Linear Programming service of Statistics

Reduction of an automotive crash model

March 24, 2016

Thuy Vuong, CIFRE RENAULT

Yves Le Guennec, IRT SystemX

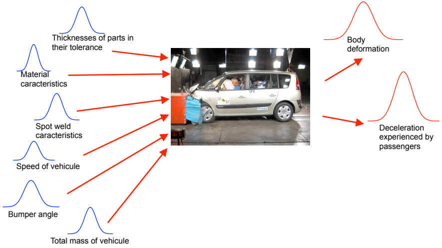
Yves Tourbier, RENAULT & IRT SystemX

Outline of the presentation

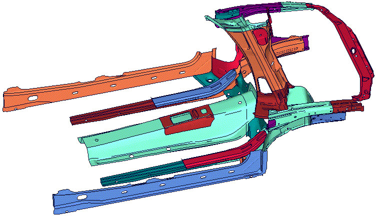
01 INTRODUCTION

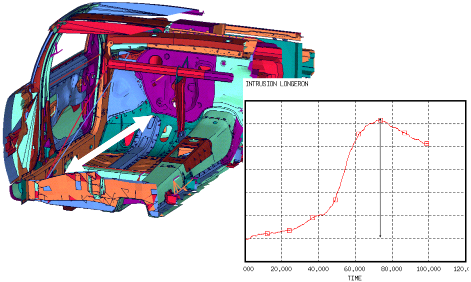
- Context
- Example of classical study
- Standard studies with DOE
- Conclusion on DOE studies

- The objective is optimization or modelisation (the objective concerns the outputs)
- The system is a black-box (we don't know how it works inside)
- Each evaluation of the black-box is very costly
- One cannot choose directly the outputs, just fix the inputs then run the black-box to measure outputs

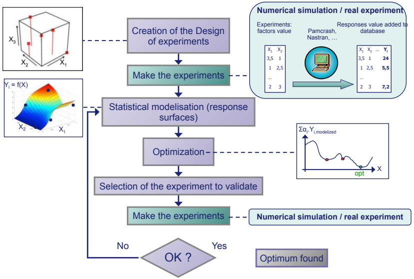


- The list of parts to vary is limited but can be long
- Frontal crash $\approx$ 20 criteria $\Longrightarrow$ 20 constraints
- Up to 100 parameters (thicknesses, materials, reinforcements, shapes, spot welds...)





1. Define the parameters, responses, optimization problem, reproductibility...
2. Build a DOE
3. Perform tests
4. Make a statistical model for each response (LP can be used here)
5. Use the models to optimize, propose one solution (or more)
6. Validate the solution(s) with new test(s)
7. If necessary, go back to 1, 2 or 4



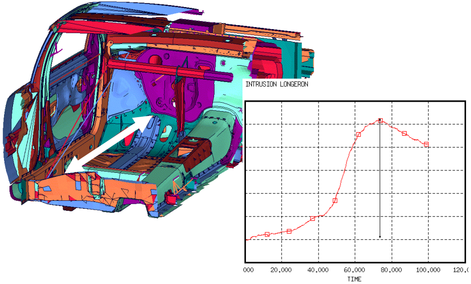
- Total cost = Number of crash simulations $\in [3, 10]$. Number of parameters
- Can be fully automatic with EGO like methods
- Number of parameters can be greater than 100 with standard DOE method
- BUT:
  - User must analyze a lot of solutions
  - One cannot use all the criteria (too numerous, scenario & subjectives criteria are only used as final selection)
  - Number of crash simulations is a bootleneck
  - The surrogate model use less than 1% of data produced by crash simulations

02 Model reduction

Replace the entire crash simulation by a surrogate

A simple crash simulation generates Gbytes

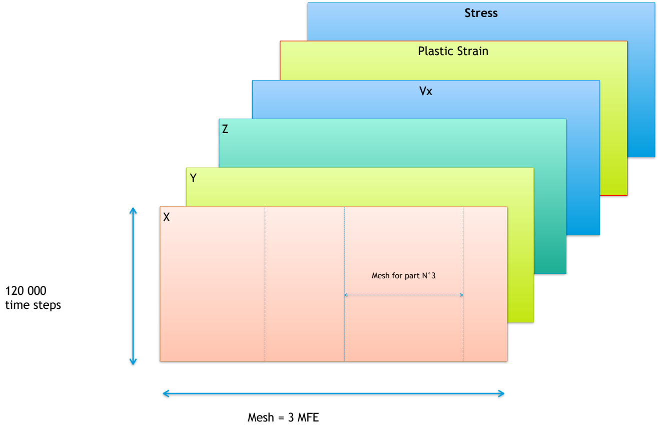
- Standard DOE → surrogate = function(max(Intrusion))
- Model reduction → max(surrogate = function(Intrusion))



$$HIC = \{[\frac{1}{t_2 - t_1} \cdot \int_{t_1}^{t_2} a(t).d_t]^{2.5} (t_2 - t_1)\}_{Max}$$

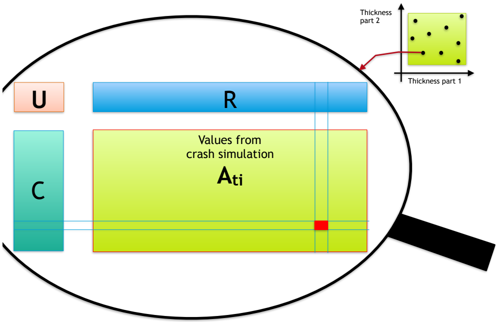
A simple crash simulation generates Gbytes

- > 15 values computed for each node / finite element at each time step
- > 40 Gb / simulation



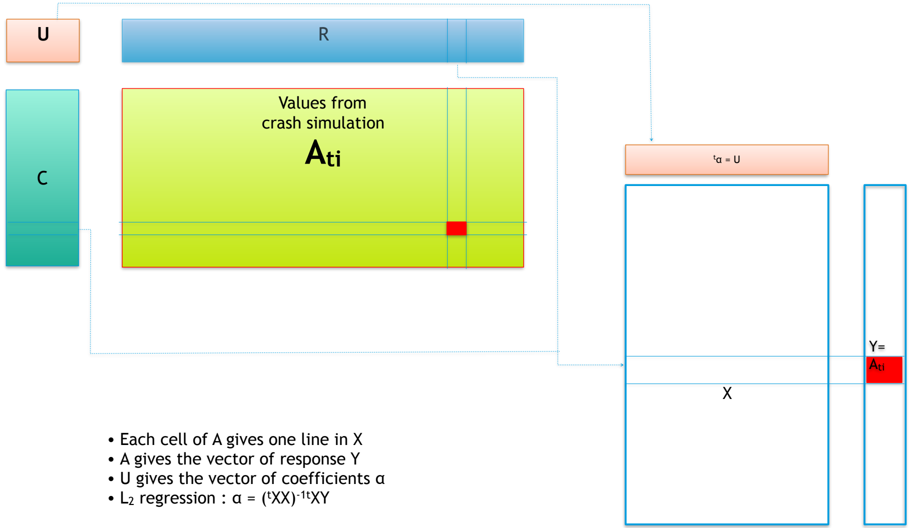
03 CUR METHOD

- The conventional CUR method
- CUR and Linear Regression
- CUR and Linear Programming



- $$A \approx C.U.R$$
- Each cell A_{ti} is fitted by a linear model:
$$A_{ti} = \sum_{k_C=1}^{d_C} C_{t,k_C} \sum_{k_R=1}^{d_R} U_{k_C,k_R} R_{k_R,i}$$
- With CUR method, C & R matrices are selected into A
- U is the matrix of coefficients which minimizes the global error criterion
- CUR retains important times and locations

CUR and Linear Regression



- Each cell of A gives one line in X
- A gives the vector of response Y
- U gives the vector of coefficients α
- L_2 regression : $\alpha = ({}^tXX)^{-1}\,{}^tXY$

L_2 REGRESSION

$$Min \sum_{t=1,i=1}^{t=Max_t,i=Max_i} \epsilon_{t,i}^2$$

Under

$$\forall t, i \quad Y_{t,i} = X_{t,i}\alpha + \epsilon_{t,i}$$

Solution:

$$\alpha = ({}^t XX)^{-1} {}^t XY$$

L_1 REGRESSION

$$Min \sum_{t=1,i=1}^{t=Max_t,i=Max_i} \epsilon_{t,i}^+ + \epsilon_{t,i}^-$$

Under, $\forall t, i$

$$\epsilon_{t,i}^+ \geqslant 0$$

$$\epsilon_{t,i}^- \geqslant 0$$

$$Y_{t,i} - \sum_k X_{t,i,k}\alpha_k = \epsilon_{t,i}^+ - \epsilon_{t,i}^-$$

Solution:

Linear Programming

L_∞ REGRESSION

$$Min \quad \epsilon_{max}$$

Under, $\forall t, i$

$$\epsilon_{max} \geqslant 0$$

$$-\epsilon_{max} \leqslant Y_{t,i} - \sum_k X_{t,i,k}\alpha_k$$

$$Y_{t,i} - \sum_k X_{t,i,k}\alpha_k \leqslant \epsilon_{max}$$

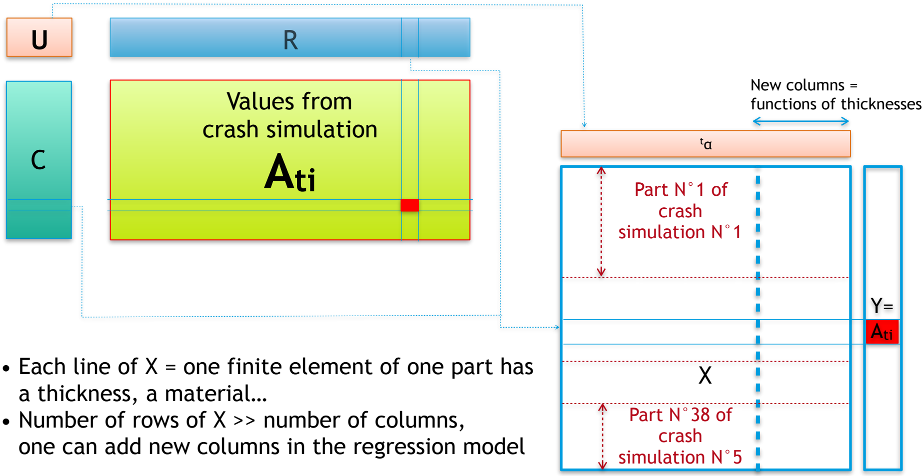
Solution:

Linear Programming

04 REGRESSION CUR

- ReCUR, how to interpolate?
- ReCUR, small example
- ReCUR, big example
- Why Linear Programming?
- Algorithm (simplified)

ReCUR, how to interpolate?



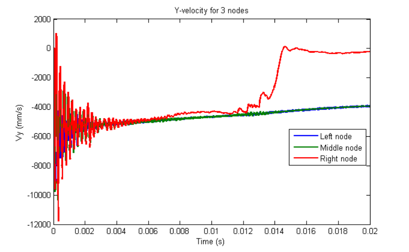
- Each line of X = one finite element of one part has a thickness, a material…
- Number of rows of X >> number of columns, one can add new columns in the regression model

ReCUR, small example

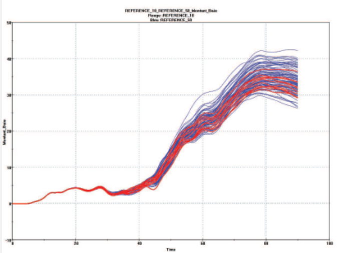
ReCUR, big example

Why Linear Programming?

Numerical convergence



Numerical dispersion



- Numerical convergence & dispersion $\Longrightarrow$ robust regression
- Size of the matrices can vary from one crash simulation to another (if a part is present in one simulation but not in the other)
- L_∞ allows to detect bad cells
- Possibility to add constraints, like total volume for the parts during the crash

$List = \{\}$
Do N times
    Solve LP

$$\min \quad \gamma.\epsilon_{max} + \sum_j \beta_j^+ + \beta_j^-$$

$$\forall t, i \notin List \quad -\epsilon_{max} \leq A_{ti} - \sum_j (\beta_j^+ - \beta_j^-).X_{tij} \leq \epsilon_{max}$$

$$\forall j \quad \beta_j^+ \geq 0, \quad \beta_j^- \geq 0$$

    If (max absolute error over $A_{ti} \notin List \leq \epsilon_{target}$)
    Then
        END
    Else
        Add the K biggest error cells to *List*
    Endif
Done

05 CONCLUSION

Linear Programming service of Statistics

- Good tools for fast prototyping
- Accept big problems
- Parallel computation up to 32 cores without effort
- Many good solvers available, easily interchangeable, even open source

Perspectives:

- Continue to add constraints
- Add rigid body movements and other functionalities
- Try to apply to other problems like CFD, NVH...
- At the end, if possible, go back to L_2 with LASSO?

Thank you for your attention.
Q&A

SystemX
INSTITUT DE RECHERCHE TECHNOLOGIQUE

Linear Programming service of Statistics

Reduction of an automotive crash model

March 24, 2016

Thuy Vuong, CIFRE RENAULT

Yves Le Guennec, IRT SystemX

Yves Tourbier, RENAULT & IRT SystemX

Investissements d'Avenir

ANR · îledeFrance · CAPS · Essonne Le Conseil Général · Campus Paris Saclay Fondation de Coopération Scientifique · Systematic Paris Region Systems & ICT Cluster · advancity Ville & Mobilité Durables · ASTech Paris Region · mov'eo