - U_2 \leq e$$

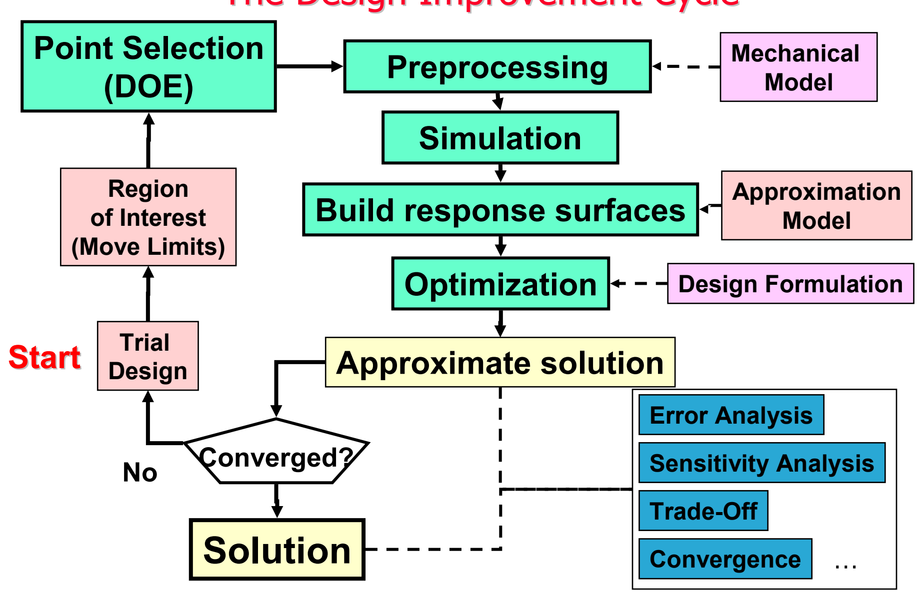
The user must normalize the constraints so that e is non-dimensional:

$$\frac{g_1 - U_1}{U_1} \leq e \Rightarrow \frac{g_1}{U_1} - 1 \leq e$$

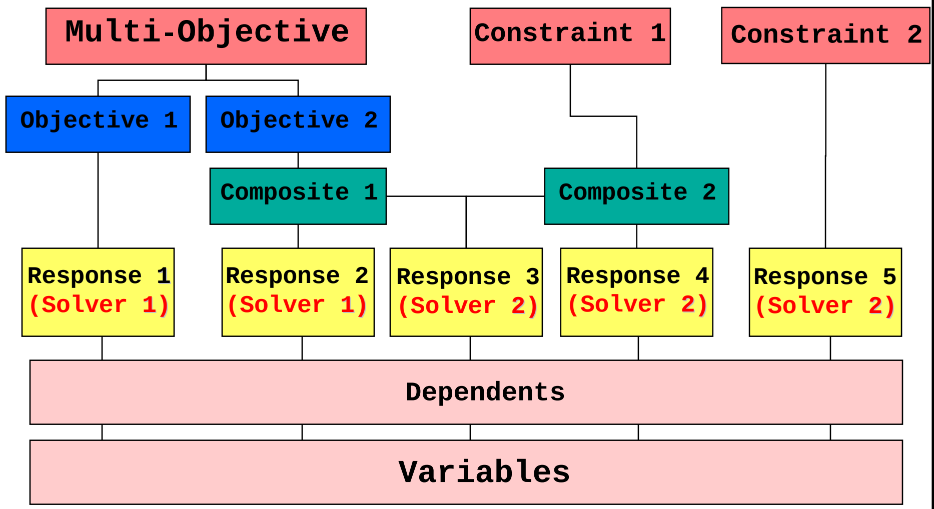
$$\frac{g_2 - U_2}{U_2} \leq e \Rightarrow \frac{g_2}{U_2} - 1 \leq e$$

by scaling individual constraints

The Design Improvement Cycle



Defining the Design Problem



Design Formulations

Standard composite functions

Mean Squared Error (MSE)

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

Weighted

$$\mathcal{F} = \sum_{j=1}^{m} W_j \frac{f_j(x)}{\sigma_j} + \sum_{i=1}^{n} w_i \frac{x_i}{\chi_i}$$

Most Feasible Design
Theory

Standard internal formulation :

Phase I :

Min. e (max. violation)

subject to

$g_j(\mathbf{x}) \le e \quad ; \quad j = 1,\dots,p$ ← ***Slack constraints (e>0 if feasibility not possible)***

$g_j(\mathbf{x}) \le 0 \quad ; \quad j = p+1,\dots,m$ ← ***Strict constraints (must be satisfied)***

$e \ge 0$

Phase II (if $e = 0$, otherwise stop) :

Min. $f(\mathbf{x})$

subject to

$g_j(\mathbf{x}) \le 0 \quad ; \quad j = 1,\dots,m$

***e* = <u>Slack</u> variable**

Note: e is automatic, internal

Most Feasible Design
Applications

- Minimize the maximum of various responses
- Targeted formulation (System identification)

Most Feasible Design
Example: Min-Max

Minimize the maximum knee force
subject to
constraints on the knee displacements

Min. e

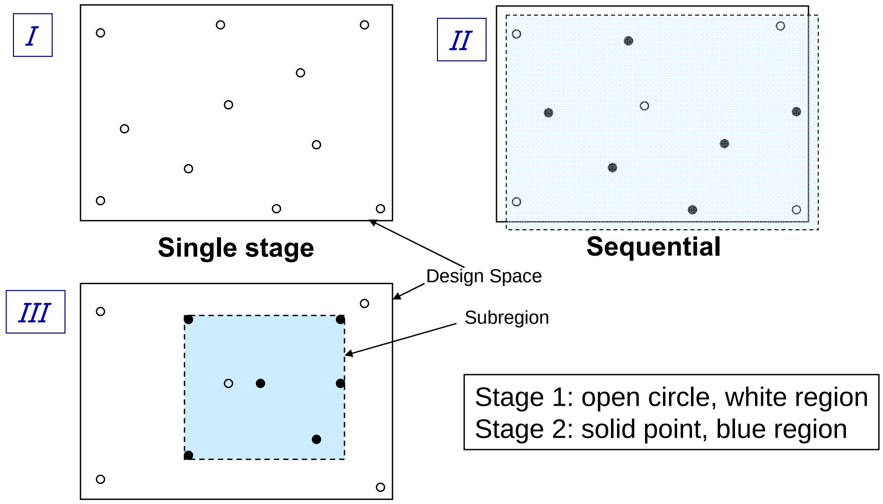
subject to

$F(\mathbf{x})_1 \le e$	Knee force # 1	Slack
$F(\mathbf{x})_2 \le e$	Knee force # 2	Slack
$d_1(\mathbf{x}) - D_1 \le 0$	Knee displacement #1	Strict
$d_2(\mathbf{x}) - D_2 \le 0$	Knee displacement #2	Strict

OPTIMIZATION STRATEGIES

Optimization Strategies

Space-filling point selection



Sequential with domain reduction

Optimization Strategies

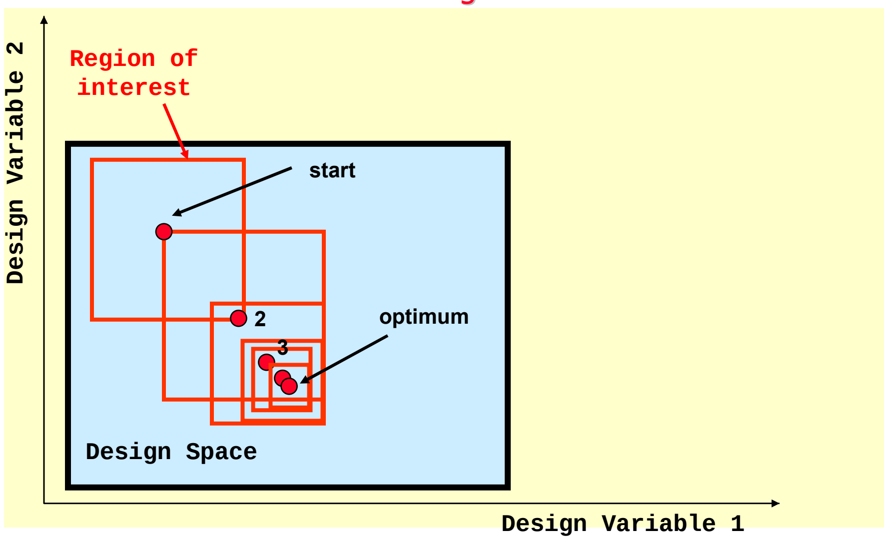
- **Single stage**
  - All the points are determined in one stage, using Space Filling
  - Highly suitable to create a global surrogate model
  - Choose a large number of Space Filling points to use NN or RBF
- **Sequential**
  - Choose a small number of points for each iteration
  - Add Space Filling points in each iteration
  - Highly suitable to create a surrogate model, e.g. NN or RBF
  - Accuracy is similar to single stage
  - More flexible than single stage. Can stop depending on accuracy

Optimization Strategies

- **Sequential with domain reduction**
  - Domain reduction in each iteration: all points within a subregion
  - Two types:
    - Sequential Adaptive Metamodeling (SAM):
      - Use RBF or NN
      - Global strategy: Points belonging to previous iterations are included
      - Same as Sequential but with domain reduction.
      - Moderately good for constructing global approximations
    - Sequential Response Surface Method (SRSM)
      - The original LS-OPT strategy using Polynomials with *D*-Optimality
      - Points belonging to previous iterations are ignored
      - Uses polynomials (typically linear) with *D*-Optimality
      - No global approximation available, so cannot construct Pareto optimal front

Reference: Stander, N. and Goel, T. Metamodel sensitivity to sampling strategies: a crashworthiness design study. *Proceedings of the 10th International LS-DYNA User's Conference, Dearborn, MI. June 9-10, 2008*

Sequential Response Surface Method: SRSM
Convergence

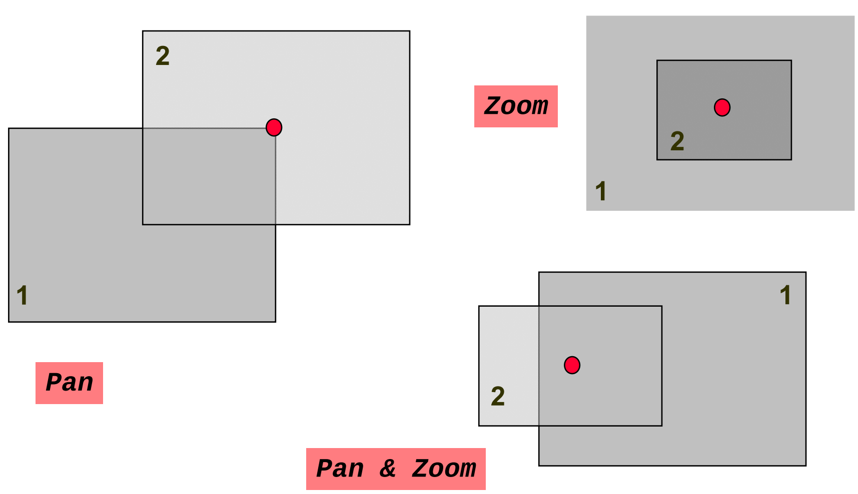


Sequential Response Surface Method
Parameters

- Sequential approximations are used to solve the optimization subproblem
- Depending upon the approximate optimum, the region of interest can either 'pan' (shift) or 'zoom'

Item	Parameter	Default value
`objective`	Tolerance on objective function accuracy ϵ_f	0.01
`design`	Tolerance on design accuracy ϵ_x	0.01
`psi`	γ_{pan}	1.0
`gamma`	γ_{osc}	0.6
`eta`	Zoom parameter η	0.6
`rangelimit`	Minimum range	0.0

Sequential Response Surface Method
Subregion reduction scheme



Sequential Response Surface Method

Explanation of parameters

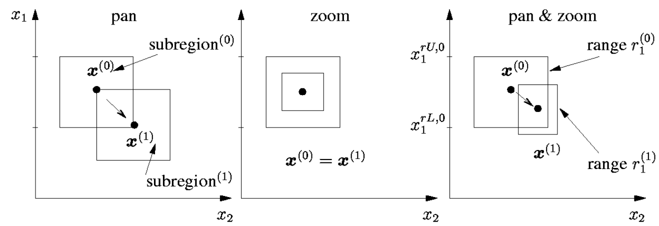


Figure 1: Successive Response Surface Methodology

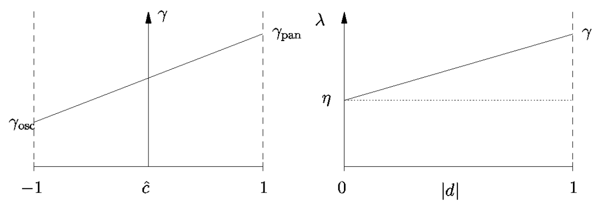


Figure 2: Oscillation and proximity criteria

Sequential Response Surface Method

Theory

The move limits are determined as:

$$x_i^{rL,0} = x_i^{(0)} - 0.5\, r_i^{(0)} \qquad \text{and} \qquad x_i^{rU,0} = x_i^{(0)} + 0.5\, r_i^{(0)} \qquad i = 1, \ldots, n \tag{1}$$

n is the number of design variables.

Oscillation: An *oscillation indicator* c is determined in iteration k as

$$c_i^{(k)} = d_i^{(k)} d_i^{(k-1)} \tag{2}$$

where

$$d_i^{(k)} = 2\Delta x_i^{(k)} / r_i^{(k)}; \quad \Delta x_i^{(k)} = x_i^{(k)} - x_i^{(k-1)}; \quad d_i^{(k)} \in [-1; 1]. \tag{3}$$

The oscillation indicator is converted to $\hat{c}$ where

$$\hat{c} = \sqrt{|c|} \quad \text{sign}\ (c). \tag{4}$$

The contraction parameter γ is calculated as

$$\gamma = \frac{\gamma_{\text{pan}}(1 + \hat{c}) + \gamma_{\text{osc}}(1 - \hat{c})}{2}. \tag{5}$$

Accuracy: The range $r_i^{(k+1)}$ for the new subregion in the $(k+1)$-th iteration is determined by:

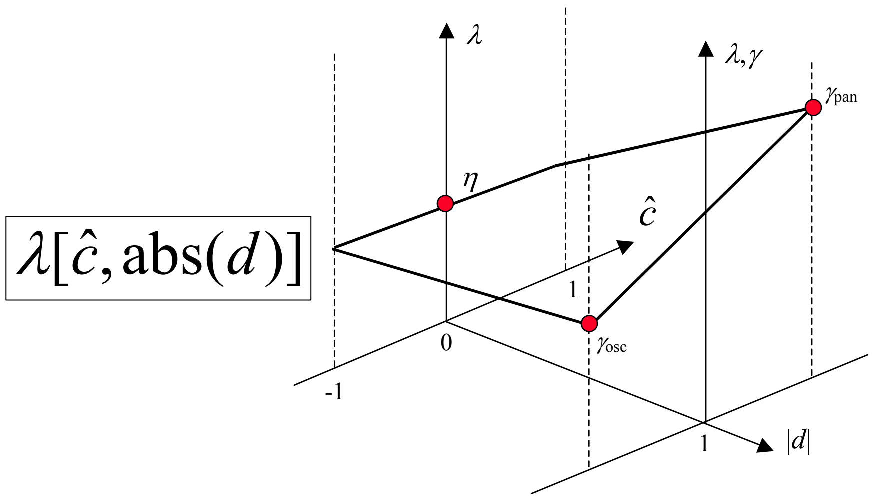
$$r_i^{(k+1)} = \lambda_i\, r_i^{(k)} \qquad i = 1, \ldots, n \qquad k = 0, \ldots, niter \tag{6}$$

where λ_i represents the *contraction rate* for each design variable.

$$\lambda_i = \eta + |d_i|(\gamma - \eta) \tag{7}$$

for each variable.

Sequential Response Surface Method
Contraction rate



113

Sequential Response Surface Method
Convergence criteria

- **Using error norm of design variables:**

$$\varepsilon_x = \frac{||\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}||}{||\boldsymbol{x}^{(k)}||}$$

- **Using objective function**

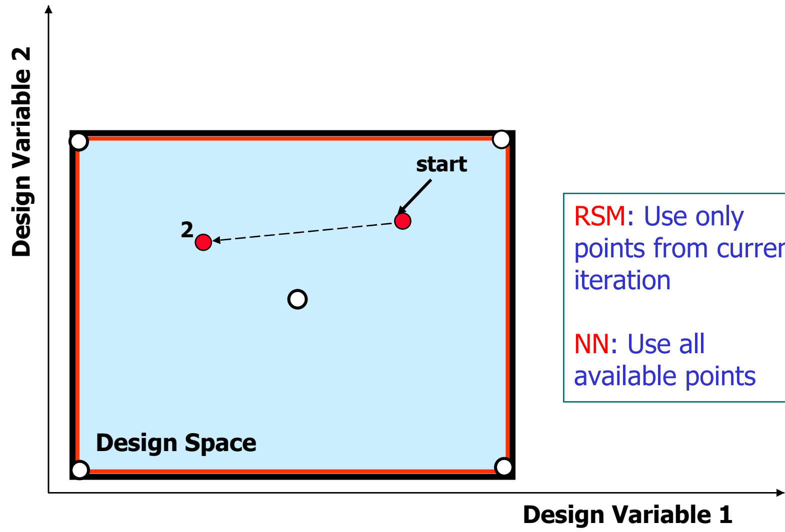
$$\varepsilon = \frac{||f^{(k)} - f^{(k-1)}||}{||f^{(k)}||}$$

- **Can choose <u>whichever comes first</u> or <u>both</u>**

114

Sequential Approximation Scheme

Iteration 1

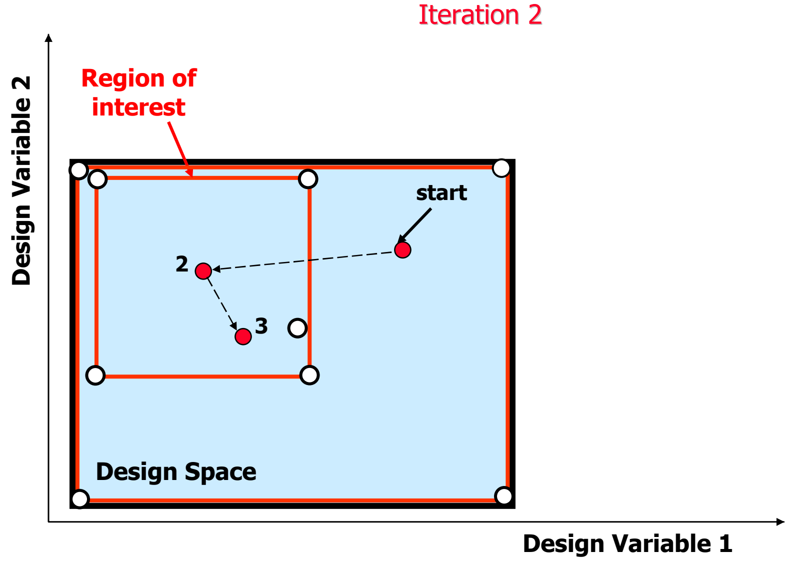


RSM: Use only points from current iteration

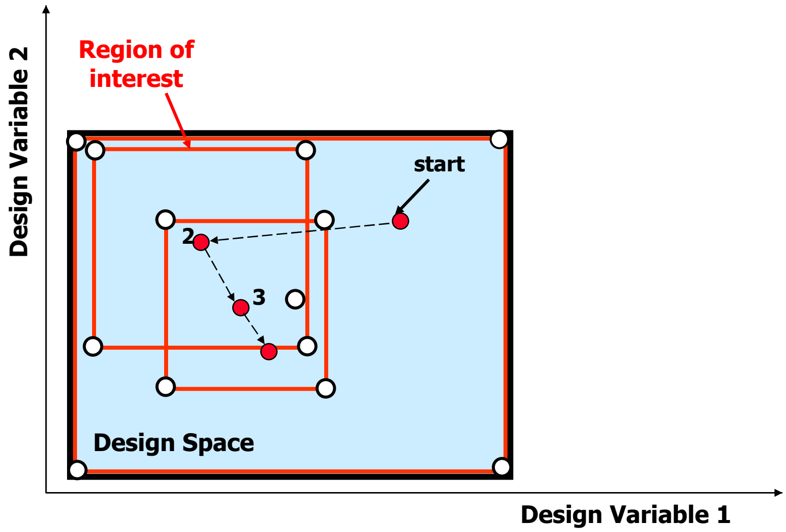
NN: Use all available points

Sequential Approximation Scheme

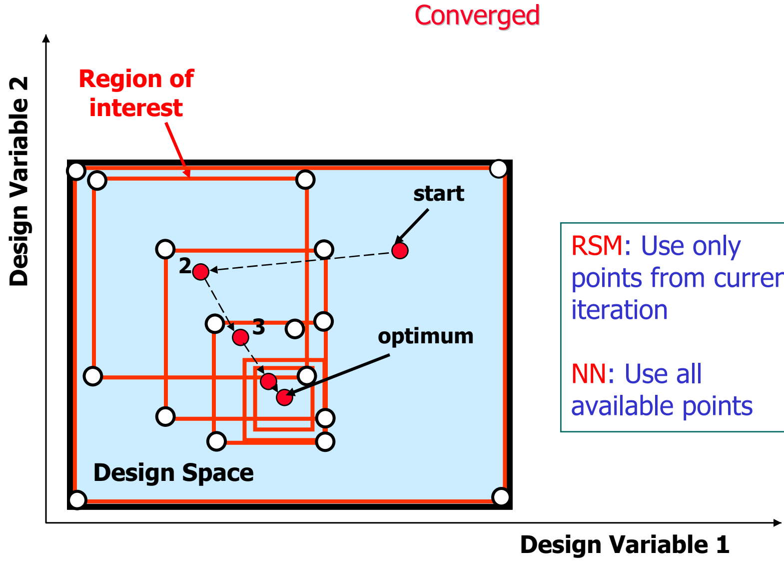
Iteration 2



Sequential Approximation Scheme
Iteration 3



Sequential Approximation Scheme
Converged



RSM: Use only points from current iteration

NN: Use all available points

Metamodels: Example

<u>Crash model</u>

30 000 elements

Intrusion = 552mm

Stage1Pulse = 14.34g

Stage2Pulse = 17.57g

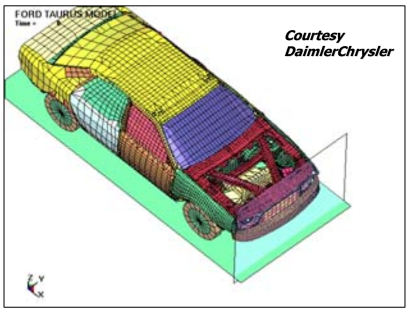
Stage3Pulse = 20.76g

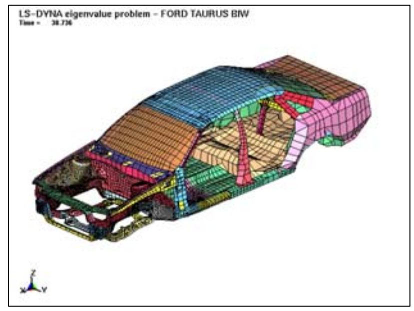
<u>BIW model</u>

18 000 elements

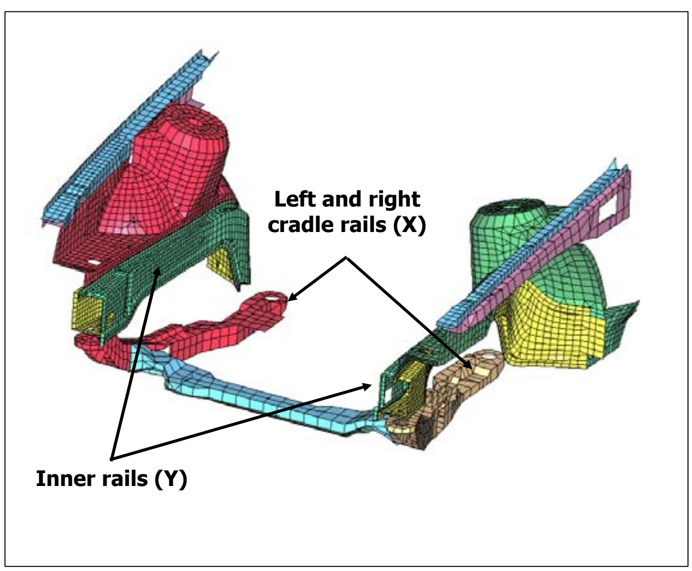
Torsional mode 1

Frequency = 38.7Hz

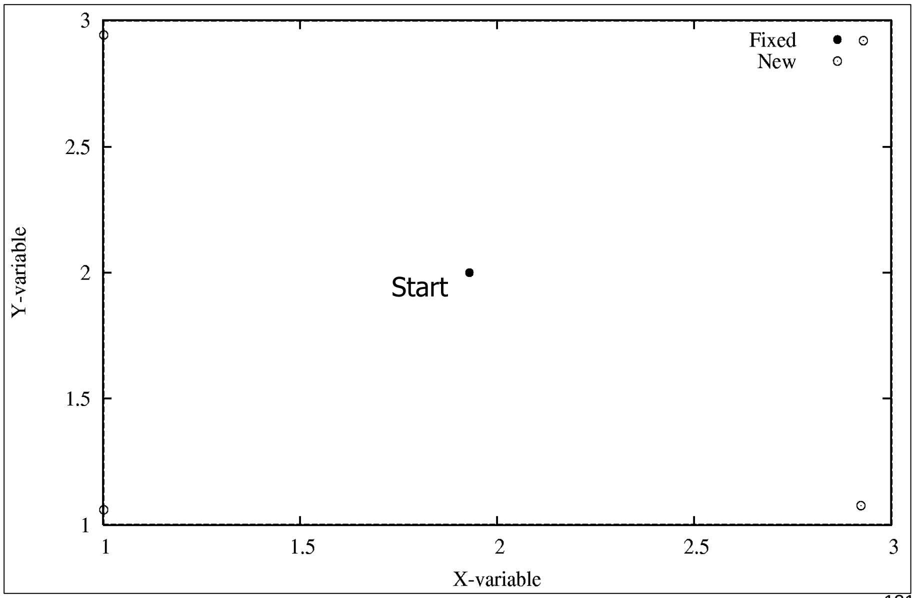




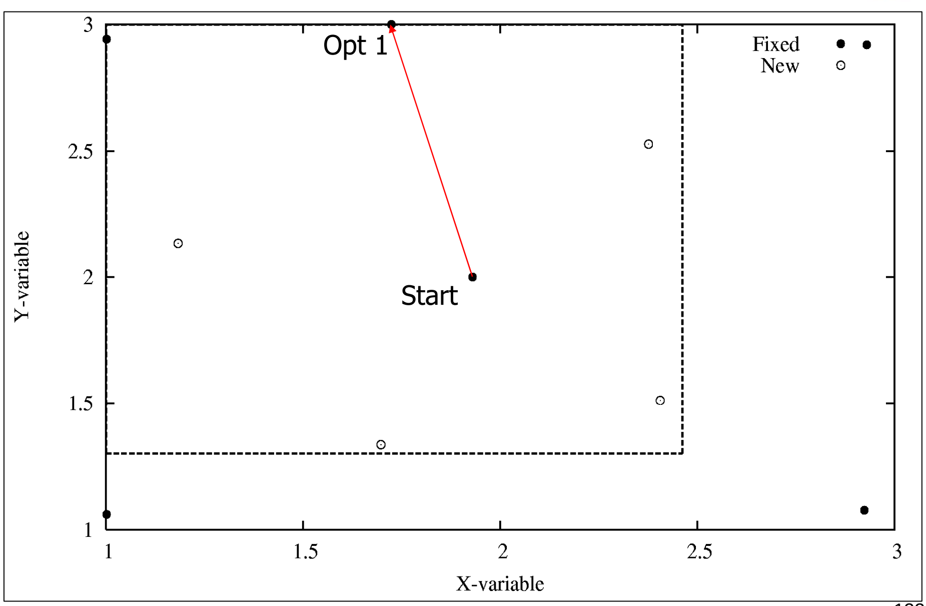
Two Design Variables (Thickness)



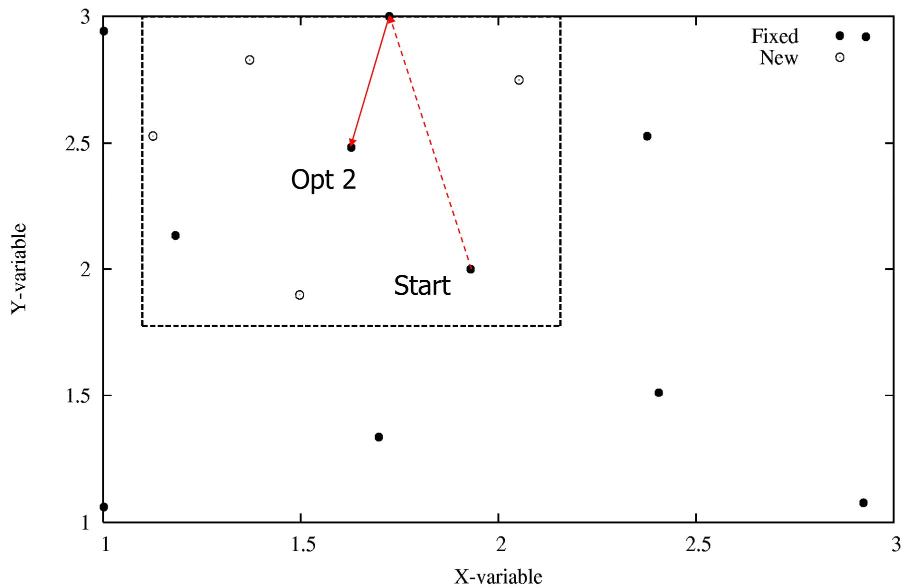
Sampling: Space Filling Method
Iteration 1



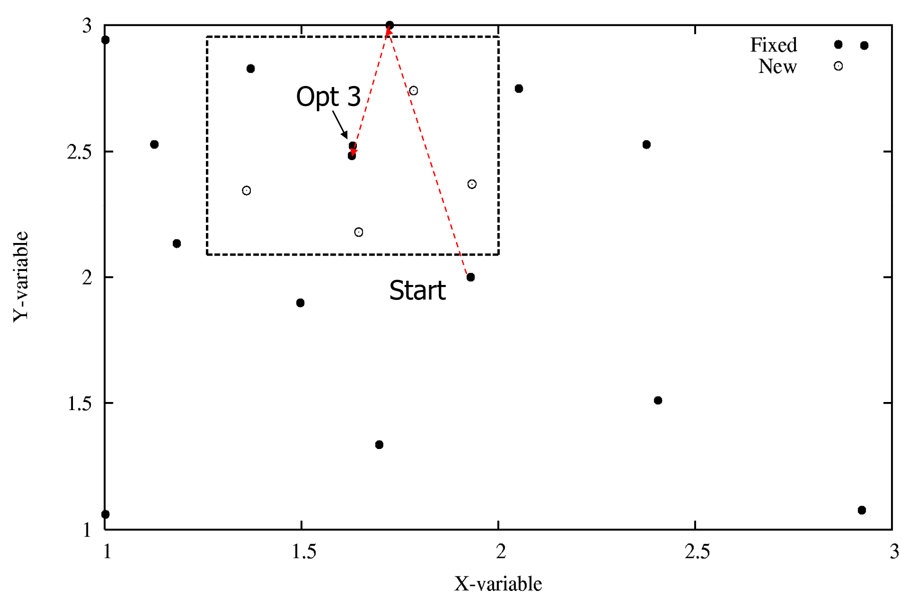
Sampling: Space Filling Method
Iteration 2



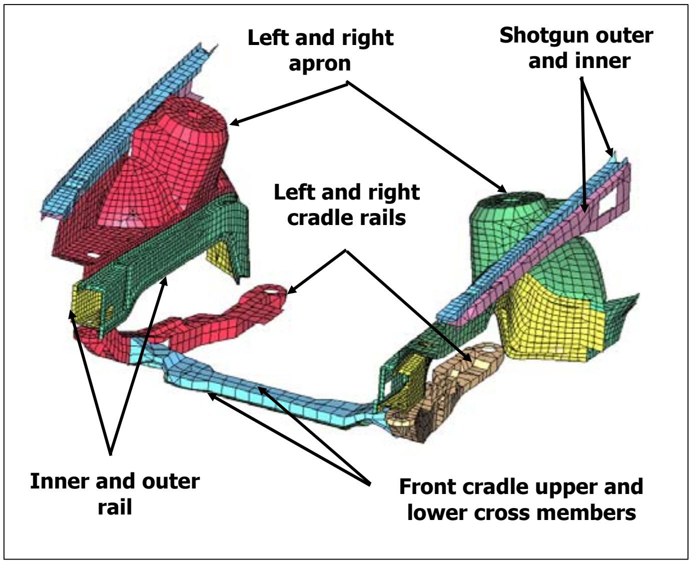
Sampling: Space Filling Method
Iteration 3



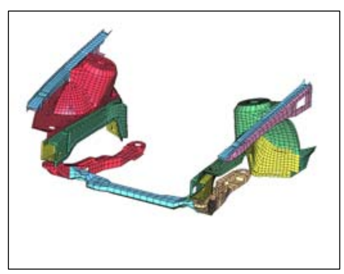
Sampling: Space Filling Method
Iteration 4



Optimization
Design variables (Thickness)



Design Formulation



Design Objective:

Minimize (Mass of components)

Design Constraints:

Intrusion < 552.38mm

Stage1Pulse > 14.58g

Stage2Pulse > 17.47g

Stage3Pulse > 20.59g

41.38Hz < Torsional mode 1 frequency < 42.38Hz

Crashworthiness design variables: 4 screened out of 7 total

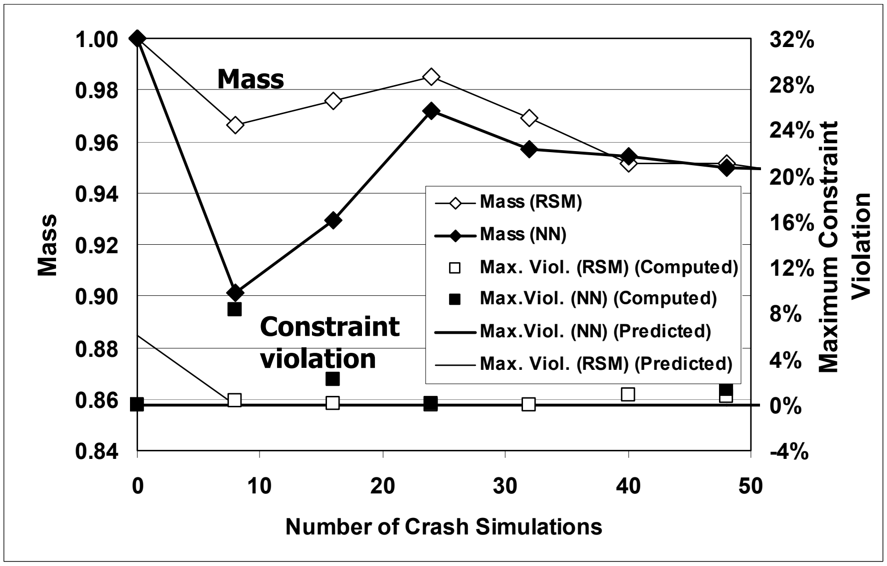
Rails (inner and outer); Aprons; Cradle rails

NVH design variables: 7 (all)

Crashworthiness responses: Intrusion, Stage Pulses

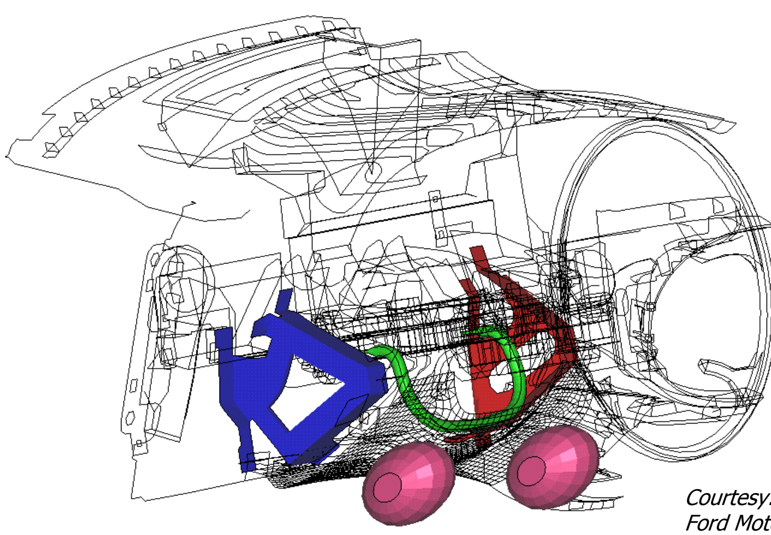
NVH responses: Mass, Frequency, Mode number (for tracking internally)

Metamodel Comparison: Optimization



127

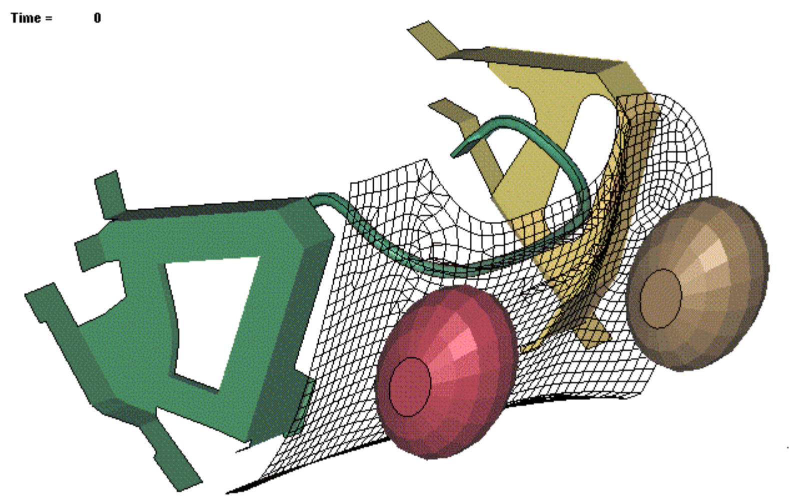
Instrument Panel with Knee Bolster System



Courtesy:
Ford Motor Company

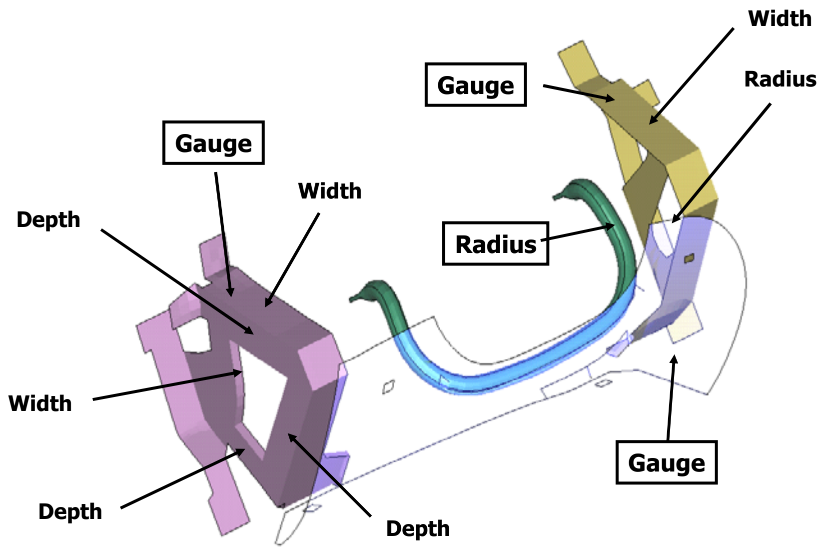
128

Instrument Panel: LS-DYNA Simulation



Instrument Panel: Design Variables

4 screened out of 11 total



Instrument Panel: Design Formulation

Design Objective:

min (max (Knee_F_L, Knee_F_R))

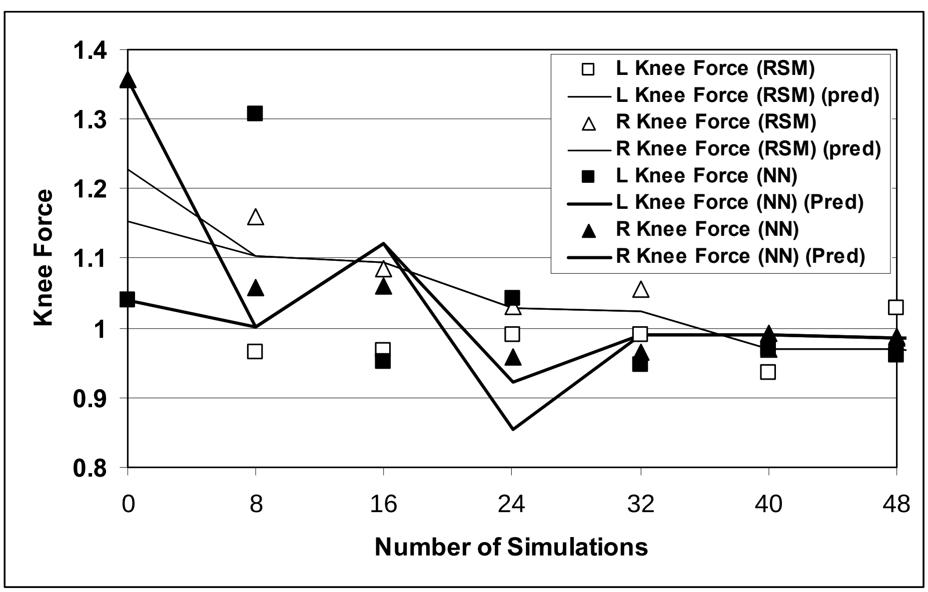
Design Constraints:

Left Knee intrusion	<	115mm
Right Knee intrusion	<	115mm
Yoke intrusion	<	85mm

Design variables

Reduced from 11 to 4 (ANOVA)

Instrument Panel: Optimization



MULTI-OBJECTIVE OPTIMIZATION

Multi-objective Optimization

- Most engineering problems deal with multiple objectives e.g., cost, weight, safety, efficiency etc.
- Often conflicting requirements e.g., weight vs. efficiency
- Mathematical formulation –

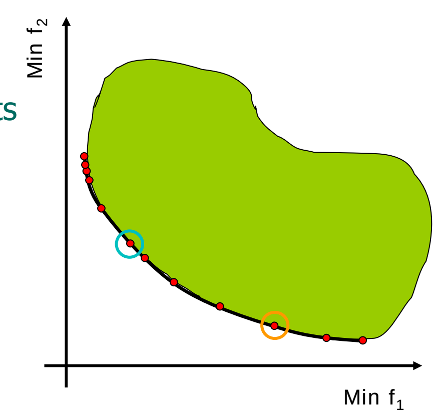
  Minimize $f_i(x)$ i = 1, M,

  $x = \{x_j: j = 1, N\}$

  Subject to:

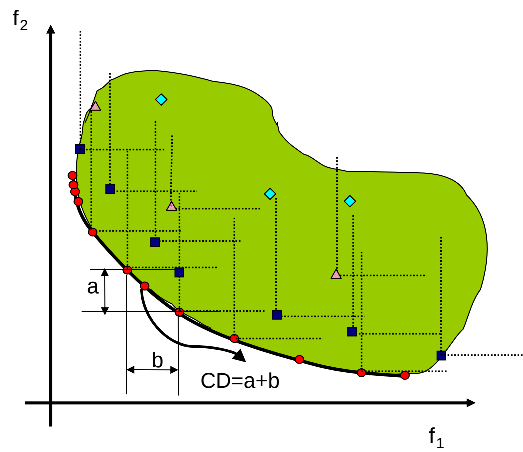
  $C_k(x) \leq 0$, k = 1, P

  $h_l(x) = 0$, l = 1, Q
- No single optimal solution



Concept of Pareto Optimality

- Non-dominated solutions
- Pareto optimal solutions
- Pareto optimal front
- Salient features
  - continuous or discontinuous
  - convex or non-convex



135

Different Methods in LS-Opt

Weighted sum strategy

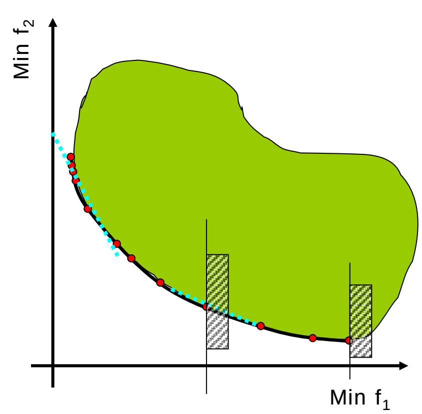
- convert multiple objectives into a single objective using weights

ε-constraint strategy

- all but one objectives are treated as constraints and optimize for the left-out objective

Multi-objective genetic algorithm

- all objectives are simultaneously optimized



136

Transition: Single to Multi-objective GA

GA is well-suited to solve multi-objective optimization problem

- Results in a set of potential Pareto optimal solutions
- Extra computational effort in optimization can be justified by the outcome – Pareto optimal front

Issues that need to be addressed are

- How to compare individuals? – easy for simple GA but not so intuitive for multi-objective GA – use ranking to identify 'fitter' individuals
- Need to preserve diversity in the solutions – complete Pareto optimal front is desired
- Convergence to the global Pareto optimal front

Popular MOGA are NSGA, SPEA, PAES, NSGA-II.

How to Use Multi-Objective GA?

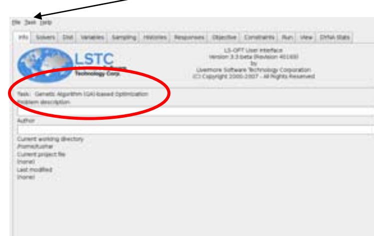
Direct simulation based GA

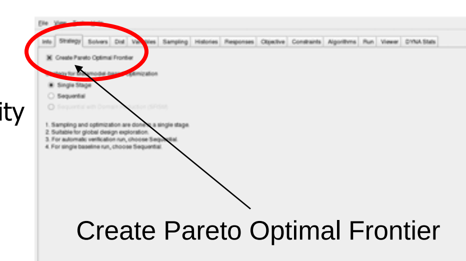
- optimize without creating meta-models
- use simulations to evaluate designs – high computational cost!
- accurate results

Meta-model based GA

- use metamodels to evaluate functions
- accuracy depends on the quality of metamodels
- computationally inexpensive

Select task





Validation of GA Using Benchmark Examples

Unconstrained test problems

$$\underset{\vec{X}}{Minimize} \quad f_1(\vec{X}) = x_1; \quad f_2(\vec{X}) = g(\vec{X})h\left(g(\vec{X}), f_1(\vec{X})\right)$$

$$\textbf{ZDT2}: g(\vec{X}) = 1 + 9(N-1)^{-1}\sum_{i=2}^{N} x_i;$$

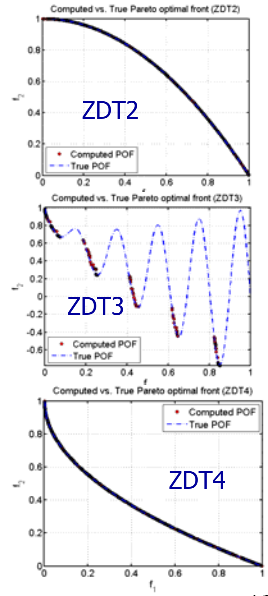
$$h(\vec{X}) = 1 - (f_1/g)^2; x_i \in [0,1].$$

$$\textbf{ZDT3}: g(\vec{X}) = 1 + 9(N-1)^{-1}\sum_{i=2}^{N} x_i;$$

$$h(\vec{X}) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1); x_i \in [0,1].$$

$$\textbf{ZDT4}: g(\vec{X}) = 1 + 10(N-1) + \sum_{i=2}^{N}(x_i^2 - 10\cos(4\pi x_i));$$

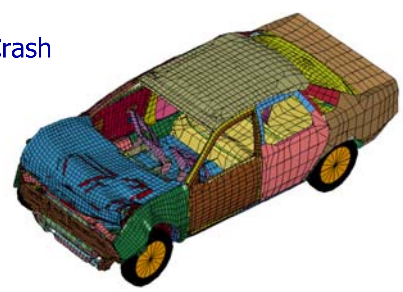
$$h(\vec{X}) = 1 - \sqrt{f_1/g}; x_1 \in [0,1]; x_{2,3,\ldots N} \in [-5,5].$$



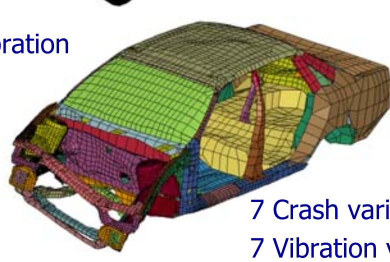
Goel T, Stander N, Multi-Objective Optimization Using LSOPT, 6th German LS-Dyna Forum, Oct 11-12, 2007, Frankenthal, Germany.

Direct Multi-Objective Optimization: Example 1

Crash



Vibration



Min. (Mass, Intrusion)

Subject to:

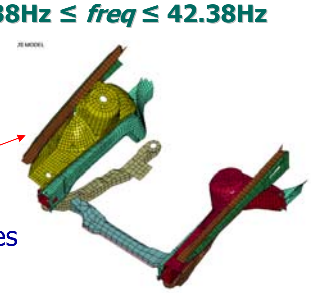
Intrusion ≤ 551mm

Stage 1 pulse > 14.5*g*

Stage 2 pulse > 17.6*g*

Stage 3 pulse > 20.7*g*

41.38Hz ≤ *freq* ≤ 42.38Hz



7 Crash variables

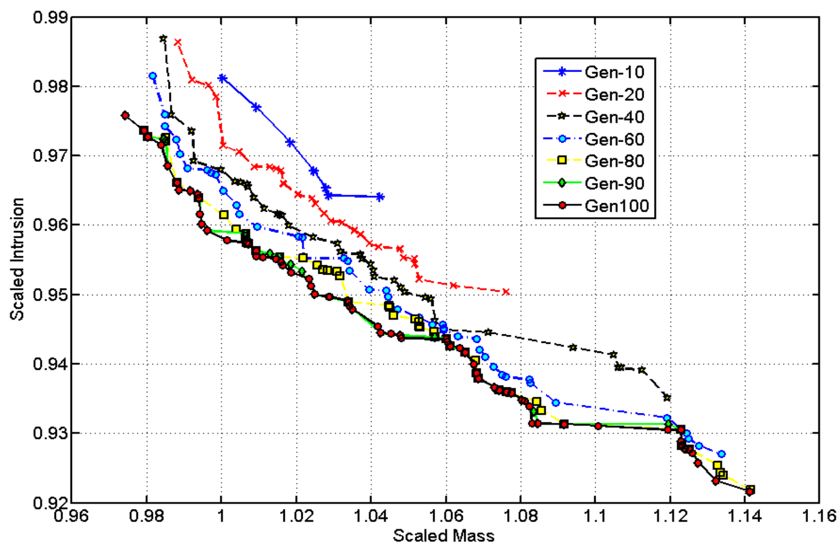
7 Vibration variables

(2 discrete)

Direct Multi-Objective Optimization: Simulation Statistics

- **IBM x3455** cluster**. 40 nodes (160 cores)**
- Queuing **through Loadleveler**
- **Crash simulation time 3,400-3,900 sec**
- **Modal analysis time 40 sec**
- Population**: 80**
- Generations**: 100**
- **Total of** 8,000 **crash runs +** 8,000 **modal analysis runs (run to** convergence**!)**

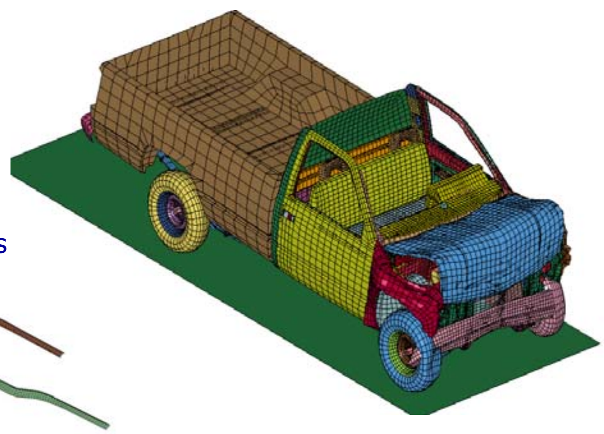
Pareto Optimal Front History



Li G, Goel T, Stander N, Assessing the convergence properties of NSGA-II for direct crashworthiness optimization, 10th International LS-Dyna Conference, Jun 8-10, 2008, Detroit, MI.

Direct Multi-Objective Optimization: Example 2

Thickness design variables



Lin Y-Y, Goel T, Stander N, Direct Multi-Objective Optimization Through LS-OPT Using a Small Number of Crashworthiness Simulations, 10th International LS-Dyna Conference, Jun 8-10, 2008, Detroit, MI.

Design Criteria

Minimize
- Mass
- Acceleration

Maximize
- Intrusion
- Time to zero velocity

9 thickness variables of main crash members

Intrusion	<	721
Stage 1 pulse	<	7.5g
Stage 2 pulse	<	20.2g
Stage 3 pulse	<	24.5g

Simulation Statistics

640-core HP XC cluster (Intel Xeon 5365 80 nodes of 2 quad-core)

Queuing through LSF

Elapsed time per generation ~ 6 hours

Population: 20

Generations: 50

Total of 1,000 crash runs

Results

Minimize

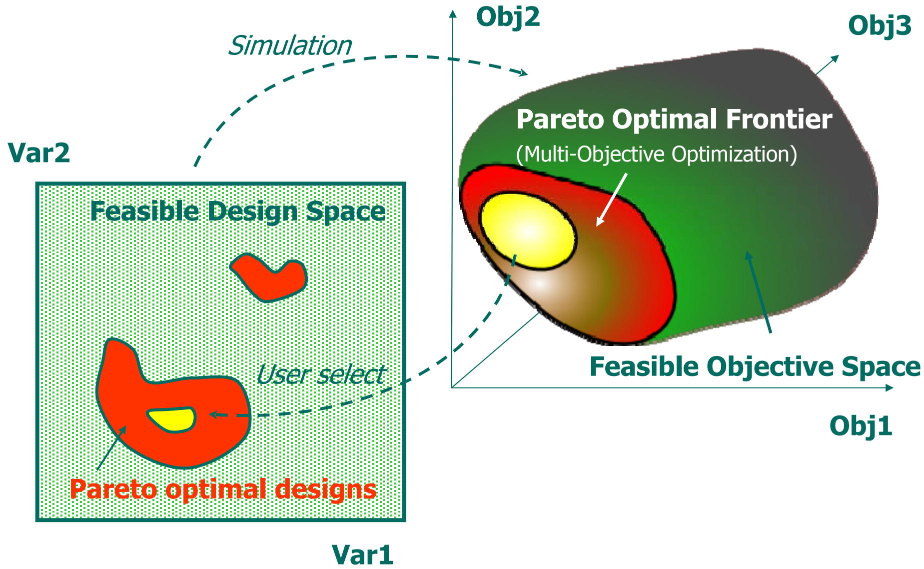
- Mass 0.3% ↑
- Acceleration 45% ↓

Maximize

- Intrusion 1% ↑
- Time to zero velocity 10% ↑

Intrusion	**< 721**	**(711 → 719)**
Stage 1 pulse	**< 7.5g**	**(7.9 → 6.9g)**
Stage 2 pulse	**< 20.2g**	**(21.1 → 20.1g)**
Stage 3 pulse	**< 24.5g**	**(25.2 → 23.6g)**

Mapping: Design Space ↔ Objective Space



Visualization of Pareto Optimal Frontier
Hyper Radial Visualization

- Hyper Radial Visualization (HRV) maps any number of objectives to 2D
- Objectives are placed in X and Y groups
- Grouping does not matter as points on the same contour have the same indifference "value" → "Best" point (closest to Utopian point) is always the same
- Objectives can be weighted by moving sliders to adjust W_i

Hyper Radial Visualization

- Conversion of the *multi-objective* optimization problem to a *two-objective* optimization problem using the objective:

$$\left(\sqrt{\sum_{i=1}^{N_x} W_i \tilde{F}_i^2}, \sqrt{\sum_{i=N_x+1}^{n} W_i \tilde{F}_i^2}\right)$$

subject to

$$\sum_{i=1}^{n} W_i = 1 \quad \text{and} \quad W_i > 0$$

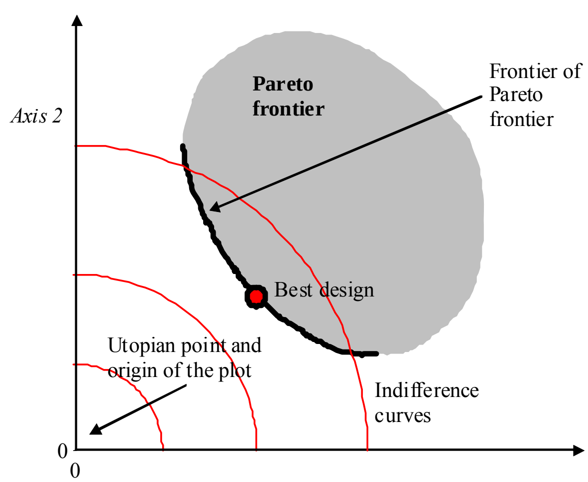
and

$$\tilde{F}_i = \frac{F_i - F_{i\ \min}}{F_{i\ \max} - F_{i\ \min}} \qquad i = 1,\ldots,n \ \text{ where } \tilde{F}_i \in [0,1]$$

- *2D Mapping:* The two additive components represent the objectives assigned to the two plot axes (see figure)

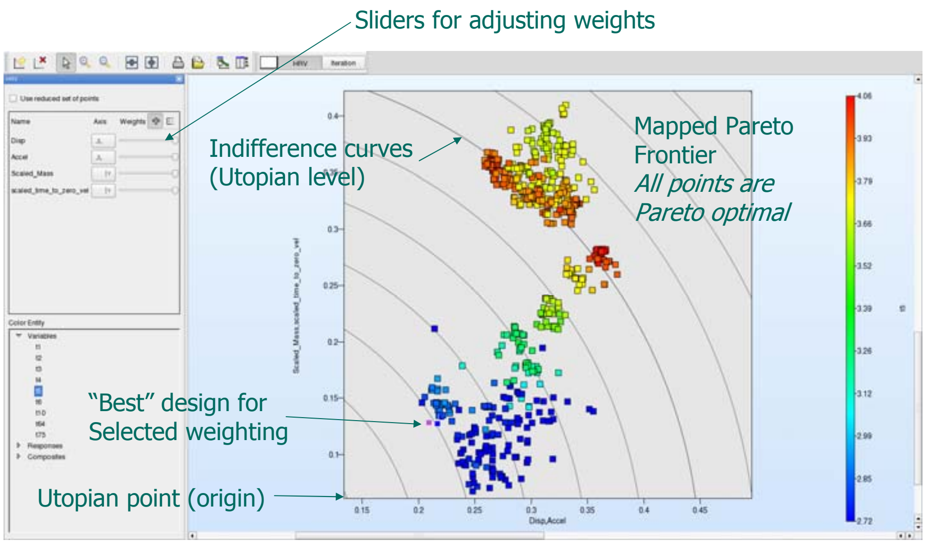
149

Hyper Radial Visualization



150

Hyper Radial Visualization



Visualization of the Pareto Optimal Frontier
Other methods

- **4-D Scatter plot**: 3D + color
- **Parallel coordinate plot**
  - Handles any number of dimensions
- **Self-Organizing Maps**

MULTI-DISCIPLINARY OPTIMIZATION

Multidisciplinary Design Optimization

- Specify multiple solvers to subdivide optimization process, e.g. multiple cases (Crash: Frontal, Offset, Side, Rollover) or disciplines (e.g. crash, vibration)
- Each solver has unique solvers, input files, job information, preprocessor, histories and responses
- Variables can be exclusive or can be shared with other solvers. Variable screening can be used to remove variables from disciplines prior to optimization
- GUI: Variables: Participating solvers can be selected for each variable
- Optimization Solution: All variables are updated after each iteration

Multidisciplinary Design Optimization

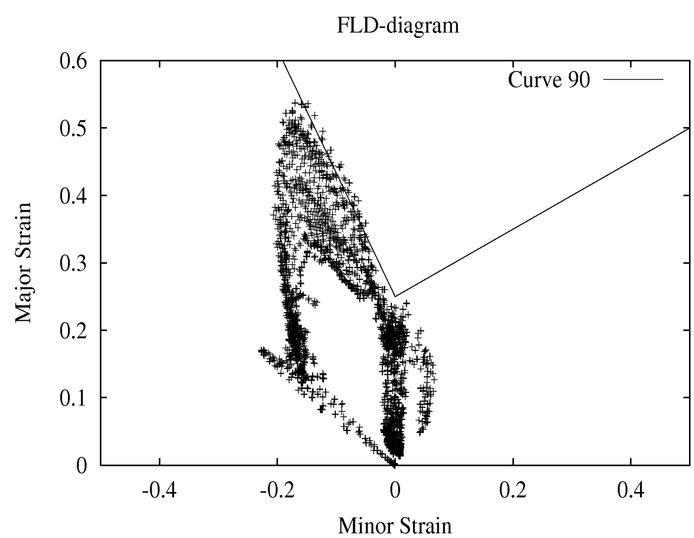
- Dependents and Composites are always global
- See MDO example involving crashworthiness and vibration properties elsewhere

APPLICATIONS

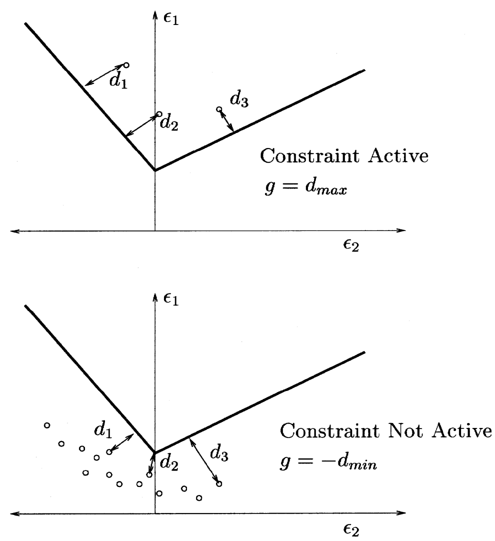
Metal Forming Criteria
Types

- Thickness, Thickness reduction
- Forming Limit criterion based on in-plane principal strains
- Average principal stress

Metal Forming Criteria
FLD criterion

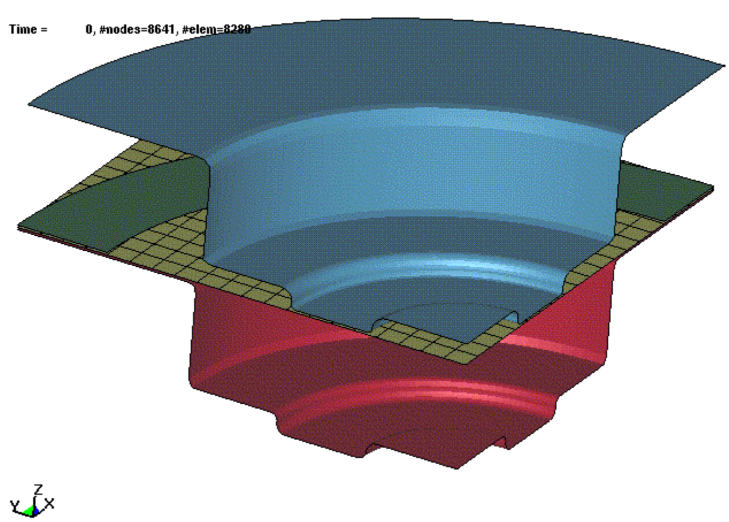


Metal Forming Criteria
FLD criterion



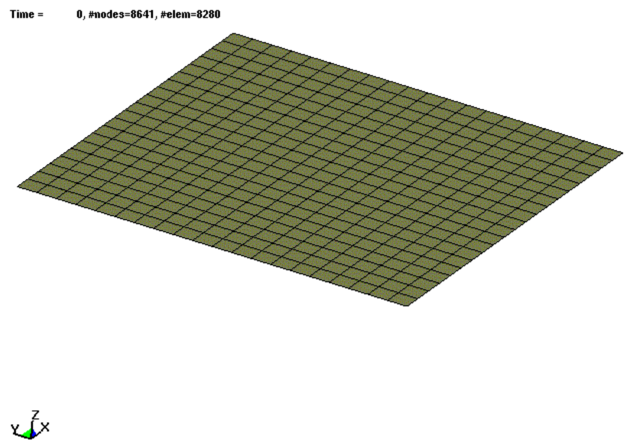
159

Metal Forming Criteria
Example: Stamping With LS-DYNA

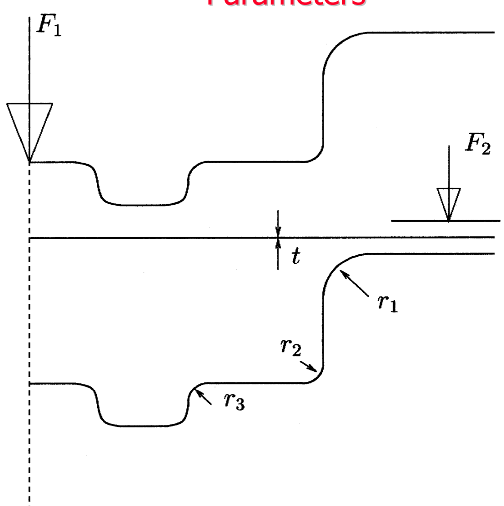


160

Metal Forming
Deformed blank



Metal Forming
Parameters



Metal Forming
Design criteria

Design Objective:

Minimize Maximum Radius

Design Constraints:

Maximum thinning (Δt) < 20%

FLD < 0

Radius design variables:

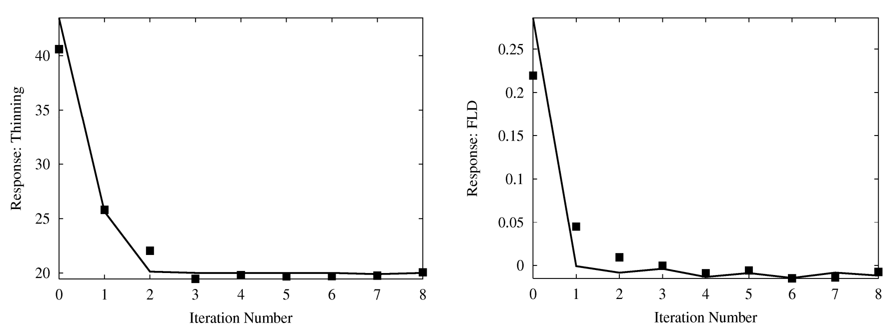
3 radii: r1, r2, r3 (see diagram)

FE model:

Adaptive meshing

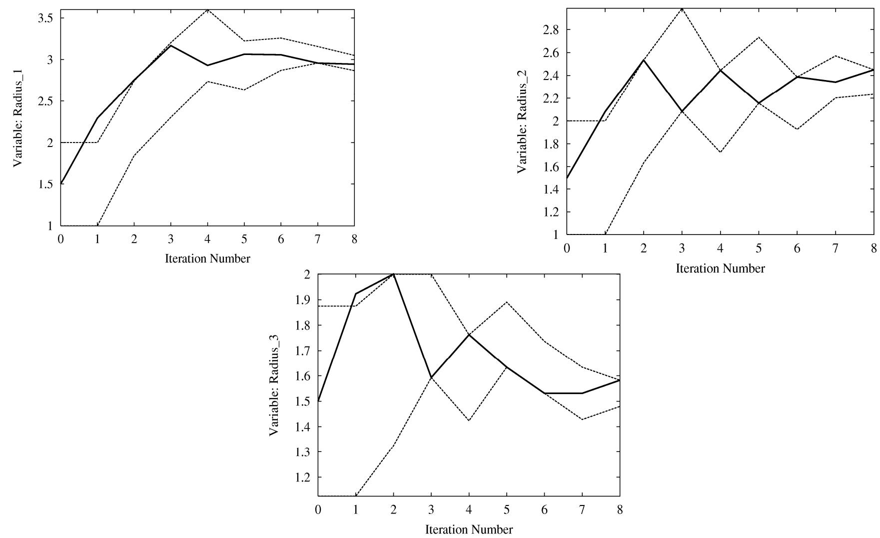
163

Metal Forming
Optimization history: Responses



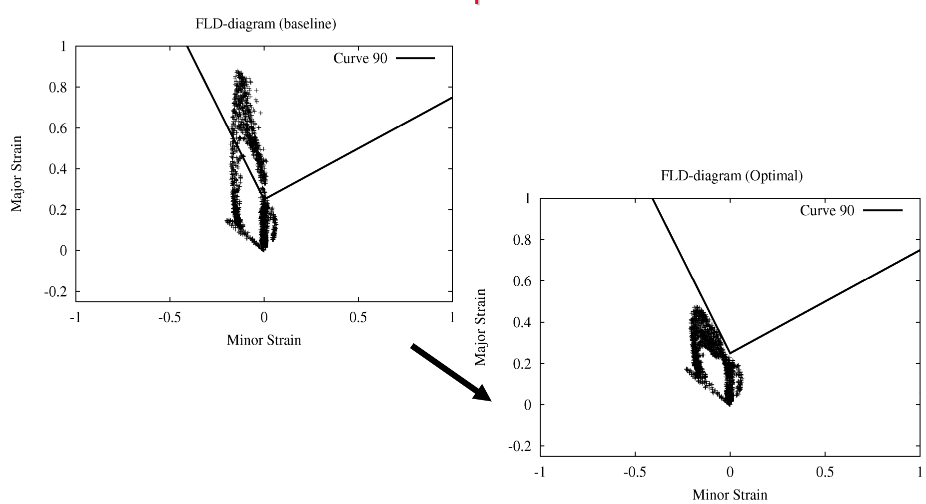
164

Metal Forming
Optimization history: Variables



165

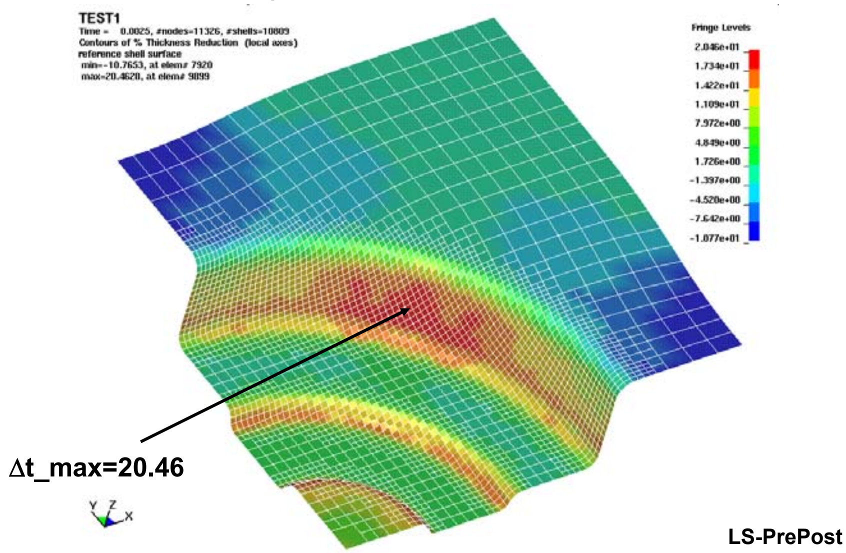
Metal Forming
FLD improvement



166

Metal Forming

Final product thickness distribution



167

Mode Tracking

- During NVH optimization necessary to track mode as mode switching can occur due to design changes
- Use eigenvalues and mass-orthogonalized eigenvectors of modal analysis
- Search for maximum scalar (dot) product between eigenvector of base mode and each solved mode:

$$\max_j \left\{ \left(M_0^{\frac{1}{2}} \phi_0 \right)^T \left(M_j^{\frac{1}{2}} \phi_j \right) \right\}$$

Reference mode (ϕ_0) — Compared mode (ϕ_j)

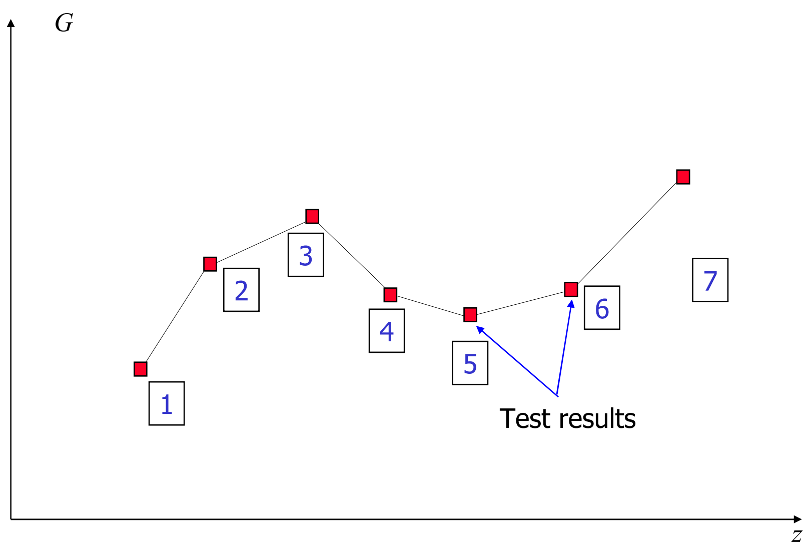
168

Parameter Identification

- Used for calibrating material or system properties
- Methodology uses optimization of the Mean Squared Error to minimize differences between test and computed results
- Response surfaces constructed at each point instead of for the total MSE
- MSE can be point-based or history-based
  - Point-based: The target value has to be specified for each point (selected as a "Composite" in Responses panel)
  - History-based: The target values can be specified in a history file and imported as a history. A single function computes the MSE

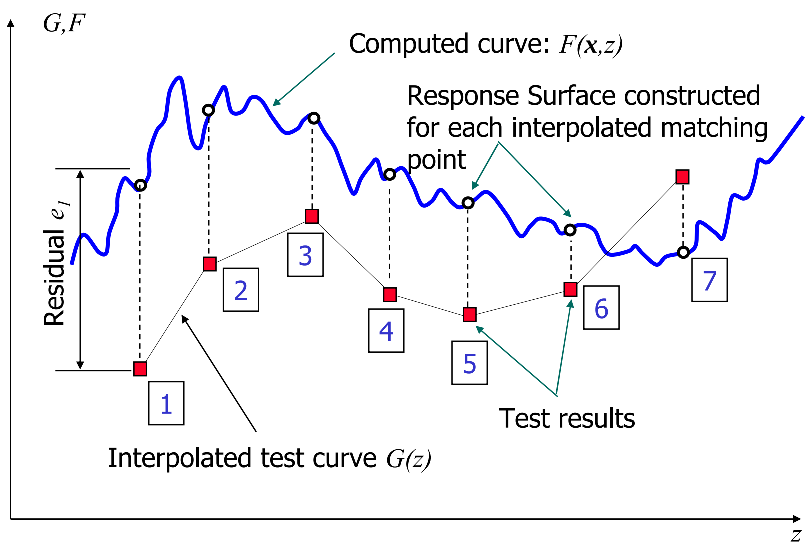
History-based Parameter Identification

Test points



History-based Parameter Identification

Test points + Computed curve



171

History-based Parameter Identification

Mean squared error

Weight (Importance of error)

Response Surface Value

Test Value

Residual

$$\frac{1}{P}\sum_{p=1}^{P} W_i \left(\frac{F_i(\mathbf{x}) - G_i}{s_i} \right)^2 = \frac{1}{P}\sum_{p=1}^{P} W_i \left(\frac{e_i(\mathbf{x})}{s_i} \right)^2$$

Variables (material or system constants)

Number of points

Residual Scale factor (Normalization of error)

172

Confidence Intervals
of Individual Parameters

Nonlinear regression model:

Measured results — History — time — Unknown parameters

$$G(t) = F(t, \mathbf{x}) + \varepsilon$$

Residual

Discrete nonlinear least squares problem

$$\min_{\mathbf{x}} \frac{1}{P} \sum_{p=1}^{P} (G_p - F_p(\mathbf{x}))^2$$

Number of points

Confidence Intervals
of Individual Parameters

The variance is estimated by

$$\hat{\sigma}^2 = \frac{\left\| \mathbf{G} - \mathbf{F}(\mathbf{x}^*) \right\|^2}{P - n}$$

Number of variables

The 100(1- α)% confidence interval for each variable is:

$$\left(\left[x_i : \left| x_i^* - x_i \right| \leq \sqrt{\hat{C}_{ii}} t_{P-n}^{\alpha/2} \} \right] \right)$$

where

$$\hat{\mathbf{C}} := \hat{\sigma}^2 \left(\nabla \mathbf{F}\left(x^*\right)^T \nabla \mathbf{F}\left(x^*\right) \right)^{-1}$$

and $t_{P-n}^{\alpha/2}$ is the Student *t*-distribution for α

History-based Parameter identification
Relevant commands

Get test data

```
History 'testcurvename' file "testfilename"
```

Construct crossplot

```
History 'curvename' {Crossplot (
    history_x_name, history_y_name,
    [numpoints, begin, end] )}
```

Dyna time-histories

Construct error norm of curve mismatch

```
Composite 'name' {MeanSqErr (
    'testcurvename', 'curvename',
    [numpoints, begin, end,
    weighting_type, scaling_type,
    weighting_value, scaling_value,
    weighting_curve, scaling_curve] )}
```

Curves

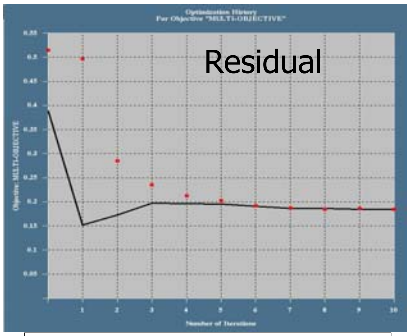
Material Identification: Concrete Material 159
11 parameters, 9 test types, 20 different test sets

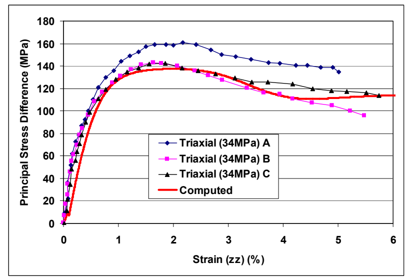
Par.	C00 UNC	T00 DP	PRS ISO-comp	UNX UNX	C07 TXC7	C14 TXC14	C20 TXC20	C34 TXC34	C69 TXC69
G	●	●	●	●					
K	●	●	●	●					
R				●	●	●	●	●	●
X_0			●						
W			●	●					
D_1			●						
D_2			●						
θ					●	●	●	●	●
λ					●	●	●	●	●
β					●	●	●	●	●
η					●	●	●	●	●

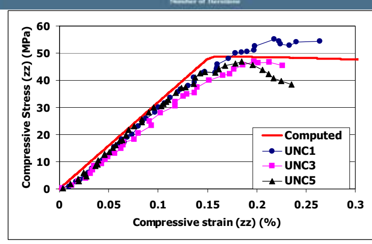
Multiple cases, shared variables

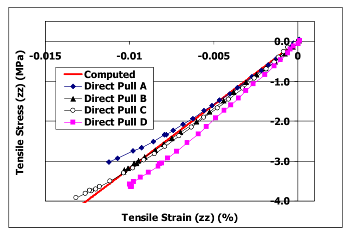
Material Identification:

Optimization (10 iterations) and Stress vs. Strain Results









177