Some Experiences of Linux Clusters for Applications in Non-linear Solid Mechanics

Larsgunnar Nilsson

Linköping University
and
Engineering Research Nordic AB

larni@erab.se

Object

- The objects of my presentation are:
  - To **motivate and illustrate** the need for HPC, in particular parallel computing, in the field of non-linear mechanics.
  - To show some **benchmark results** and discuss three years experience from Beowulf clusters.

Outline

- Background
- Design optimization
  - General
  - Gradient based optimization
  - RSM
  - Examples of application
- Beowulf cluster
- VDI/Audi benchmark
- Conclusions

Motivation

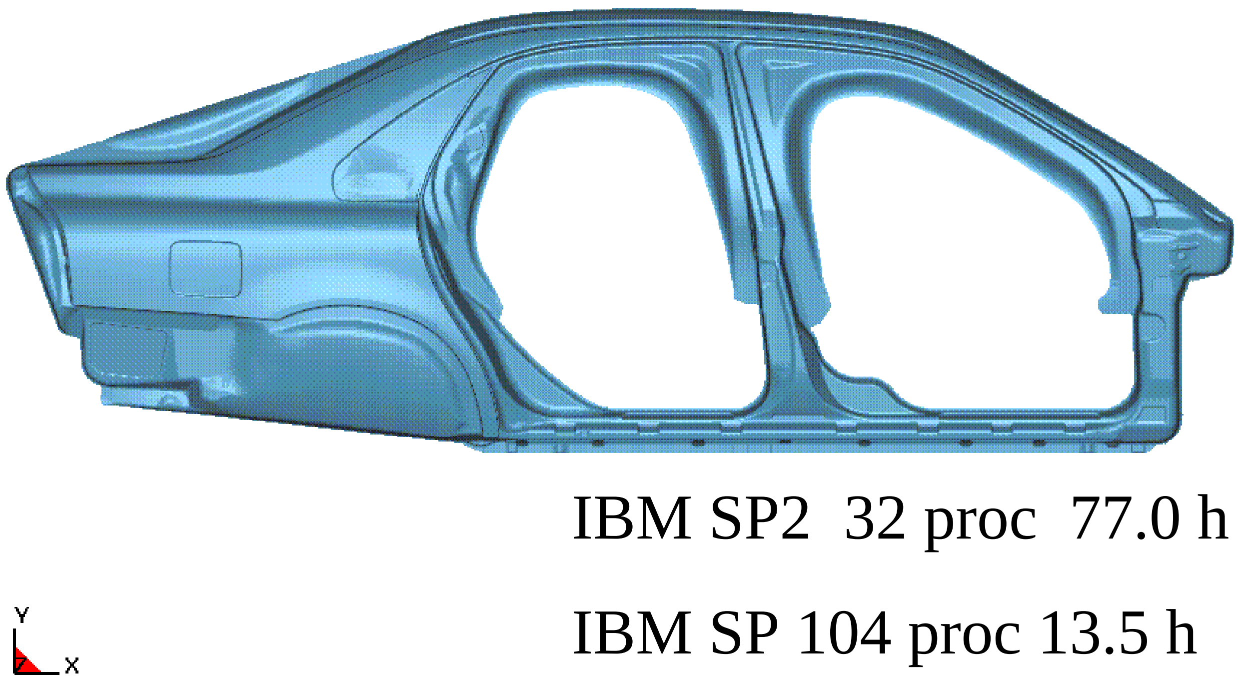
- The use of Simulation Based Design (SBD) in industry increases rapidly:
  - more detailed models
  - more design parameters being investigated
  - design optimization
- Cases:
  - Metal forming simulation of complete car side body panel requiring 1,200,000 shell elements (VOLVO Car Corp)
  - Design optimization of 30,000 shell element car structure, requiring 20 + 130 runs
    (SAAB Automobile)

VOLVO S80 Body panel
Sheet metal forming simulation

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics

1.200.000 shell elements

7.200.000 degrees of freedom

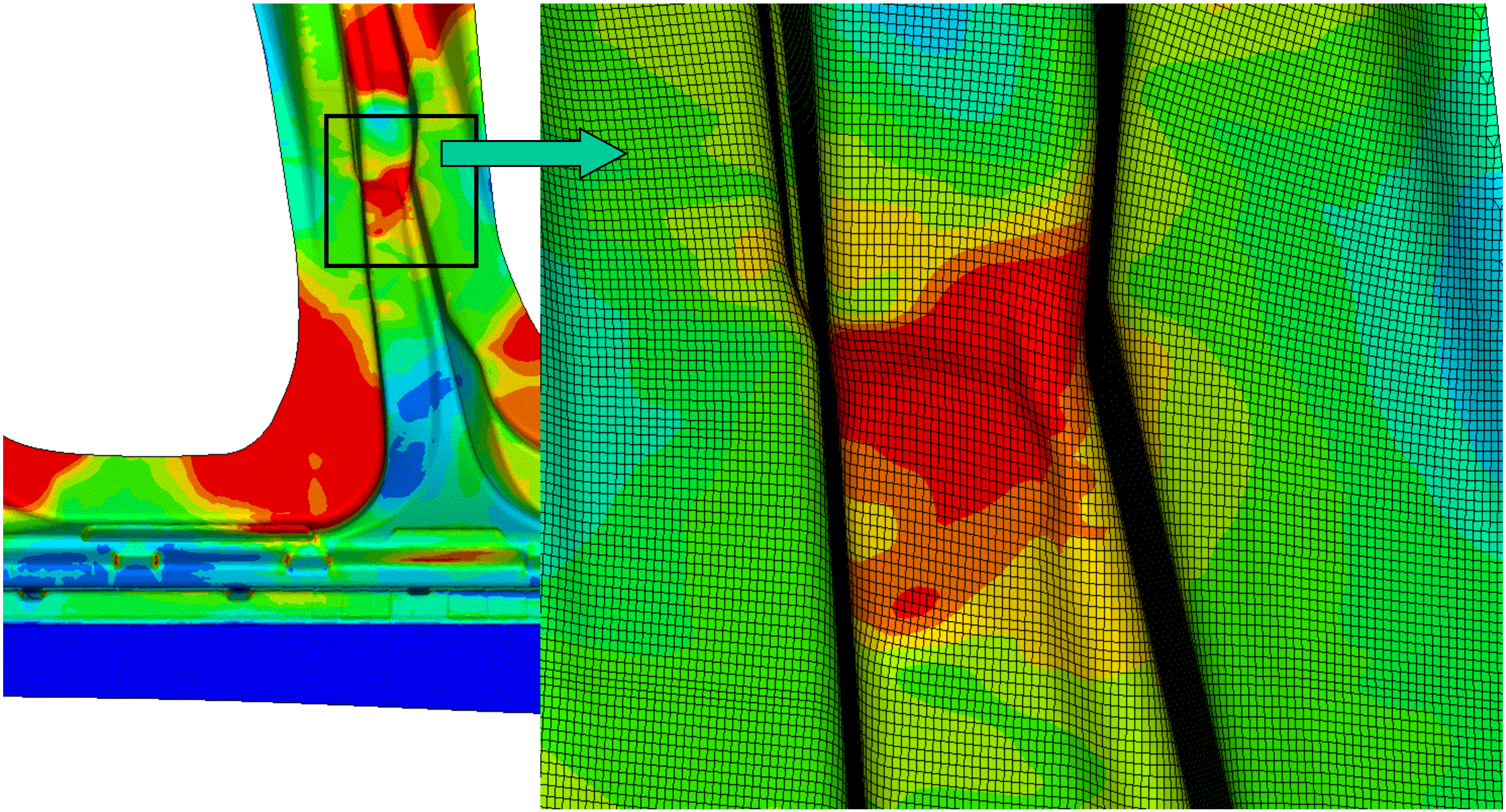


IBM SP2 32 proc 77.0 h

IBM SP 104 proc 13.5 h

VOLVO S80 Body panel
Sheet metal forming simulation

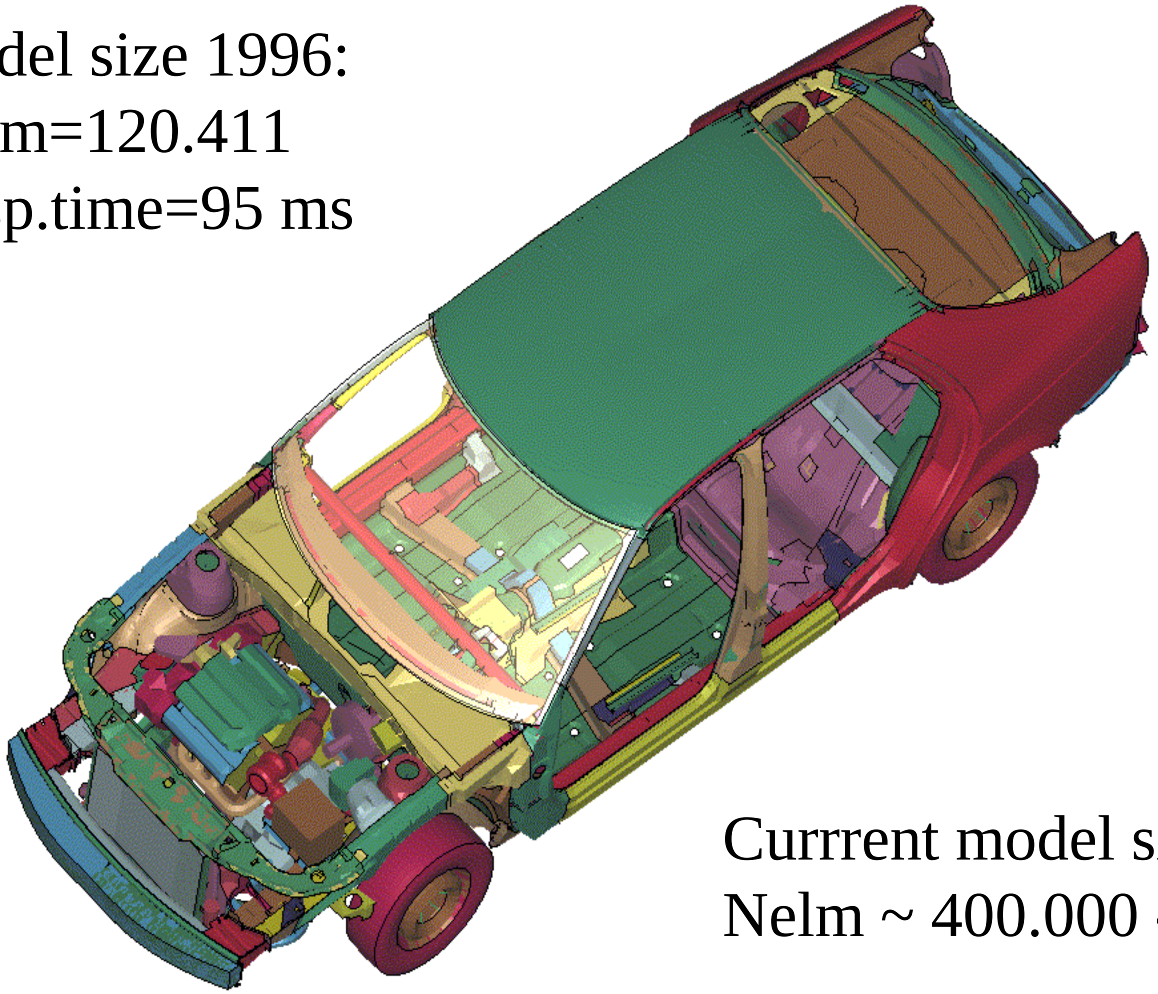
TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics



Saab 9^5 frontal crash

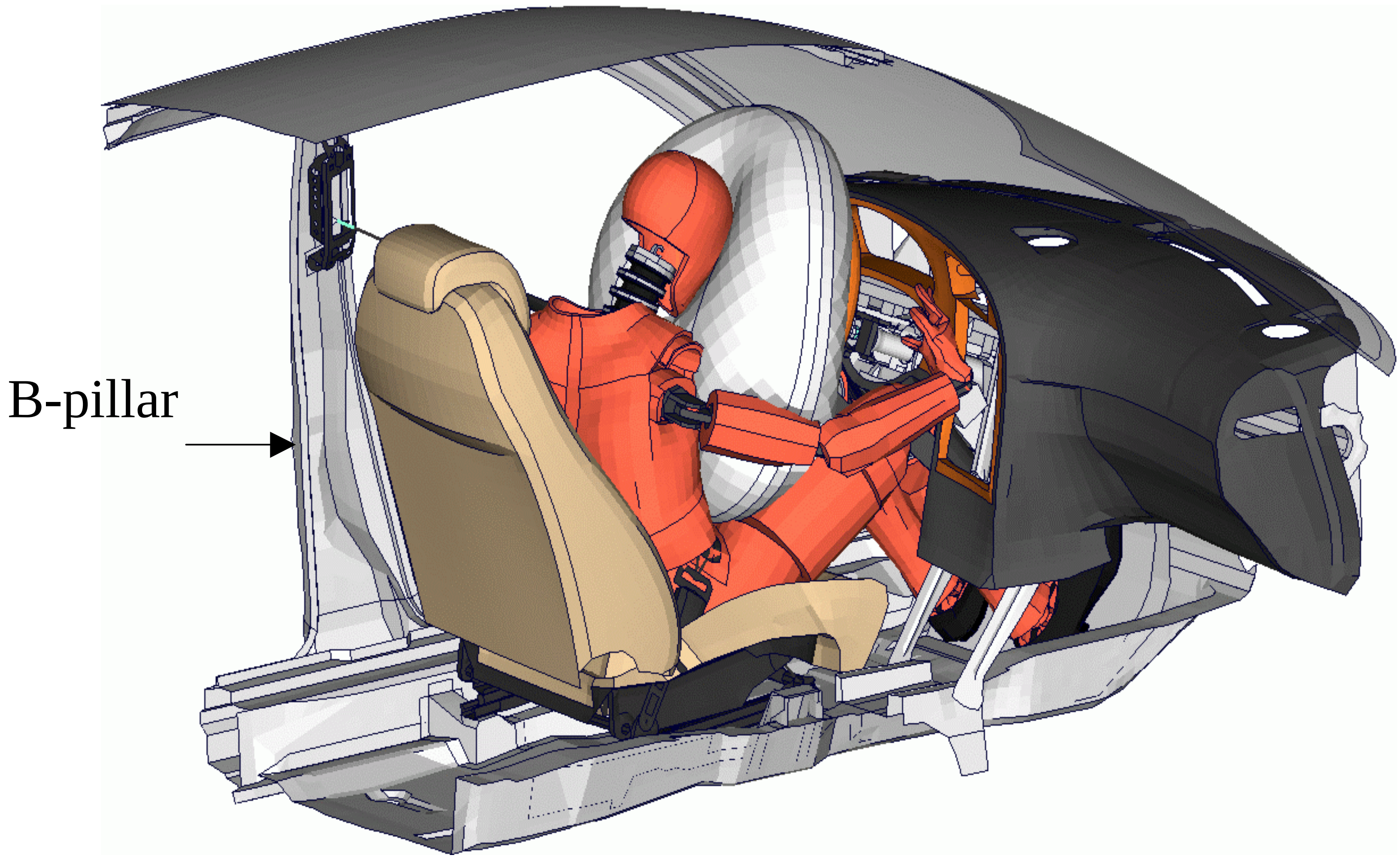
Solid Mechanics

Model size 1996:
Nelm=120.411
Resp.time=95 ms



Currrent model sizes:
Nelm ~ 400.000 - 500.000

Saab 9^5 frontal crash
Occupant crashworthiness



Design Optimization

What is Design Optimization?

- Conventional Approach Propose a design, compute the response and then make design changes to comply with safety criteria or improve efficiency. Improvement of the design may be partially rational, partially intuitive.

- Design Optimization   Parameterize the design problem. Develop simple design rules within a practical range. Cast the design rules in an Optimization Problem and solve to find a ‘better’ design. Repeat systematically until measure(s) of ‘goodness’ of the design cannot be further improved.

Problem Statement: Constrained Minimization

$$\min \; f(x)$$

subject to

$$g_j(\mathrm{x}) \leq 0 \;\; ; \;\; \mathrm{j} = 1,2,\ldots,\mathrm{m}$$

f : *cost or objective function*

g : *constraint function*

x : *design variables (parameters)*

Design Formulation
Quantities to identify

- 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
- $$\min p[f_i(\boldsymbol{x})] \quad ; \quad i = 1, 2, 3, ..., N$$
- Design constraints
- Limits on the design, e.g. strength, intrusion, deceleration
- $$L_j \leq g_j(\boldsymbol{x}) \leq U_j \quad ; \quad j = 1, 2, 3, ..., m$$

Karush-Kuhn-Tucker conditions

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

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

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

$$\lambda \geq \mathbf{0}.$$

Gradient Computation

Gradient based optimization algorithms require gradient computation

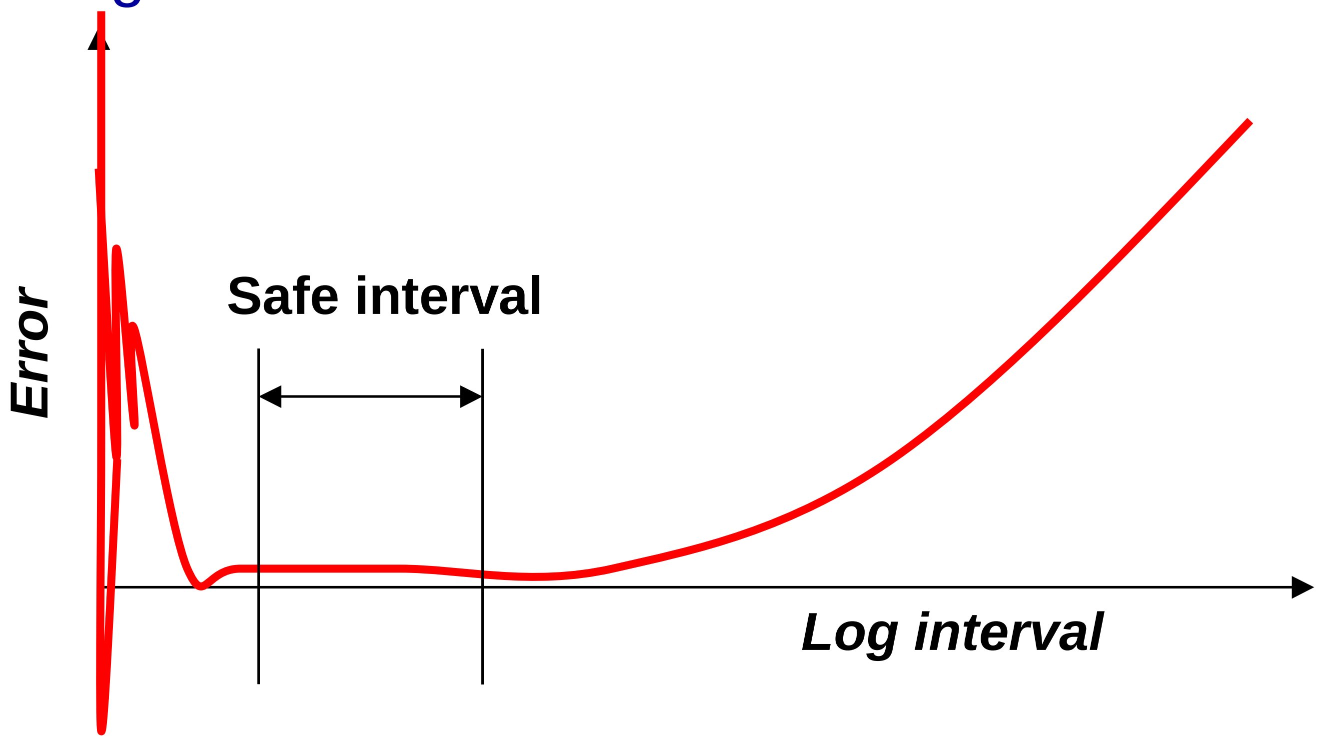
$$\frac{\mathrm{d}f}{\mathrm{d}x} \qquad \frac{\mathrm{d}\boldsymbol{g}}{\mathrm{d}x}$$

Gradients are

- **Analytical: Derivatives are formulated explicitly and implemented into the code. Complicated.**
- **Numerical: Design is perturbed and (n+1) analyses are simulated. Simple but expensive and error prone.**
- **Semi-Analytical : Partly numerical, partly analytical (chain rule)**

Numerical gradients: accuracy

- Accuracy.
- If the perturbation interval is too large, lose accuracy
- If the perturbation interval is too small, find spurious gradients



Causes of spurious derivatives

- Spurious derivatives computed using small intervals are due to:
- Chaotic structural behavior.
  Especially in crash analysis.
- Adaptive mesh refinement.
  Different designs have different meshes.
- Numerical Round-off error.
  Usually single precision computations.

Design Environment

- Non-linear behavior and adaptivity.
- Noisy response.
- Analytical design sensitivities not available

Optimization algorithms directly based on gradients are infeasible !

Approximations
Local and Global

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics

- Local
  - Design Sensitivity Analysis (DSA)
    - Analytical: Formulate and implement the derivatives
    - Numerical: Perturb the design. Uses *n+1 simulations*
- Global
  - Response Surface Methodology (RSM) *[Box and Draper (1959)]*
  - Neural Networks (non-linear regression), Radial basis functions (linear regression)
  - Gaussian processes (Kriging - non-parametric regression)

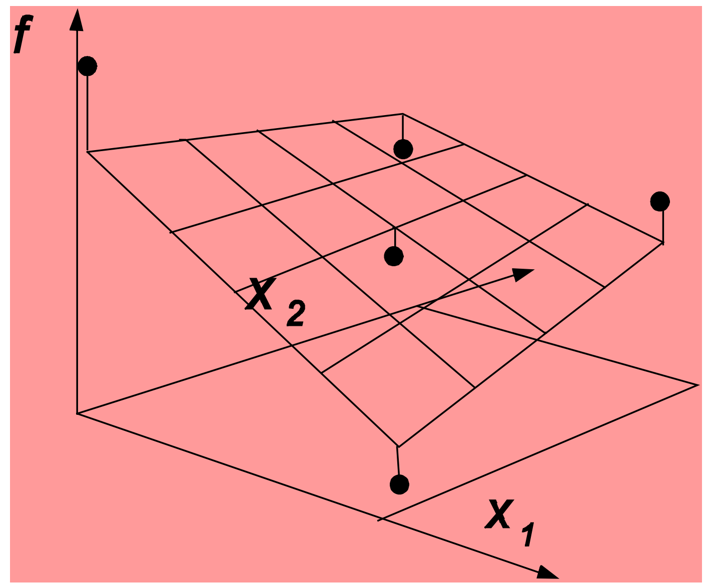
Response Surface Methodology

- Creates design rules based on global approximations
- Does not require analytical sensitivity analysis
- Smoothes the design response and stabilizes numerical sensitivities
- Accurate design surfaces in a sub-region allow for inexpensive exploration of the design space (e.g. sensitivity analysis, multi-objective design) without further function evaluation. Trade-off curves developed interactively.

How does it work?

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics

- Design surfaces (*f* and *g*) are fitted through points in the design space to form approximate optimization problem



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

Approximating the response

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

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.

Approximating the response

Sum of the square error:

$$\sum_{p=1}^{P}\{[y(x)-f(x)]^2\} = \sum_{p=1}^{P}\{[y(x)-\sum_{i=1}^{L}a_i\phi_i(x)]^2\}.$$

P: number of experimental points
y is the exact functional response at the experimental points x_i.

Approximating the response

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.

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

Approximation models

$$1 \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \begin{bmatrix} x_1^2 & x_1x_2 & \dots & x_1x_n \\ x_2x_1 & x_2^2 & \dots & x_2x_n \\ \vdots & \vdots & \dots & \vdots \\ x_nx_1 & x_nx_2 & \dots & x_n^2 \end{bmatrix}$$

Linear

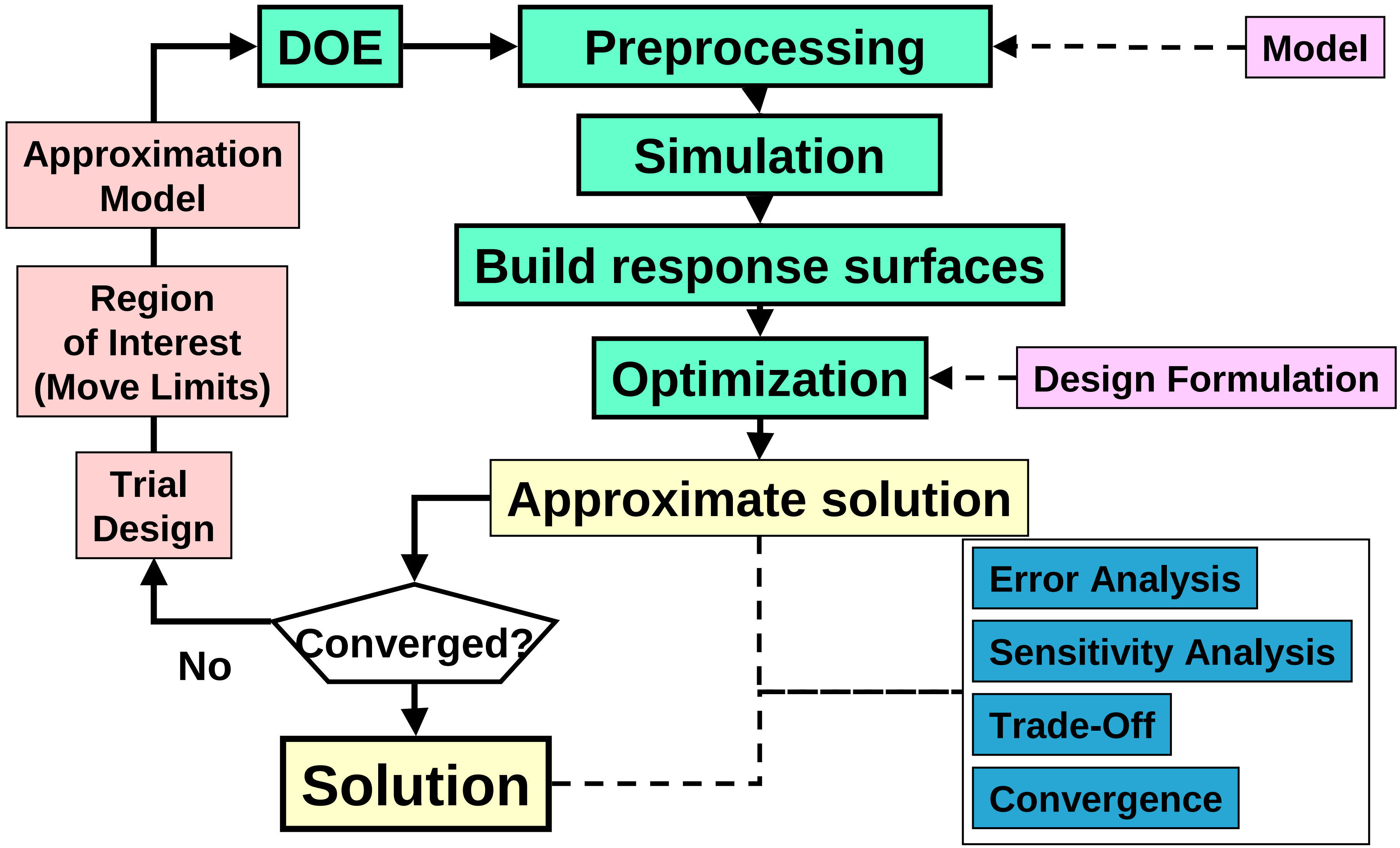
Elliptic

Quadratic

Approximations

- First order approximations
  - Inexpensive. Cost ~ *n*
  - Cycling (oscillation) can occur. Successfully addressed by adaptive optimization algorithm
  - Robust iterative method
- Second order approximations
  - More expensive. Full Quadratic: Cost ~ *n*-squared
  - Elliptical approximation: Cost ~ 2*n*
  - More accurate. Good for trade-off studies
- Linear Approximation is recommended

The Optimization Process

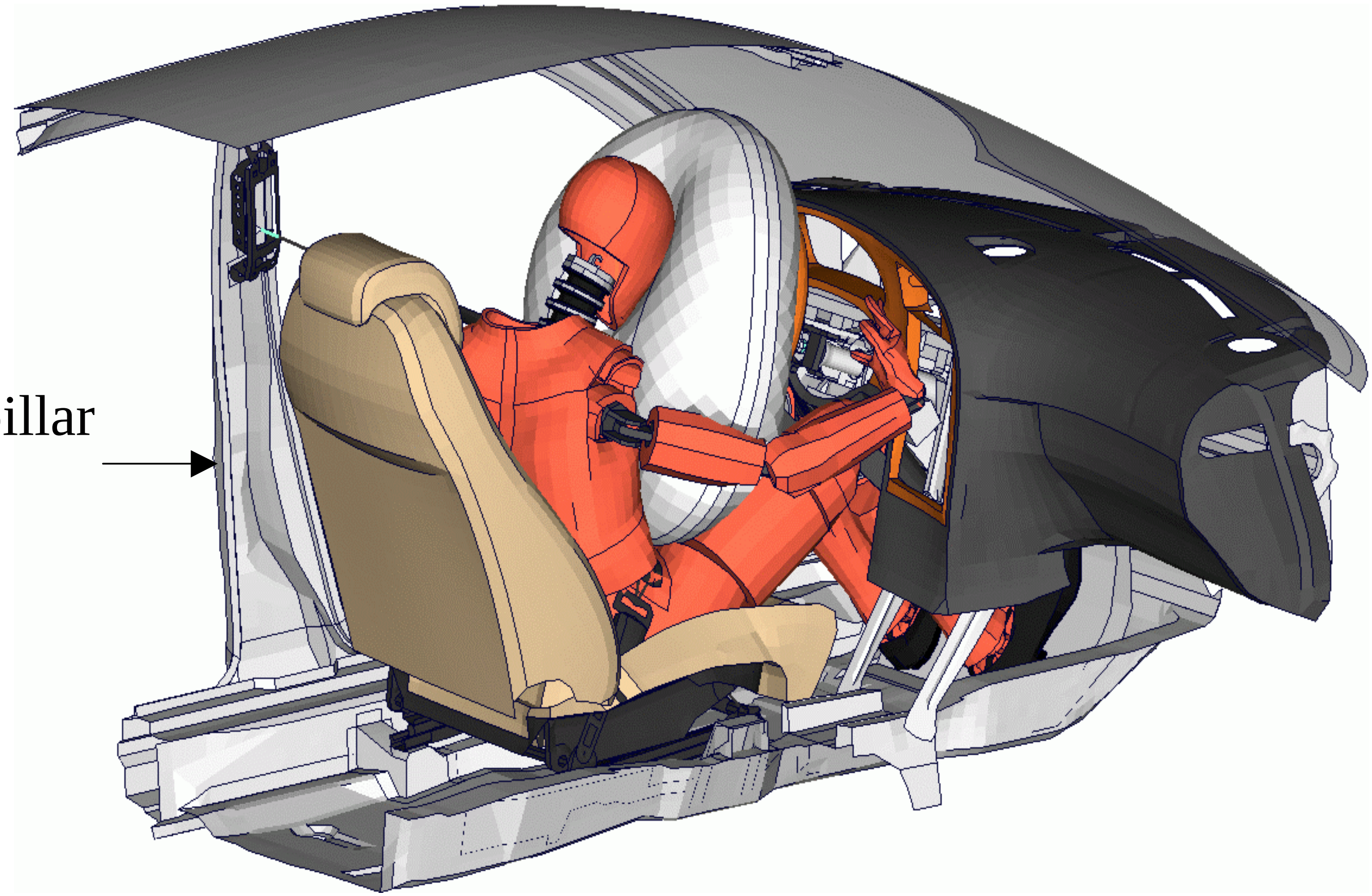


Optimization of a car body component subjected to side impact

Saab 9^5 frontal crash
Occupant crashworthiness

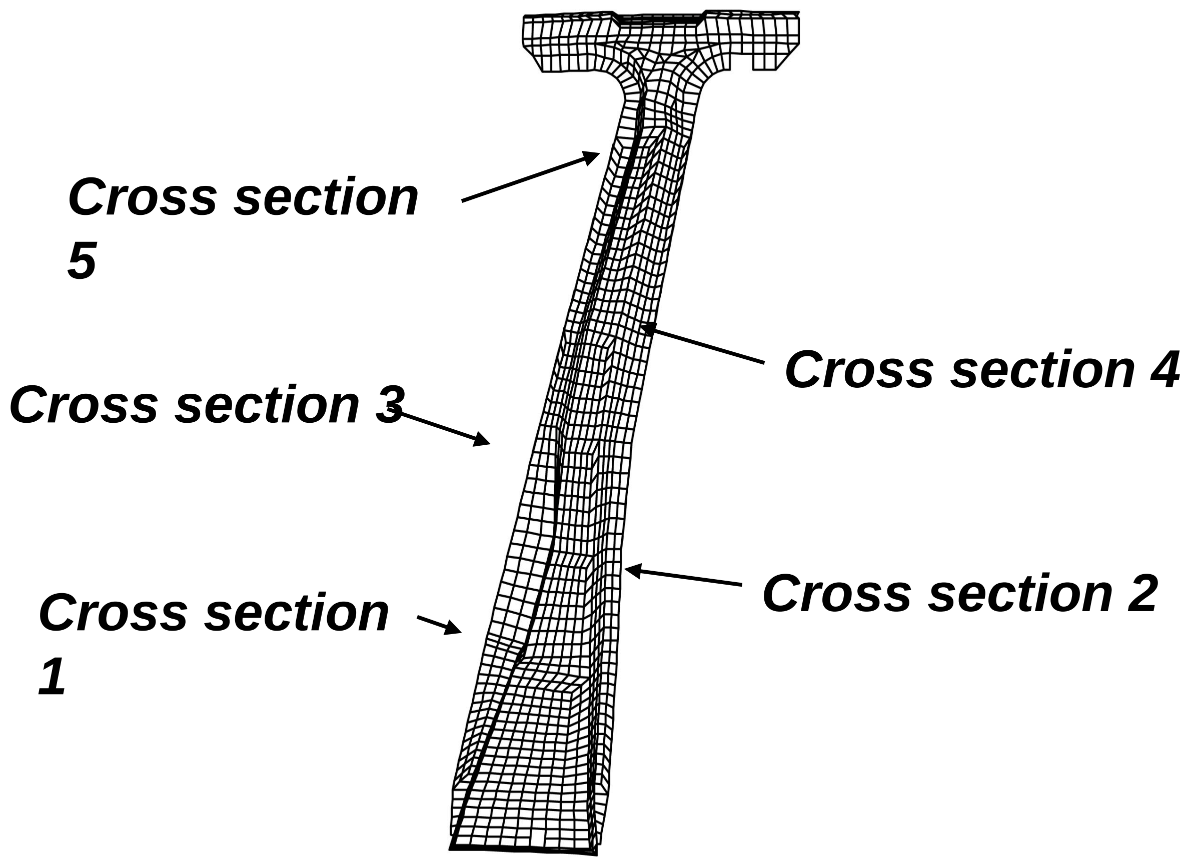
Solid Mechanics

B-pillar

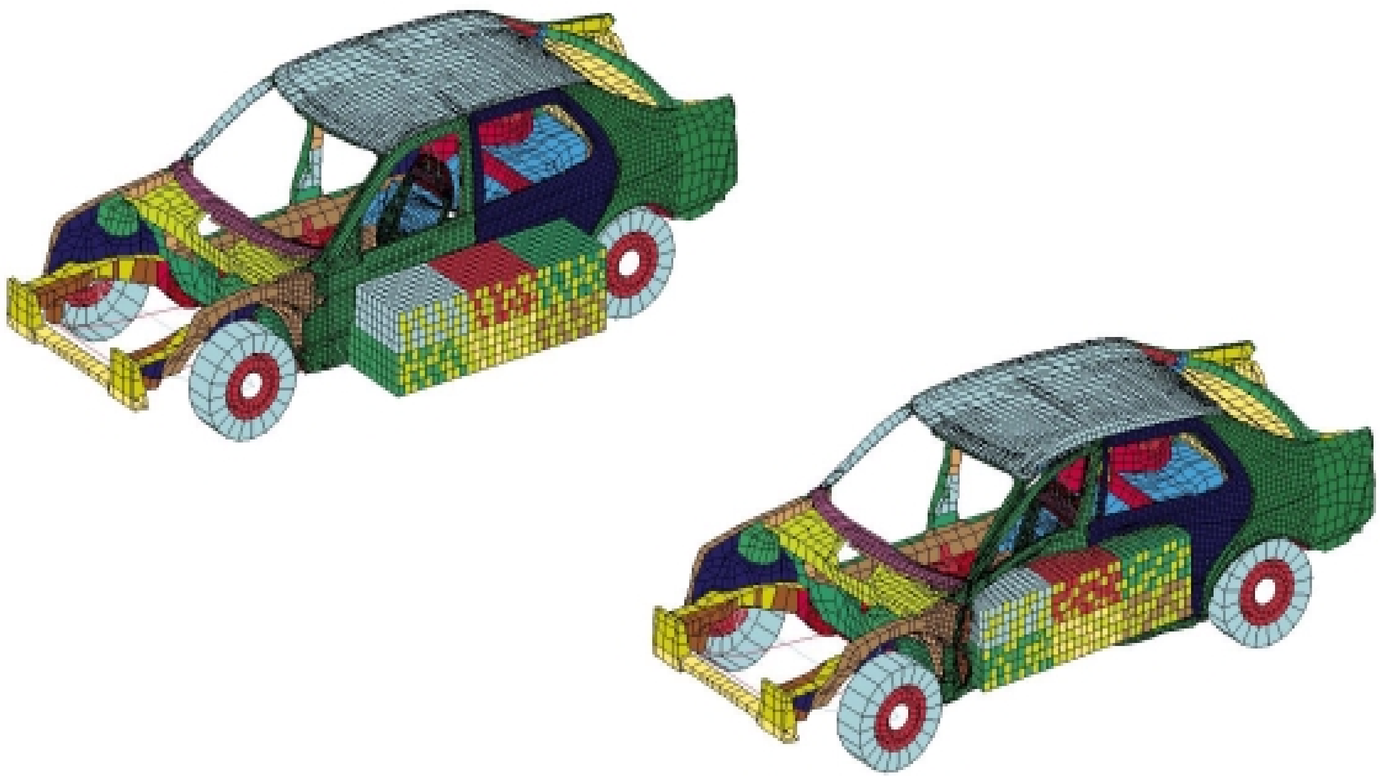


Parametric B-pillar.

Totally 11 design parameters



Saab 9^5 side impact model



B-pillar weight optimization

Solid Mechanics

Minimize

Mass (x)

subjected to the constraints:

$$v_{top}(x_i) \leq v_{top}^{orig}$$

$$v_{mid}(x_i) \leq v_{mid}^{orig}$$

$$v_{bot}(x_i) \leq v_{bot}^{orig}$$

With design variables intervalls:

$$x_i^{\min} \leq x_i \leq x_i^{\max}$$

Linear response surfaces

11 design parameters:

- Full factorial design
  - 2^{11} = 2048 design points
- Koshal design
  - 11+1=12 design points
- D-optimal design
  - Using (11+1)*1.5 ≅ 20 design point
  - Plus 5 check points

Quadratic response surfaces

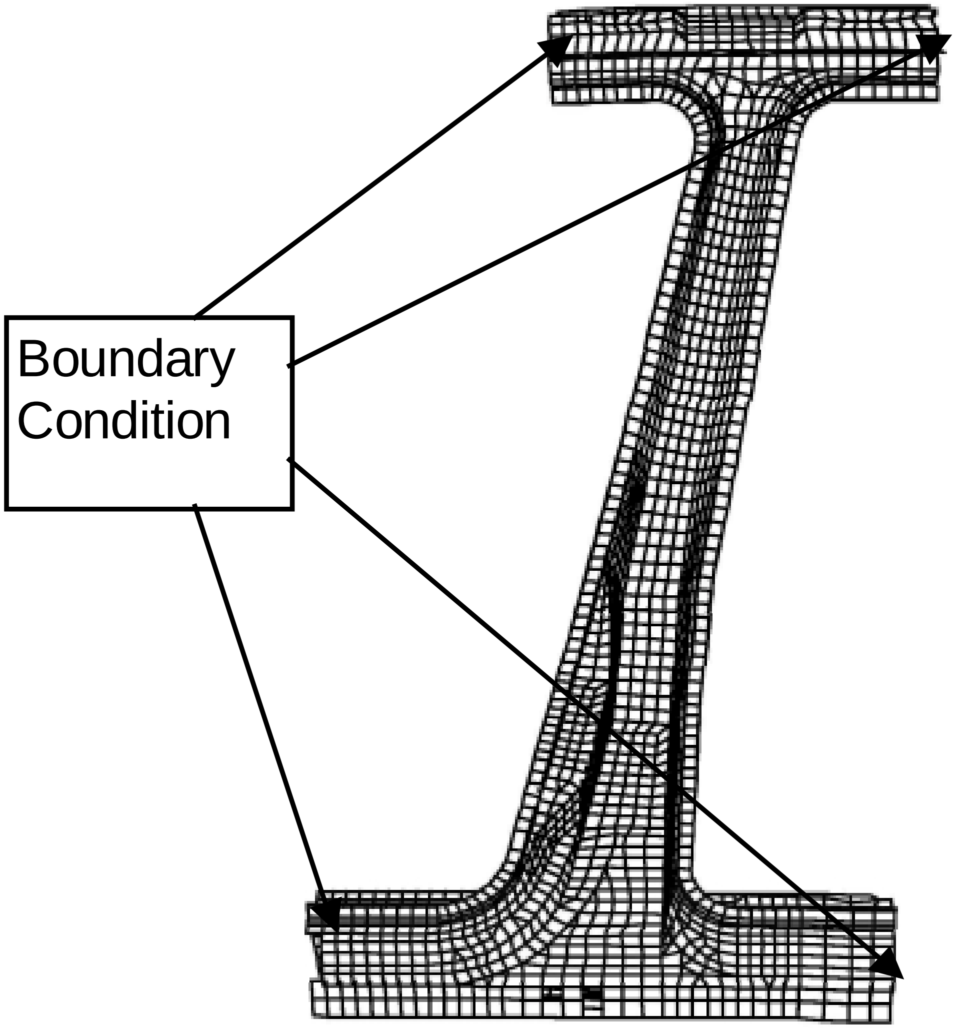
11 design parameters

- Full factorial design
  - 3^{11} = 177,147 design points
- Koshal design
  - (11+1)*(11+2)/2 = 78 design points
- D-optimal design
  - Using 78*1.5 ≅ 120 design point
  - Plus 10 check points

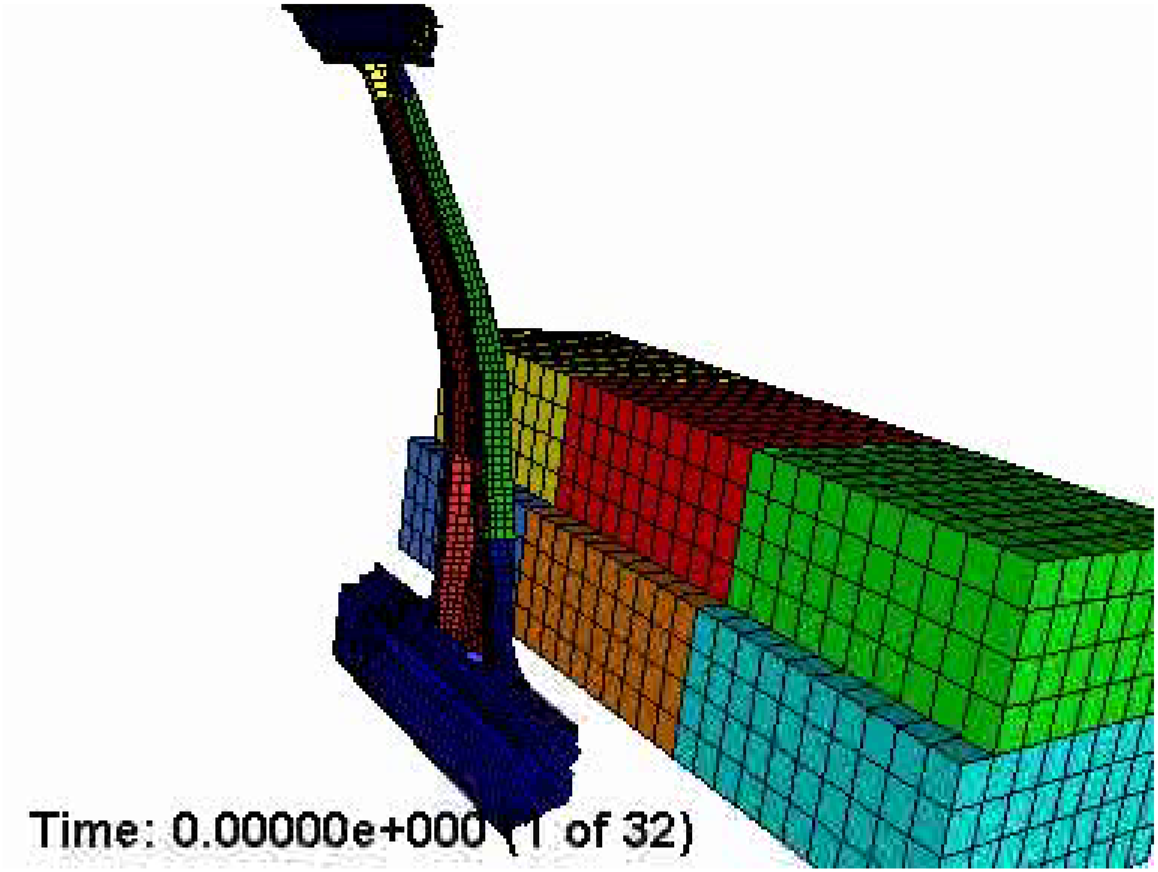
Model idealization

- Side impact with full car model
  - Each LS-DYNA run takes about 22 hours
- Reduced model; side impact on B-pillar
  - Each LS-DYNA run takes about 5 hours
  - Boundary conditions on interfaces to roof and door sill from side impact on full car model (original design). ”Re-analysis”

Boundary conditions for B-pillar

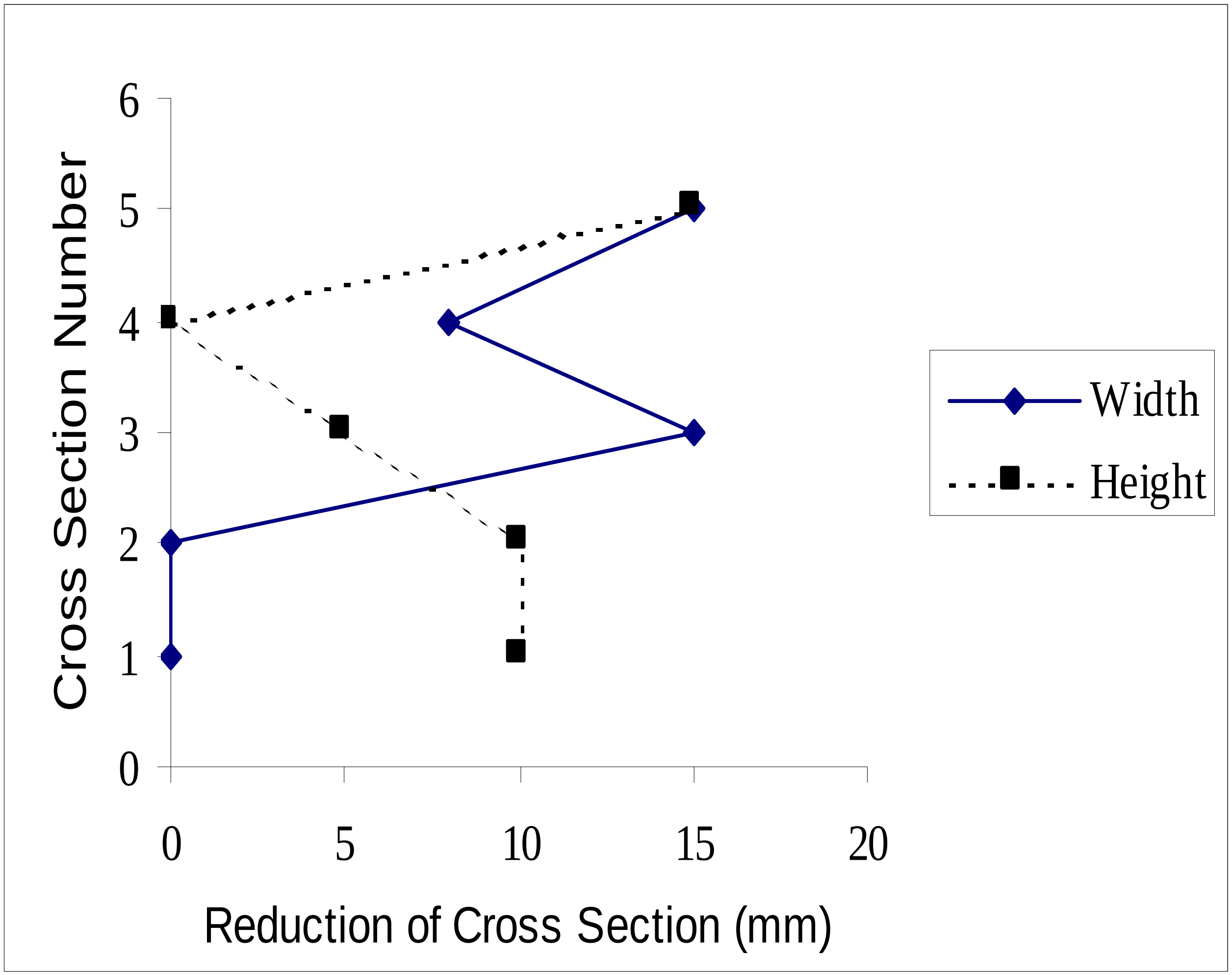


Component analysis



Design parameters at optimum

Solid Mechanics



Weight reduction summary

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics

	Change
Reduced section	-1,9 kg/car
Part reduction	-1,6 kg/car
Total weight red.	-3,5 kg/car About 25%

Surrogate models and Space Mapping

Space Mapping

Solid Mechanics

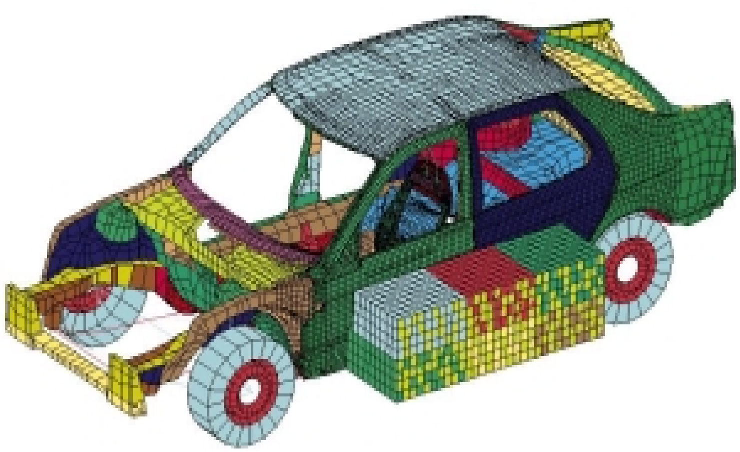
Original “costly” model

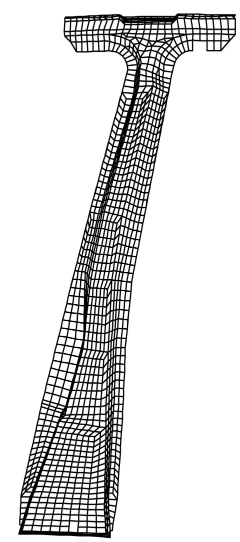
Surrogate model

+ Detailed
+Accurate
- Slow
- Costly

Mapping

+ Fast
+ Less costly
- Less accurate





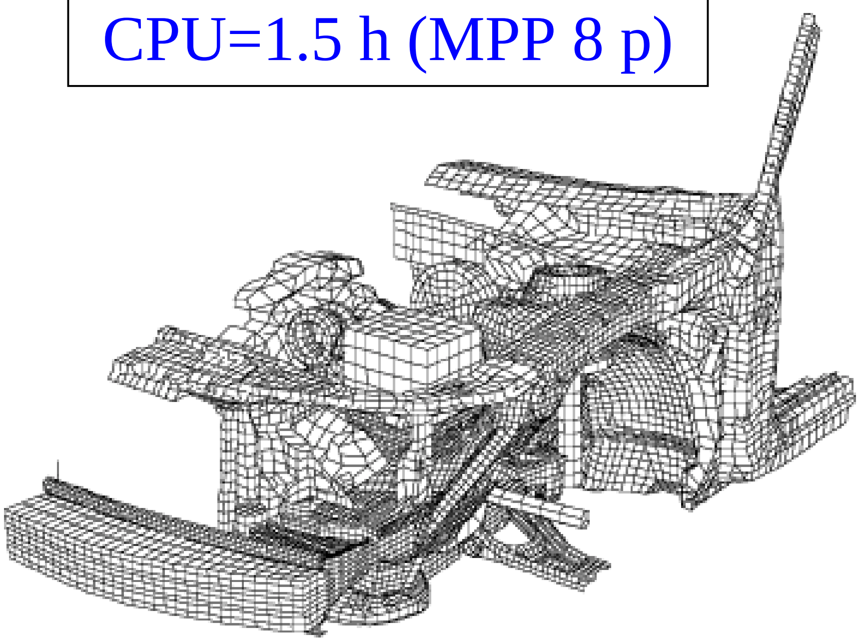
Carbody, frontal structure

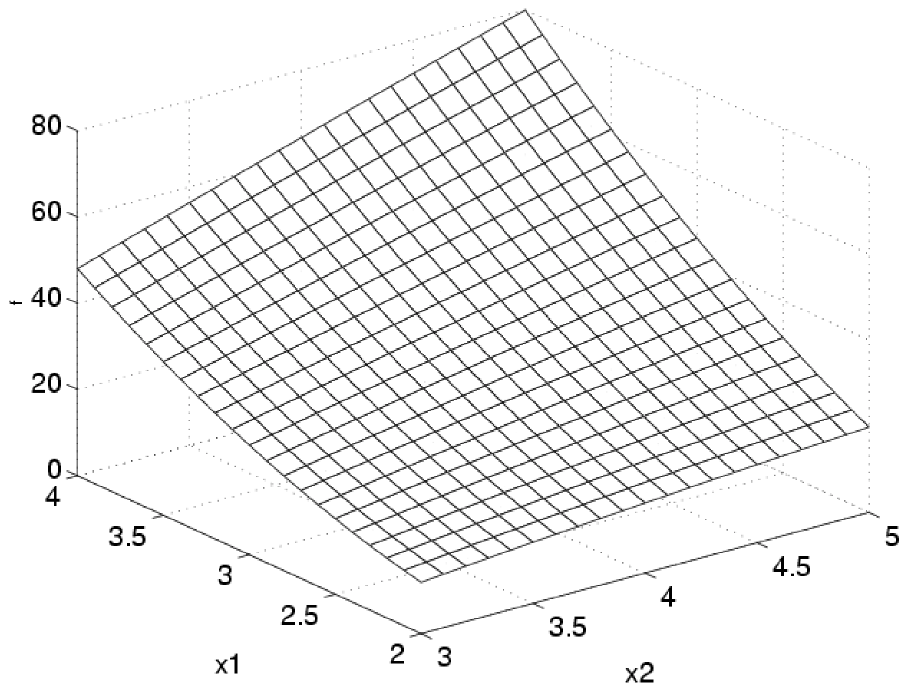
Solid Mechanics

4 design parameters,
Quadratic response surfaces
=> 23 runs

Surrogate model
CPU=0 h

Fine model
CPU=10 h (SMP)
CPU=1.5 h (MPP 8 p)





CPU=10 h *23=230h
Total=230/8~30h

Example – Frontal crash

$$max\,(initial\ \ acceleration\,(a1))$$

$$intrusion < a_{ref} = 72.2\ mm$$

$$stop\ time > t_{ref} = 78.3\ ms$$

$$acc_{max} < \mathrm{a}_{\mathrm{ref}} = 624\ m/s^2$$

$$1.4\ mm < t_1 < 2.0\ mm$$

$$1.5\ mm < t_2 < 2.0\ mm$$

$$1.2\ mm < t_3 < 1.8\ mm$$

$$180\ MPa < \sigma_y < 420\ MPa$$

a1=mean(acc), 0<t<20 ms

Design parameters:
3 thicknesses
1 material property

Space Mapping
- Frontal crash

Table 4.5: Optimization results of the vehicle model using space mapping

	a1	stop time	intrusion	maximum acceleration
Start point	155.0	0.078	0.072	624
Iteration 1	166.6	0.067	0.056	607
Iteration 2	172.2	0.079	0.081	529
Iteration 3	170.7	0.079	0.074	484
Iteration 4	174.0	0.079	0.073	386
Iteration 5	177.6	0.079	0.072	510
Iteration 6	171.5	0.079	0.072	440
Optimum point	171.8	0.079	0.067	540

Space Mapping – Frontal crash

CPU (SMP) =7*10+230=300h=12.5 days

CPU (MPP) =7*1.5+30=40.5h=1.7 days=>

7.3 speed-up on 8 proc

LS-DYNA performance on Linux cluster

HPC future ...

- The increase in single processor performance will slow down
- Most HPC will be based on parallel computing
- Most mpp's will be based on mainstream processors
- The Beowulf "supercomputer" is the "poor-man's" choice
- mpp/LS-DYNA offers very good performance

Linux environment

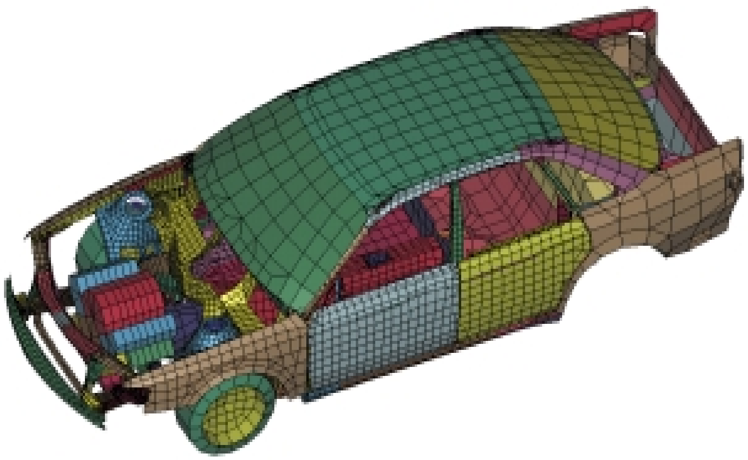
- Operating system Red Hat Linux 7.0
- Message Passing Interface (MPI), MPIch and LAM
- Portable Batch System (PBS)
- mpp/LS-DYNA 940.02 and 960

mpp/LS-DYNA

- Domain decomposition
  - RCB, RSB, Greedy
- Single Program Multiple Data (SPMD)
- Message Passing using MPI
- Linux versions available
  - MPIch and LAM

VDI Audi frontal crash

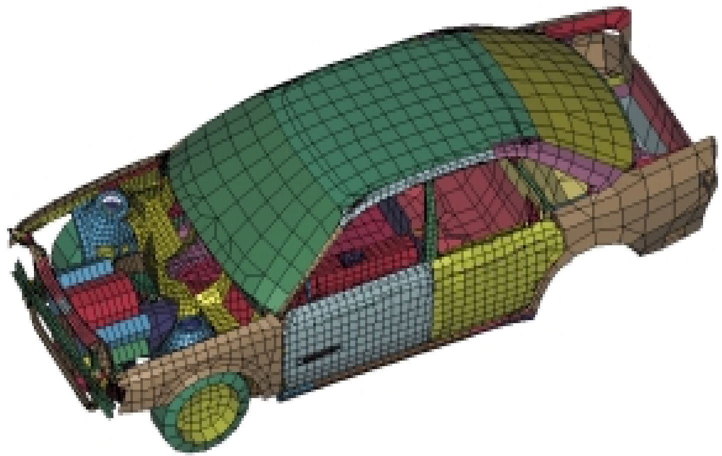
Solid Mechanics



 # shells = 28007

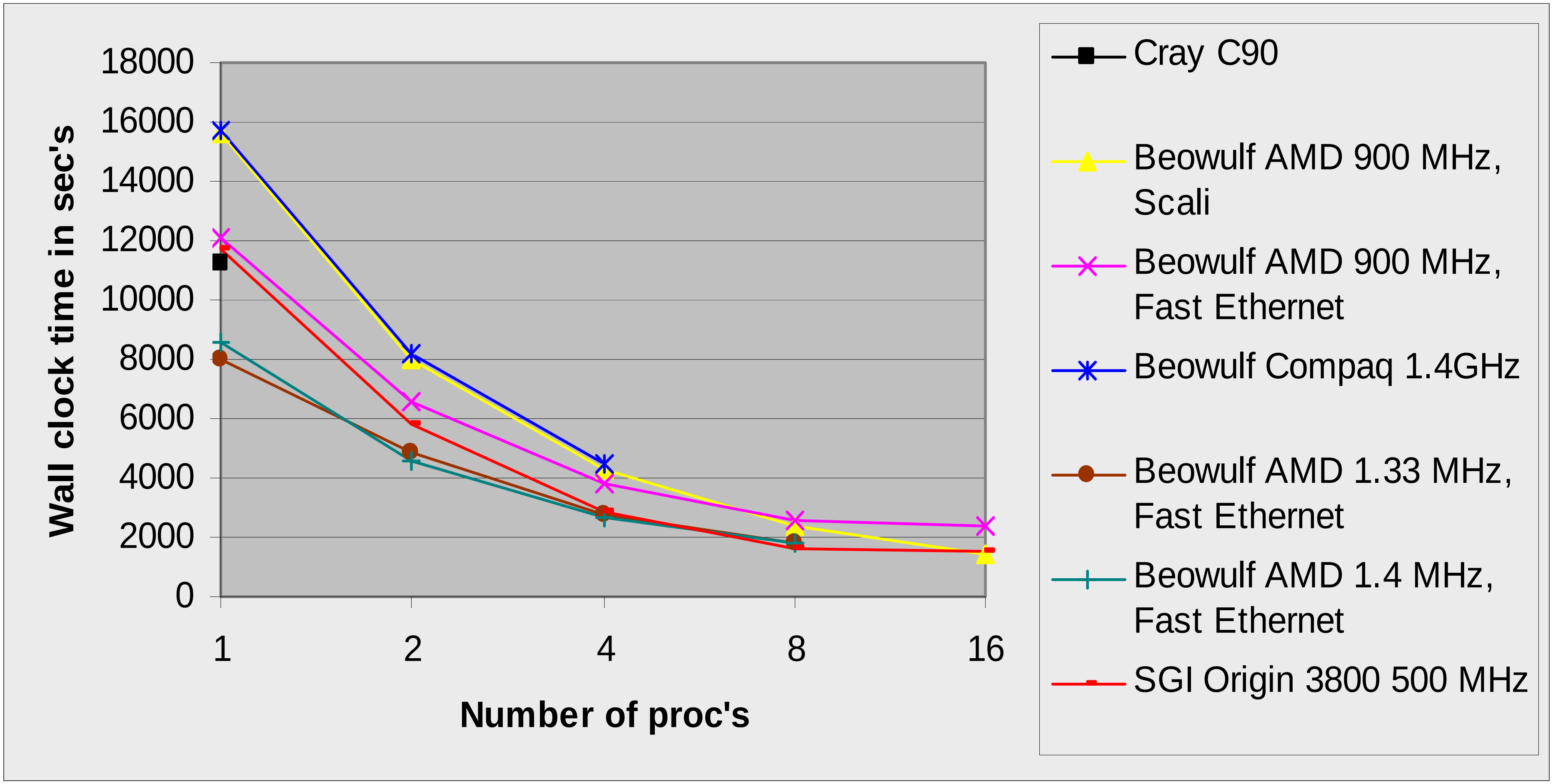
 # beams = 216

Response time = 50.35 ms



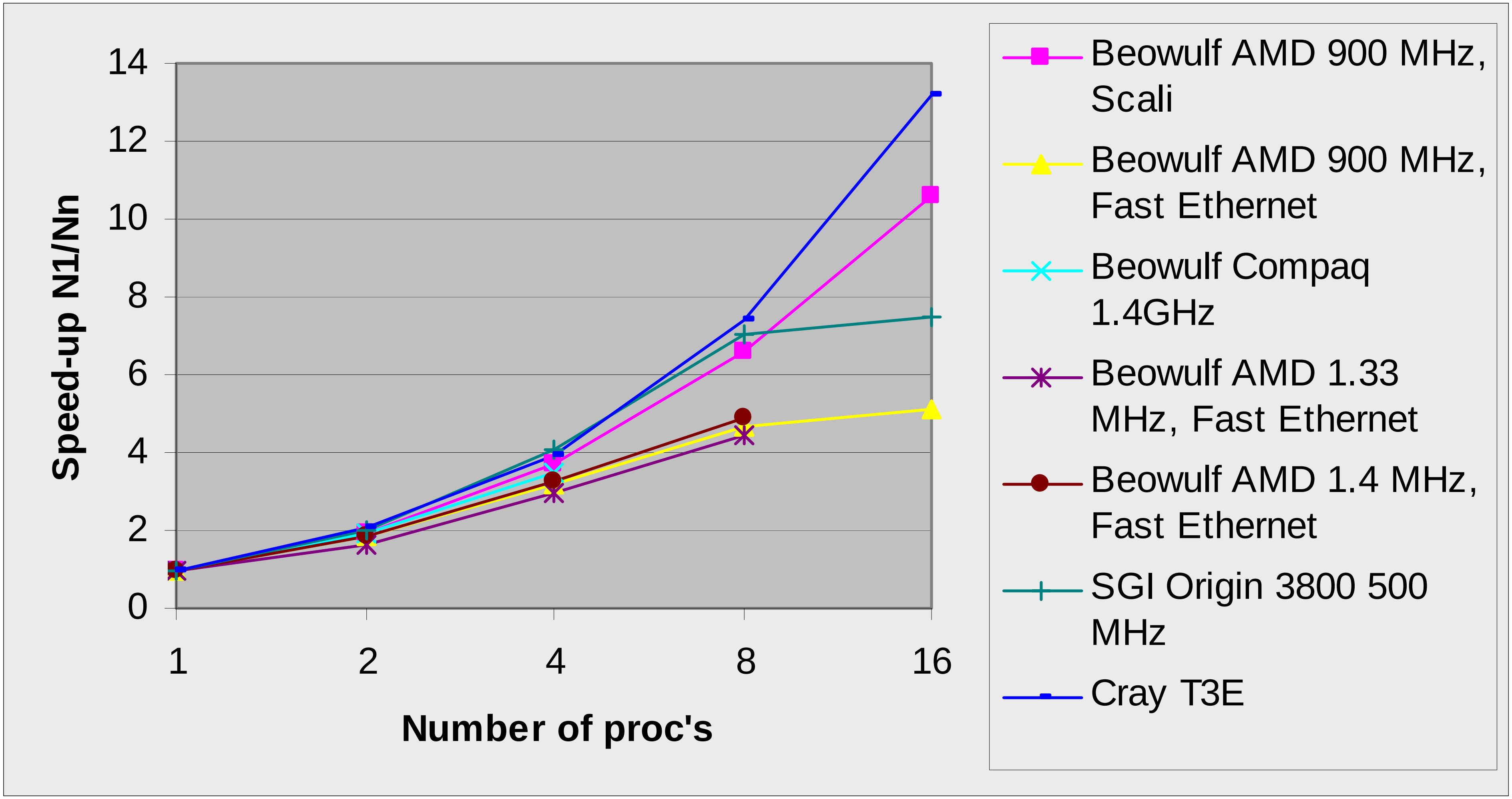
VDI/Audi benchmark

Solid Mechanics



VDI Audi frontal crash
Speed-up (N1/Nn)

Solid Mechanics



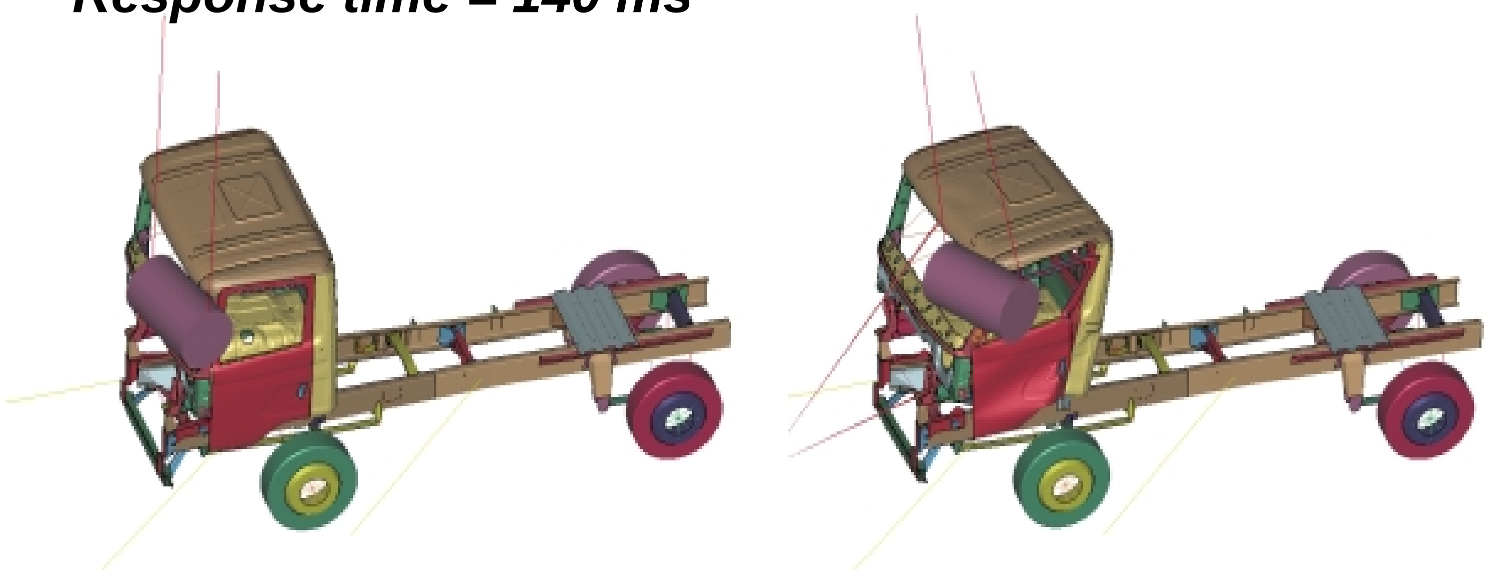
Swedish test on Scania truck

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics

 # shells = 159260

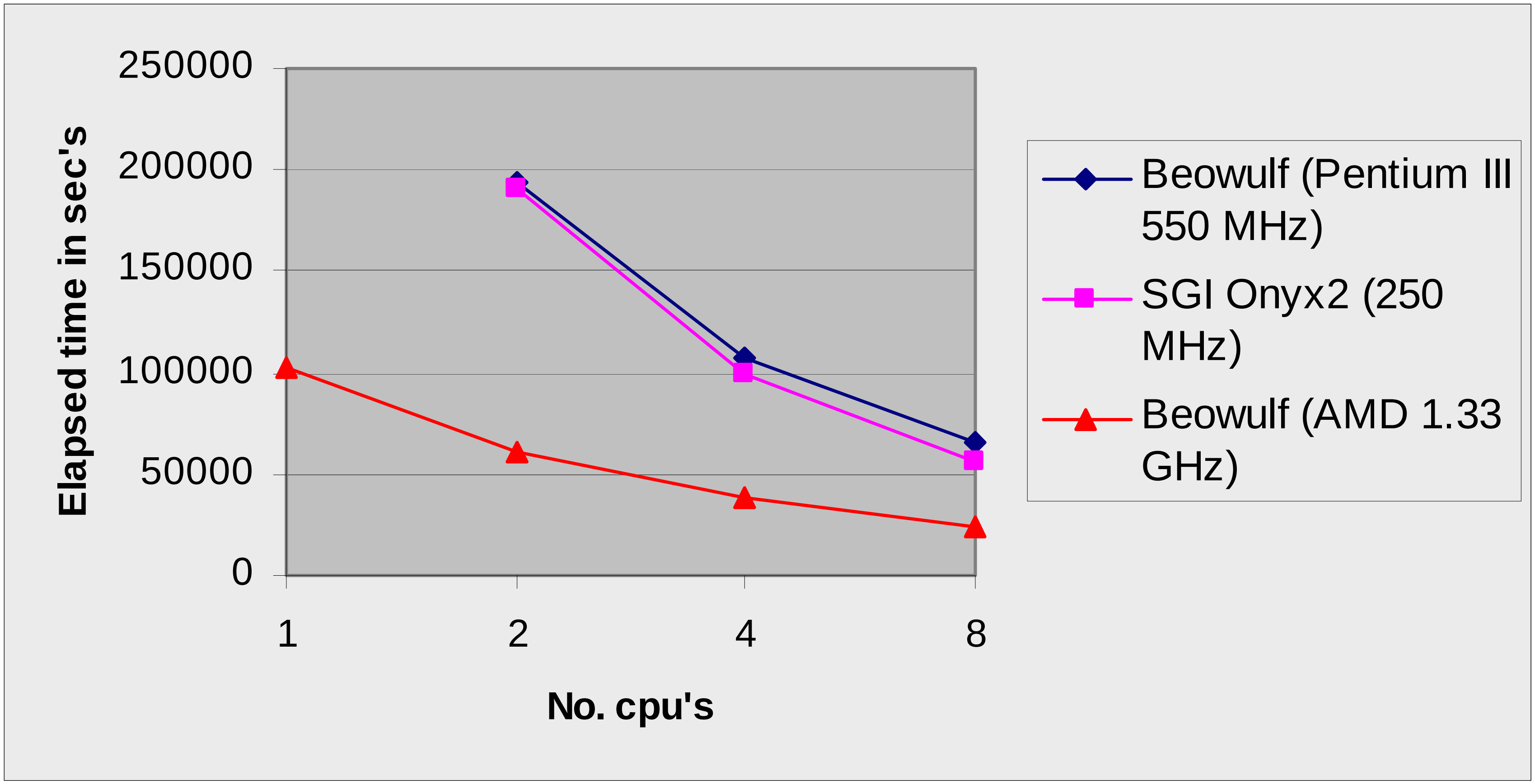
 # beams = 449

Response time = 140 ms



Swedish test on Scania truck

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
Solid Mechanics



Conclusions

- Finite Element simulations are used as a day-to-day tool in industry
- Simulation is evolving into Simulation Based Design, where optimization is a key feature
- Standard gradient based optimization techniques are in-feasible in non-linear mechanics applications
- Response Surface Methodology (RSM) is a global optimization technique that has shown very efficient
- For accuracy, RSM needs a large number of functional evaluations, and each of these requires high performance computing, in particular MPP
- Beowulf is the ”poor mans” choice for high performance computing using MPP