STOCHASTIC SIMULATION FOR CRASHWORTHINESS

M. Bulik†, M. Liefvendahl†, R. Stocki‡, C. Wauquiez†

† Mecalog, 2 Rue de la Renaissance 92160 Antony, France,
‡ Institute of Fundamental Technological Research, Polish Academy of Sciences,
Swietokrzyska 21, 00-049 Warsaw, Poland

Abstract

This paper describes the M-Xplore extension of the Radioss software. The module contains facilities for the exploration of a parameterized finite element model design space. It supports facilities for interactive choice of variables and responses, definition of a sampling on a design space, automatic submission of the computations, and post-processing of the results. The computations are run automatically, either locally or in ASP-mode, i.e. as a client of a high performance computing server. The software is described first in general, then we illustrate its exploration possibilities in terms of a model problem and a more typical application problem of crash simulation.

1 Introduction

Automotive crashworthiness is a major area of application of nonlinear finite element analysis. The rapidly decreasing cost of computers and the robustness of explicit codes, such as Radioss [1], have revolutionized design in the past decade. Complete crash simulations are performed to evaluate early design concepts. Computations are also used to investigate the details of the final design such as internal paddings, material selection or 'tuning' of parameters.

The advances in hardware and software also open up the possibilities to do statistical analysis and optimization of the design for crashworthiness. The goal of M-Xplore is to provide tools to make such investigations as efficient and simple as possible.

A crash simulation is very computationally expensive, furthermore, for the purpose of optimization or statistics, many such simulations must be done. For this reason such investigations are on the limit of what is possible with the current state of the art technology. One of the main features of the module is the built-in possibility to submit the computations to a high performance computer center. This means that the definition of the task and the post-processing are performed locally on a workstation, while the FE crash analysis is launched on a supercomputer. One of the first applications of M-Xplore was a statistical investigation for a complete car model with 200 samples, i.e. 200 full crash simulations. The simulations for this particular example were launched at Fujitsu Technical Computing Center in Rungis near Paris. The computational time for this task was in the order of weeks.

In this paper we describe the tools for statistical analysis which are implemented in M-Xplore. First we do this generally by going through the steps concerned and the corresponding GUI-features. Then we present two case studies where the techniques are applied.

2 Presentation of the software

In this section we discuss the different steps involved in a statistical investigation of a FE model using M-Xplore. The presentation contains an overview both of theoretical concepts and the actual implementation and GUI.

In section 2.1 we describe how M-Xplore is integrated in the standard FE preprocessor to allow the user to define the variables and responses of a model. Section 2.2 describes the possibilities to define a sampling-type task:

distributions for the input variables, number of samples, and choice of a sampling technique. The algorithm for finding so-called optimal Latin hypercubes is described in section 2.3. This technique is crucial for the stochastic analysis of crash problems when the simulations cost is very high. In section 2.4 we describe how the computations are launched. Finally, in section 2.5, the post-processing facilities are described. In addition to the standard statistical results e.g. statistical moments of the outputs, confidence intervals (or corridors if the time dependence is studied), histograms or scatter plots more advanced topics are mentioned such as analysis of correlation matrices, clustering, and principle component analysis.

2.1 Model Parametrization

The aim of this step is to create a list of variables and responses. In this paper we consistently use the terminology variable for input parameters to the simulation, i.e. parameters describing the model or the initial conditions. The term response is used for any output of the simulation.

M-Xplore is completely integrated with the M-Crash preprocessor. In Figure 1 one can see a screen-shot of a typical session.

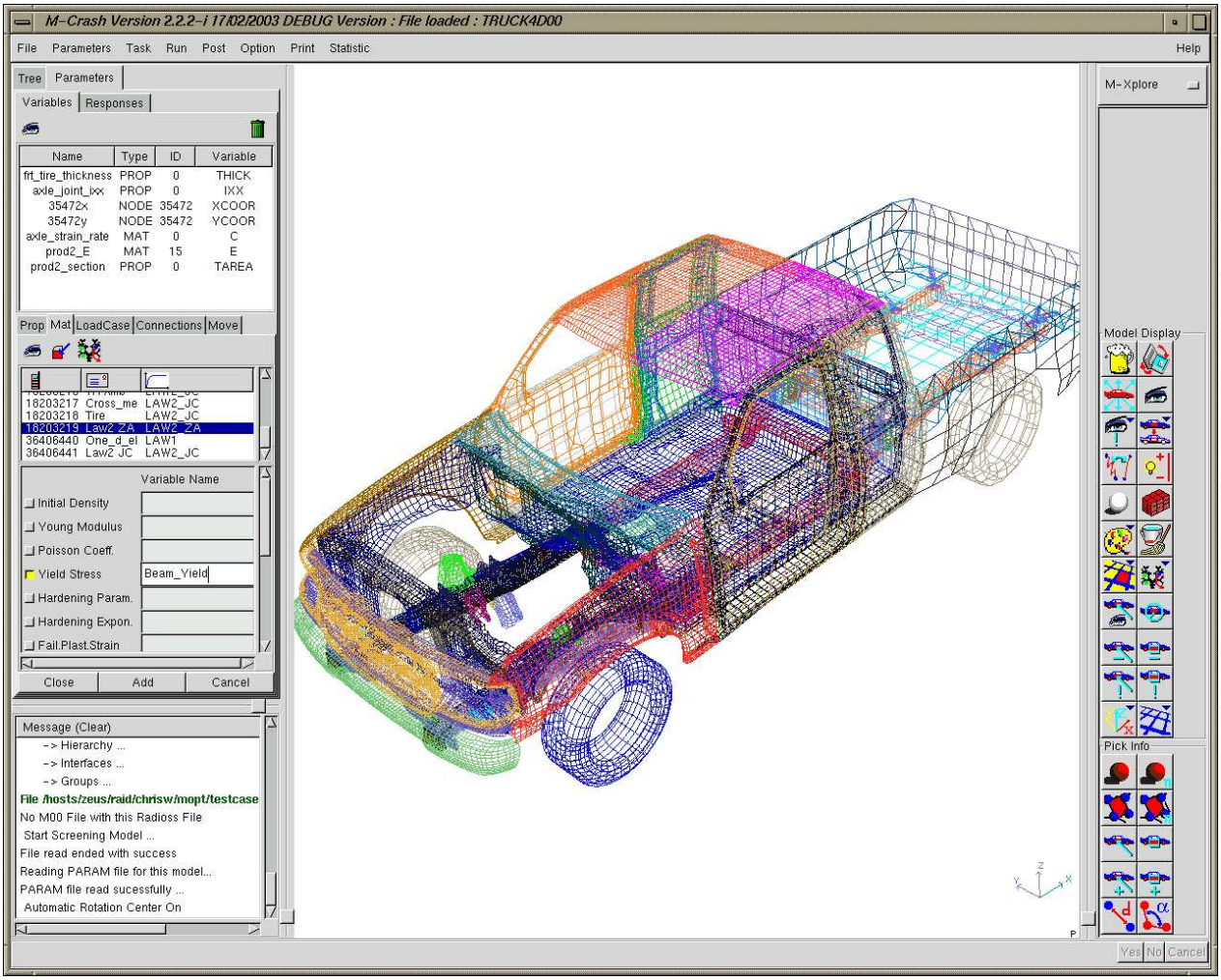

Figure 1: GUI for the choice of variables and responses.

Variables
Almost any attribute of the finite elements model can be defined as variables:

- Geometric properties (e.g. thickness of a shell part, moments of inertia of a beam part, etc.).
- Materials (e.g. Young modulus, yield stress, hardening coefficient, etc.).
- Connections (e.g. spotwelds, glue, welding line, etc.).
- Load cases, initial and boundary conditions, (e.g. added mass, initial velocities, rigid walls, contact interfaces, imposed displacements, imposed velocities, concentrated loads, monitored volumes, etc.).

More advanced variables (meta parameters), controlling many model data, are also available. Scaling, translation and rotation can be applied to a set of parts or a set of nodes. For example, rotations applied to the rigid wall allow to consider random variations of the impact angle in car crash simulations.

Responses
The available responses include e.g. energies, displacements, velocities, accelerations, stresses, curvatures, elongations, forces, etc. They are actually all the time history variables that can be observed globally on the model, or on parts, nodes, elements, sections, accelerometers, rigid walls, and monitored volumes.

2.2 Task Definition

The next step is to define a task (a statistical investigation), see Figure 2 for a screen-shot of the task definition window. The user chooses which variables are active, defines the probability distribution of the active variables, enters the number of samples to generate, and chooses a sampling method.

The following information is specified by the user :

- Which of the parameters and responses are to be included in the study. Choice of ”Active”/”Not active” for each.
- Choice of distribution law (in the current version only uniform or normal) and parameters of the distribution, e.g. mean and standard deviation.
- Number of samples, or number of levels for the full factorial type.
- Sampling method. See below for a discussion of available choices.

From this window it is also possible to generate the samples when the above information has been given.

2.3 Sampling methods

A Monte Carlo simulation is the basic choice in this type of statistical investigation. However there are sampling methods which have better statistical properties. Included are MC (**Monte Carlo**), RLH (**Random Latin Hypercube**) and OLH (**Optimal Latin Hypercube**) which we describe below. In addition to these sampling techniques the software offers also the 2-level **Full Factorial Design**. This technique of exploring the design space is still often used in practice. The information it provides are main effects and interactions of the variables on responses.

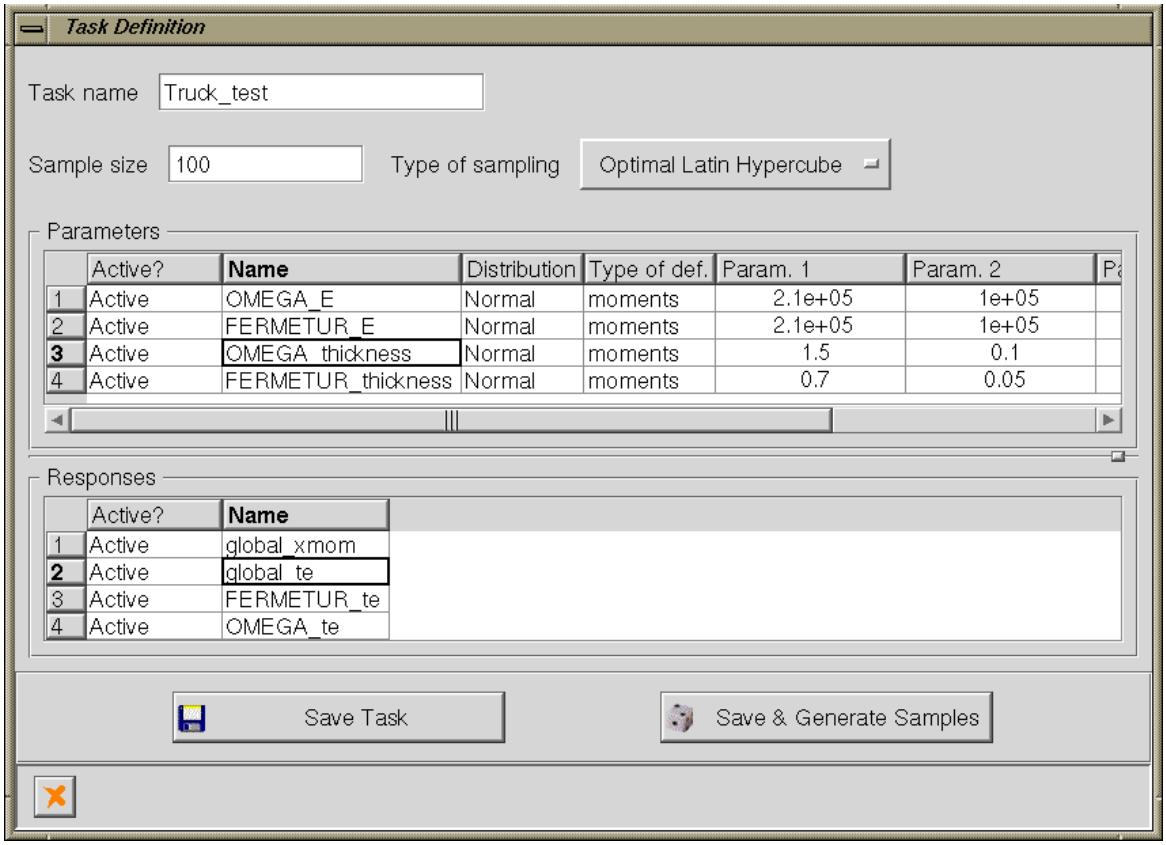

Figure 2: The task definition window. In the two lists, for parameters and responses, the user specifies the properties according to the possibilities described in the text.

Both RLH and OLH are based on the idea of descriptive sampling, which we now describe in the case of N samples and p (continuous) random variables X_i with cumulative distribution functions F_i. For each variable X_i, we determine points $\{x_i^{(k)}\}_{k=1}^{M_i}$ by solving the equations

$$F_i(x_i^{(k)}) = \frac{2k-1}{2M_i}, \qquad k = 1, \ldots, M_i.$$

Informally, this means that the real axis is divided into M_i intervals with equal probability (according to F_i), then $x_i^{(k)}$ is chosen as the probabilistic mean point of the k-th interval. Descriptive sampling means that we restrict our choice of samples to the points

$$\mathbf{x} = (x_1^{(k_1)}, \ldots, x_p^{(k_p)}), \qquad 1 \le k_i \le M_i. \tag{1}$$

What remains is thus to choose our N samples from this finite set (with $\Pi_{i=1}^{p} M_i$ elements). Some authors restrict the term descriptive sampling by requiring that each point of the type (1) can occur at most once in the set of samples.

The second (discrete) step of choosing samples of the form (1) can be formulated in the following way. The goal is to obtain a $N \times p$ matrix $\boldsymbol{Y}$ with elements y_{ij}. Each row of this matrix gives the coordinates of one sample. We can now indicate the choice of samples with an $N \times p$ matrix $\boldsymbol{A}$ with integer elements a_{ij} which is connected with the sample matrix by the relation

$$y_{ij} = x_j^{(a_{ij})}.$$

Obviously it is required that a_{ij} is an integer in the interval $[1, M_j]$.

We now turn to Latin hypercubes (LHs). A LH is given by an matrix $\boldsymbol{A}$ where each column is a permutation of the numbers 1 to N. Thus for descriptive sampling for LH we must choose $M_j = N$ above. This requirement implies that two different samples have all their coordinates different. Also each number $x_i^{(k)}$ occurs once in the $\boldsymbol{Y}$ matrix with the samples. The construction of (the matrix $\boldsymbol{A}$ describing) a random LH is elementary and we do not describe this step. In M-Xplore we have given the name Random Latin Hypercube to the sampling resulting from this algorithm. The reason for this name is that there is a random step in the LH construction, different samplings give different LHs.

Now we will describe the optimal (or optimized) Latin hypercube sample generation. The most important step is the determination of a (LH) matrix $\boldsymbol{A}$ with as good 'separation' properties as possible. By this we mean that we want to avoid clustering of the samples as much as possible. The starting point for the OLH-algorithm is a RLH, which then is optimized by an algorithm called the "columnwise pairwise"(CP)-method. This is described in detail in [2] and [3].

The criteria with respect to which the LH is optimized is the following. Recall that the i-th row in $\boldsymbol{A}$ gives the coordinates of the i-th sample (in $\mathbb{R}^p$). The distance between sample i and sample j, ξ_{ij} is thus given by

$$\xi_{ij}^2 = \sum_{k=1}^{p} (a_{ik} - a_{jk})^2.$$

Now we define the criterion as (see [4])

$$d(\boldsymbol{A}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{\xi_{ij}^2}, \tag{2}$$

and the optimization problem consists of finding an LH (represented by $\boldsymbol{A}$) which minimizes $d(\boldsymbol{A})$. As a physical analogy, if we consider the samples as electrically charged particles, then the problem corresponds to a minimization of the sum of the absolute values of the repulsive forces. From the point of view of this analogy, it would be natural with the exponent 1 (instead of 2) in the denominator of the terms in (2). However, with the power 2 the computation of the square root for each term is avoided.

The CP-method is computationally expensive because it searches a very large number of LHs during the optimization process. Its execution time is approximately proportional to pN^5. In table 1 we give the results of some experiments investigating the execution time. The numbers given in the table were taken from a particular run (i.e. they are not mean values). The variations of the execution time due to the random starting point are usually in the range 10-30%. The derivation of the complexity of the OLH is found in [3]. By the term complexity we mean the estimate of how the execution time depend asymptotically on the size of the problem i.e. the numbers N and p.

To illustrate the benefits with the OLH-algorithm we compare our three sampling methods MC, RLH and OLH for the following model problem.

We study the function

$$Y = f(X_1, X_2) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2,$$

of two random variables. This function is strongly nonlinear and has a 'valley' in the shape of the parabola $X_2 = X_1^2$. It is sometimes referred to as the Rosenbrock function. Now we take X_1 and X_2 to be uniformly distributed in the interval $[0, 2]$. We estimate the mean value of Y by sampling X_1 and X_2 and calculating the corresponding Y-values and their mean. In this simple case the mean can be analytically computed and is 187. In table 2 we compare the accuracy of this estimate of the mean for our three sampling methods. From these results we can clearly rank the methods with OLH as the best and MC as the worst.

N\p	3	5	7	10
50	0.86	3.4	12	21
60	2.1	11	29	37
70	6.8	21	51	110
80	9.9	37	110	210
90	11	92	170	440
100	24	110	220	650

Table 1: Execution time in seconds for the OLH-algorithm with different values of N (number of samples) and p (number of variables). These calculations were performed on a PC with Pentium IV (2.4GHz) processor.

N	OLH	RLH	MC
10	9.1	20.7	37.8
20	3.5	14.3	17.8
50	1.5	10.2	12.5
100	1.1	6.6	9.4
200	0.6	5.6	6.0
500	-	2.4	4.8
1000	-	1.5	3.2
2000	-	1.1	2.2
5000	-	0.7	1.2

Table 2: The average of the error percentage for the different sampling methods for different sample sizes. OLH with N greater then 200 have not been computed because of the long computational time.

It is evident from this example that if one can afford the OLH-sampling then it is the best choice. In the area of crash simulation it is very expensive to perform the calculation for one sample. Thus one is forced to economize with the samples and this is a typical situation where OLH is recommended.

2.4 Job submission and monitoring

The FE simulations which are the most computationally expensive part of the analysis can be performed on the local computer or on a remote server. For the latter purpose, a client/server architecture for the execution and monitoring of computations has been developed.

The interface between the client and server is transparent, the user never quits M-Xplore when submitting and monitoring the computation. All transmission of data is encrypted and the execution is performed in a confined environment to ensure complete security.

The monitoring of the progress of the computations is possible on two levels. On the ‘coarse’ level the user can get information on the number of completed, running and waiting simulations. To be able to discover numerical or physical instabilities in the model, it is also possible to get more detailed information, both on completed and running computations. An analysis of this information allows the user to change the parameters of the waiting computations.

2.5 Statistical post processing

The software provides a number of post processing tools to explore the statistical properties of random variables and responses.

For each sample, the user has access to the entire time history of the chosen responses (In the case when the computations are run on a server, only this data is communicated and not the entire output of the FE computation). Scalar values can be obtained by the following operations: time average, maximum or minimum value over time, time integral, value at specified time instant. Some other values that are important mainly in crash test simulations like the so-called Head Injury Criterion (HIC) can also be computed (see e.g. [5]).

In M-Xplore the following statistical information can be computed for all (input and output) random variables:

- Statistical moments and intervals of confidence
- Histograms and scatter plots
- Confidence corridor plots
- Correlation coefficients
- Principal component analysis

The confidence intervals are determined using either the cumulative frequency curve or assuming a distribution for the responses according to t-distribution. Still it is realized that a proper parametric study should be performed to determine the probability law, which is planned in further developments of the software. Plotting the histograms of a given response (see Figure 3) can help in verifying the assumptions about the distribution type.

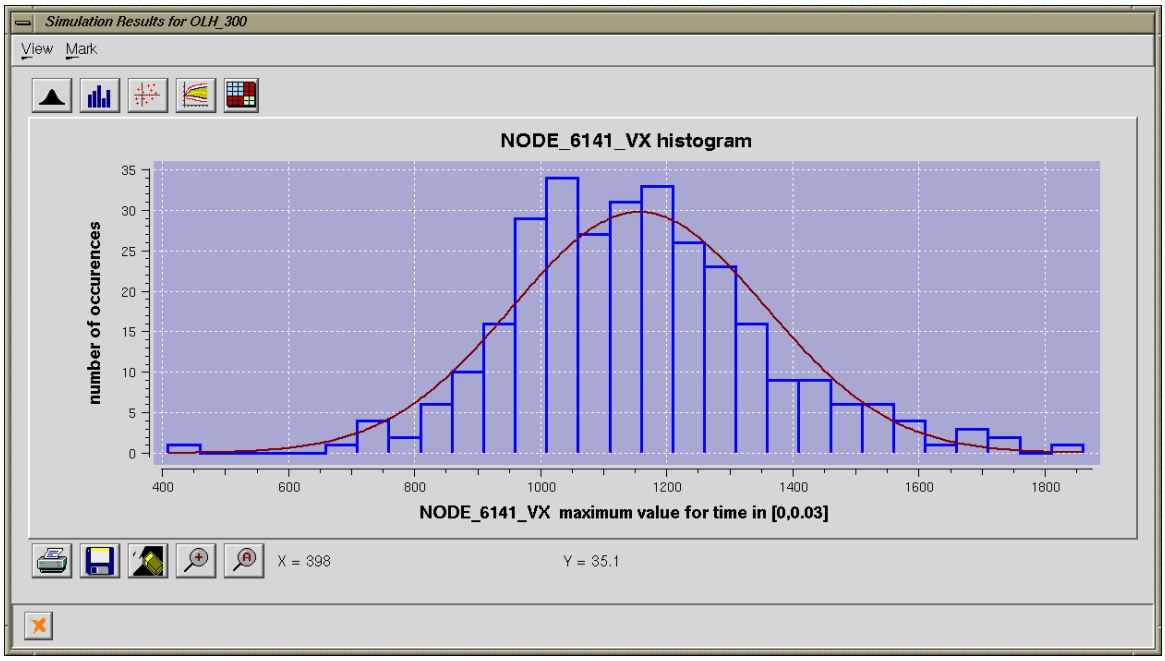

Figure 3: Histogram of a response with superimposed normal probability density function.

The confidence corridor plots, see Figure 4, provide very useful information about the changes of a scatter of results in time. By analyzing the representative responses it may be observed that the abrupt change of the

corridor's width (increase in a scatter of results) often indicates the existence of various post-critical behaviors of a structure.

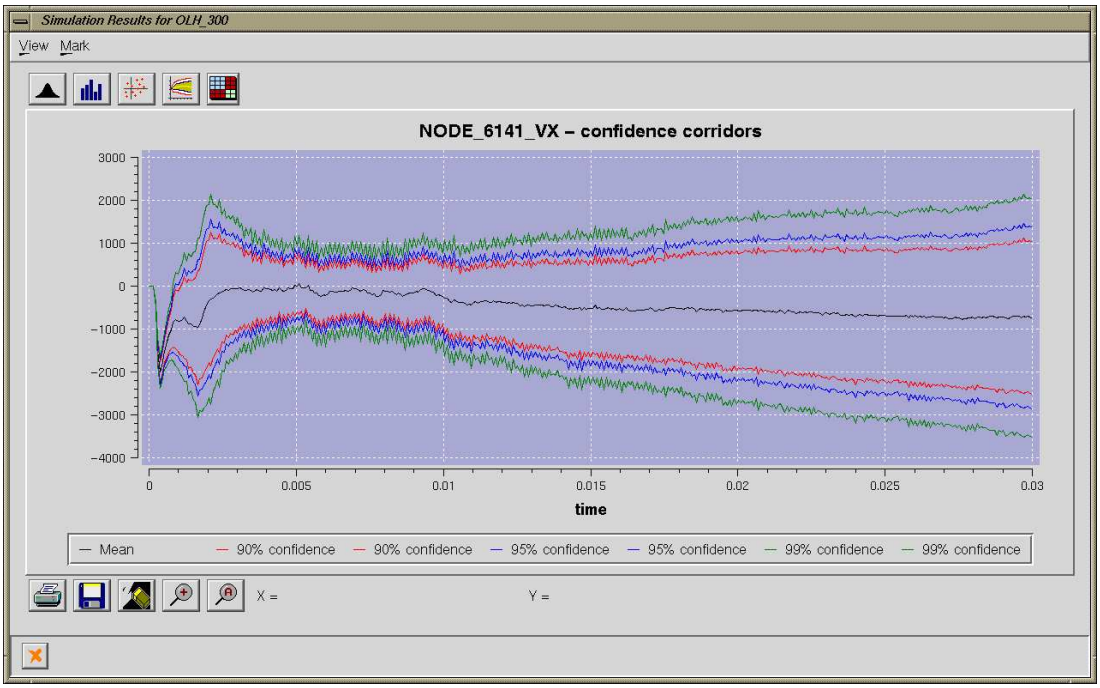

Figure 4: Confidence corridor plot of a nodal velocity. The sudden change of the corridor's width may indicate the 'separation' of significantly different deformation patterns corresponding to various failure modes.

Similar information can also also be found be examining scatter plots. Clustering of points in a scatter plot should always be the subject of careful analysis. Points situated far from the main cloud may be of particular importance for the proper understanding of the structural behavior. See Figure 5 for the GUI for scatter plots.

The functionality is provided for building a data matrix from the whole set of variables and time responses, and computing the corresponding correlation matrix. To understand how changes in the different variables affect the behavior of the model, it is useful to study the data matrix for all variables and one, suitably chosen, response.

More advanced statistical post-processing is offered with the Principal Component Analysis (PCA). For investigations involving many descriptors (responses and/or variables), it is often useful to simplify the analysis by considering a smaller number of linear combinations of the original descriptors. By doing this we want to summarize, in a few dimensions, most of the variability of a covariance matrix of a large number of descriptors. Because of the varying physical nature of the considered descriptors, the implemented PCA operates on the correlation matrix $\boldsymbol{R}$ which is the covariance matrix of standardized descriptors. The principal component analysis is based on the solution of the following eigenproblem:

$$(\boldsymbol{R} - \lambda_i \boldsymbol{I})\boldsymbol{u}_i = \boldsymbol{0}\,, \qquad i = 1, \ldots, p\,, \tag{3}$$

where p is the number of descriptors and λ_i and $\boldsymbol{u}_i$, are the eigenvalues and normalized eigenvectors, respectively. We assume the eigenvalues are sorted according to size with the largest first (all eigenvalues are real and positive). The line through the origin directed along $\boldsymbol{u}_1$ is called the first principal axis. The corresponding line directed along $\boldsymbol{u}_2$ is called the second principal axis etc. The matrix $\boldsymbol{U}$, the columns of which are the normalized eigenvectors, allows to compute the values of principal components for to the p-dimensional data

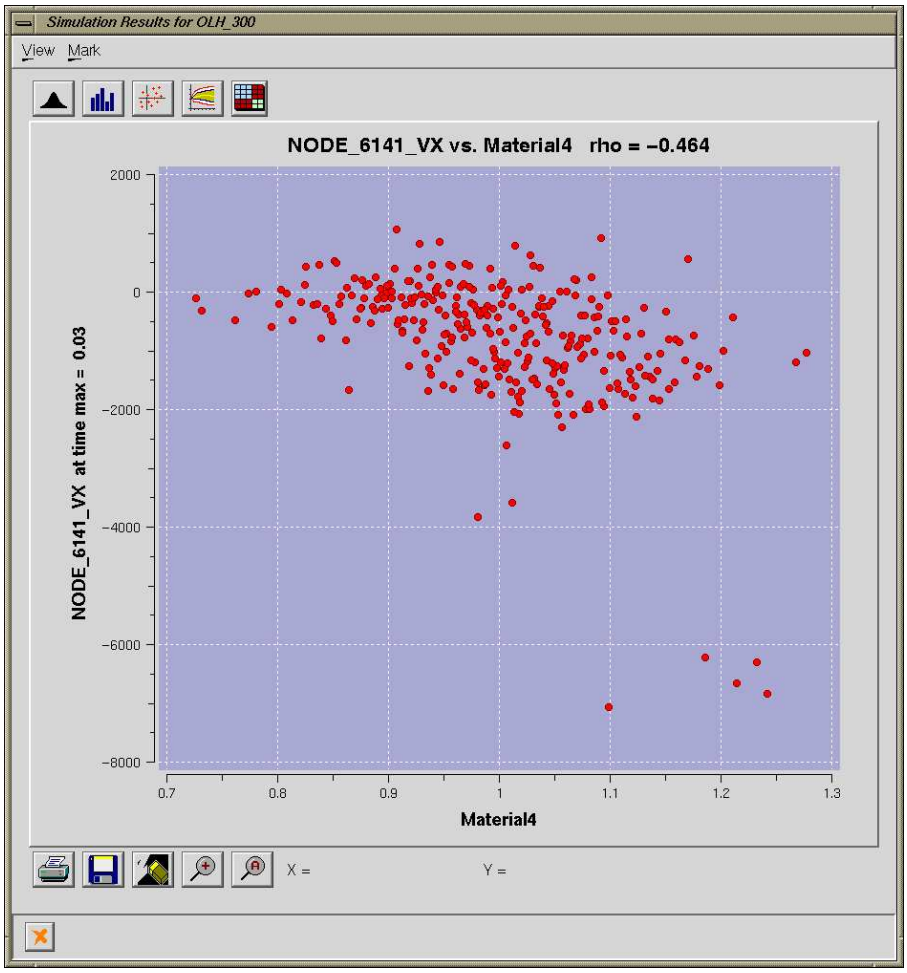

Figure 5: Scatter plot. The small cluster of five points in the lower right corner is clearly separated from the remaining samples and corresponds to different failure mode.

points by means of transformation

$$\boldsymbol{F} = \begin{bmatrix} \frac{x_{11}-\bar{x}_1}{\sigma_{x_1}} & \frac{x_{12}-\bar{x}_2}{\sigma_{x_2}} & \dots & \frac{x_{1p}-\bar{x}_p}{\sigma_{x_p}} \\ \frac{x_{21}-\bar{x}_1}{\sigma_{x_1}} & \frac{x_{22}-\bar{x}_2}{\sigma_{x_2}} & \dots & \frac{x_{2p}-\bar{x}_p}{\sigma_{x_p}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_{N1}-\bar{x}_1}{\sigma_{x_1}} & \frac{x_{N2}-\bar{x}_2}{\sigma_{x_2}} & \dots & \frac{x_{Np}-\bar{x}_p}{\sigma_{x_p}} \end{bmatrix} \boldsymbol{U}\,, \tag{4}$$

where N is the number of data points (samples), $\boldsymbol{F}$ is the $N \times p$ matrix containing the coordinates of data points in the space of principal components and $\bar{x}_i$ and σ_{x_i} are the mean values and standard deviations of the descriptors, respectively. Correlation of the i-th descriptor with j-th principal component is given by the formula

$$u_{ij}\sqrt{\lambda_j}\,, \tag{5}$$

where λ_j is the eigenvalue corresponding to the j-th principal component and u_{ij} is the i-th component of the corresponding eigenvector. A very convenient way of presenting these quantities is by the so-called correlation circle (see Figure 6). In this plot descriptors are represented by points which coordinates are equal to their corresponding correlations with the selected principal components. The correlation circle shown in Figure 6 was made for the first two principal components explaining almost 80% of the total variability (check the value of cumulative variance in the table under the plot).

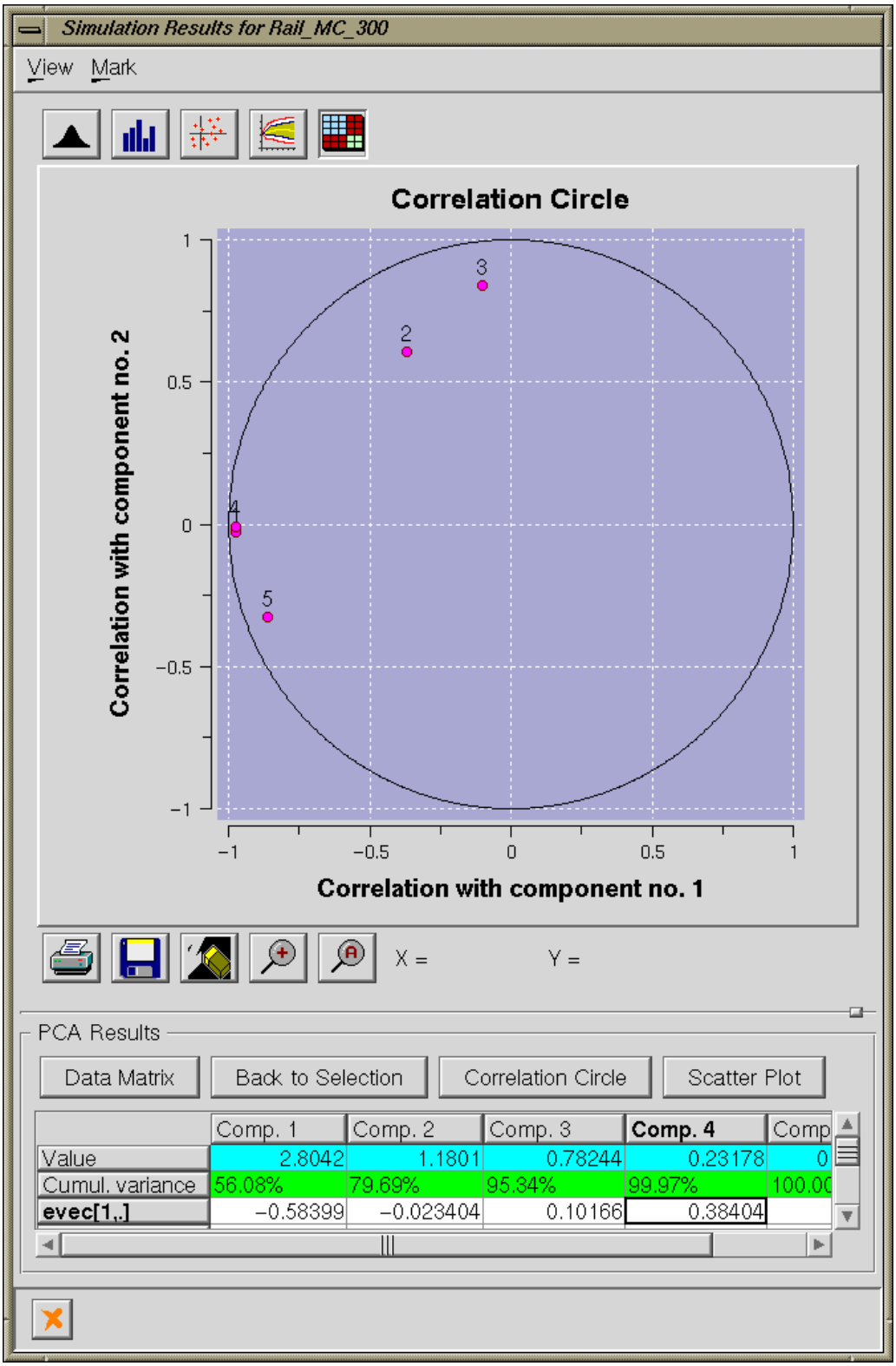

Figure 6: Correlation circle. The points represent selected descriptors (variables and/or responses). Their coordinates are equal to the respective correlation coefficients of descriptors and the first two principal components.

Having identified a group of samples leading to a certain structural behavior it is interesting to study the corresponding domain of design space. In other words the next step is to obtain a simple description of the set of designs which lead to the behavior in question.

In the case of one or two design variables a simple plot of the samples corresponding to the two behaviors (the behavior under study and the rest, the 'complementary' behavior) gives practically all information. With more than two variables, however, the investigation can be extremely complicated. In M-Xplore there is implemented an algorithm to find a separating hyperplane, or a hyperplane which separates the two clusters as 'good' as possible (precise statement below).

Even when the boundary between the two behaviors is curved the separation hyperplane gives useful information. It can also be seen as a first approximation to the shape of the real boundary. Furthermore it is, of course, convenient to have an explicit formula for the boundary.

Now we turn to the formulation of the optimization problem which determine the separation hyperplane. A hyperplane is described by the equation

$$\boldsymbol{n} \cdot \boldsymbol{x} = d.$$

Here $\boldsymbol{n}$ denotes the unit normal vector of the hyperplane and d denotes the signed orthogonal distance from the origin to the hyperplane. The problem is now to find $\boldsymbol{n}$ and d such that the hyperplane separates two clusters $\{\boldsymbol{y}_i\}_{i=1}^{N_y}$ and $\{\boldsymbol{z}_i\}_{i=1}^{N_z}$, i.e. such that all $\boldsymbol{y_i}$ are on the side to which the normal points and all $\boldsymbol{z_i}$ are on the other. We formulate this as an optimization problem in the following way. First we introduce one additional unknown r, this will turn out to be the shortest distance between any point and the hyperplane.

The optimization problem is as follows:

$$\begin{aligned}
\text{find} \quad & n_1, \ldots, n_p, d, r \\
\text{that maximize} \quad & r && (6)\\
\text{subject to} \quad & \sum_{i=1}^{p} n_i^2 = 1 && (7)\\
& \boldsymbol{n}\boldsymbol{y}_i - d \geq r\,, \qquad i = 1, \ldots, N_y\,, && (8)\\
& \boldsymbol{n}\boldsymbol{z}_i - d \leq -r\,, \qquad i = 1, \ldots, N_z\,, && (9)\\
& -1 \leq n_i \leq 1\,, \qquad i = 1, \ldots, p\,, && (10)\\
& -\delta \leq d \leq \delta\,, && (11)\\
& 0 \leq r \leq \gamma\,. && (12)
\end{aligned}$$

Here n_i denotes the i-th component of the $\boldsymbol{n}$-vector, δ is chosen as the maximum distance between the origin and a sample and γ is the minimum distance between samples of the two clusters.

To solve the problem (6)-(12) the sequential quadratic programming algorithm NLPQL is used, see [6], . This has proven successful in many studies including the two examples of section 3. In Figure 7 we see a snap shot of the window containing the results of a successful cluster separation calculation. If NLPQL fails to find a solution, often such a failure is due to inaccuracy of the hypothesis, i.e. the boundary between the two clusters is far from a hyperplane. In such a case the algorithm tries to use a quadratic hypersurface to separate clusters.

3 Case studies

In this section we will illustrate some of the features described above in the context of two case studies. The first is a stationary model problem with only one degree of freedom. The second one deals with FE crash simulation of the rear frame of a car.

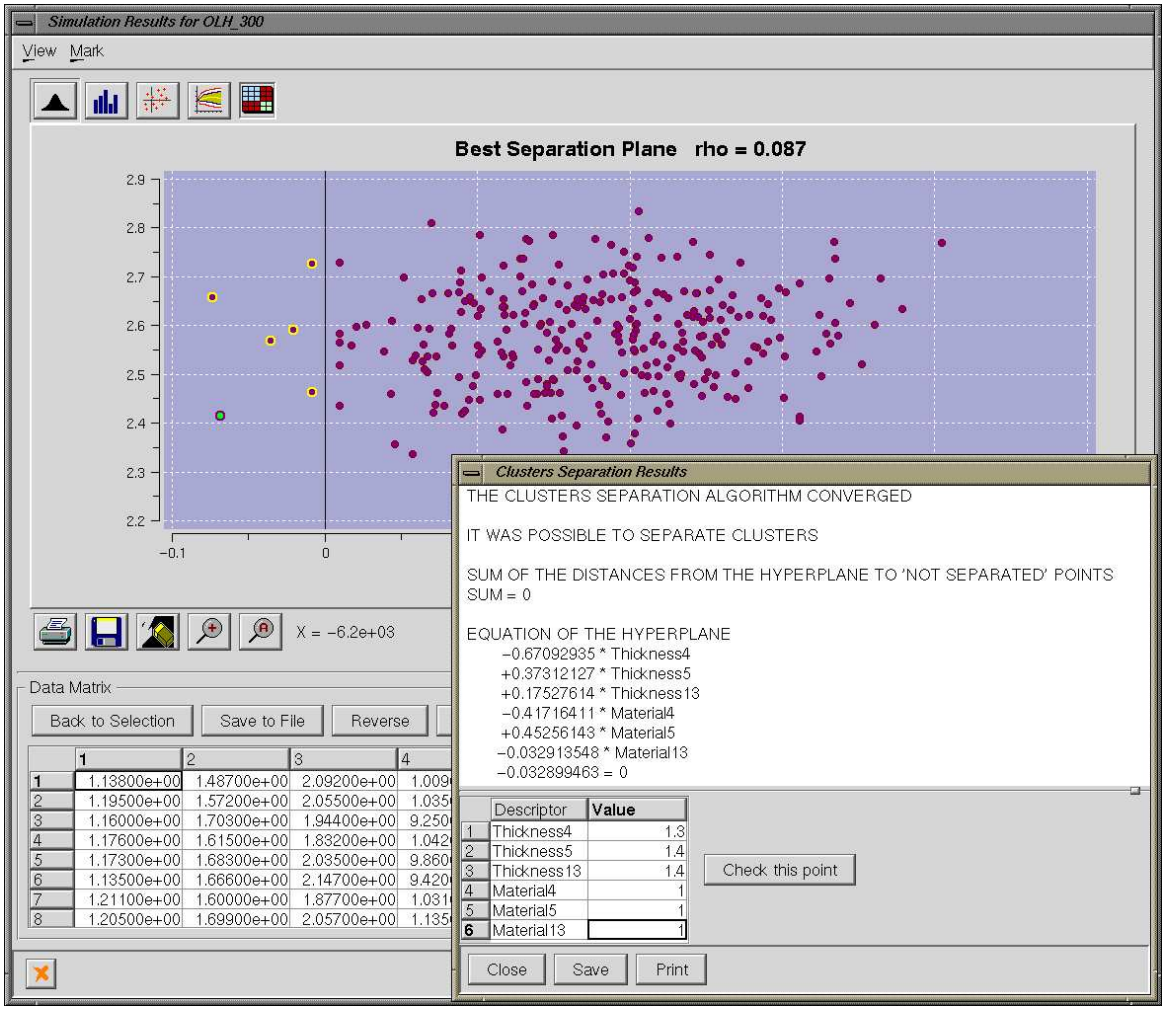

Figure 7: Cluster separation. The small cluster of five points to the left of the straight line (which is the projection of the hyperplane) is separated from the rest of the samples. In the window "Clusters Separation Results" the equation for the hyperplane is shown.

3.1 A model problem

Here we treat a nonlinear stationary model problem with one degree of freedom, taken from [7] pp.2. The geometry of the problem is shown in Figure 8. The nonlinearity results from the geometry, the constitutive relation for the bar and the spring are taken to be linear elastic. The unknown of the problem is w, the vertical displacement of the right node of the bar. The data are

- E: The Young modulus of the bar material.
- A: The cross-sectional area of the bar.
- l: The length of the unloaded bar.
- $z > 0$: The vertical coordinate of the right node of the bar when it is unloaded.
- K_s: The stiffness of the spring.

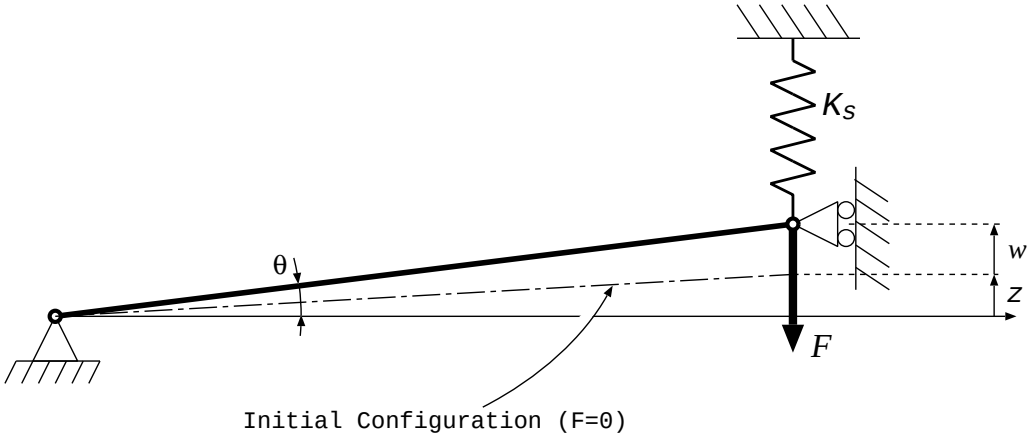

Figure 8: The single bar structure prone to 'snap-through' type instability. The force F shown on the picture has the negative sign.

- F: The vertical force applied to the right node of the bar.

The displacement w can be determined from the following equation (cf. [7]):

$$F = \frac{EA}{l^3}\left(z^2 w + \frac{3}{2}zw^2 + \frac{1}{2}w^3\right) + K_s w. \tag{13}$$

In the derivation of this equation, it is assumed that the angle θ (see Fig. 8) is small which implies $z, w \ll l$. Equation (13) is a third degree polynomial equation for w. For some values of the parameters there is one (unique) real root, for other values of the parameters there are three roots. In the case of three roots we choose the one corresponding to the smallest magnitude of displacement. This situation occurs when the applied force is smaller than the critical force that cause the bar to snap-through to the other equilibrium position. The case with one root of the equation (13) corresponds to the state of the bar after snap-through or when the stiffness of the spring is so big that it prevents this kind of instability.

Now we turn to the description of the task we have performed for this problem. The purpose of our test is to illustrate the clustering caused by the strong nonlinearity. We choose parameter values so that our samples will give solutions of the two types, i.e. with or without snap-through. We take the spring stiffness and the force as random variables

$$\begin{aligned} K_s &\quad \text{uniformly distributed in } [0.9, 1.1](\text{N/mm}) \\ F &\quad \text{uniformly distributed in } [-25, -20](\text{N}). \end{aligned}$$

The rest of the parameters are given the following fixed values

$$EA = 5 \cdot 10^7 \text{N},\ l = 2500\text{mm},\ z = 25\text{mm}.$$

We generate 100 samples with an OLH. These samples are evenly spread over the allowed square in design space $K_S \times F$. We next solve (13) for all the samples. In Figure 9 we show the resulting scatter plot in the K_S-w plane and the F-w plane. The clustering is clearly seen in both planes.

The most important question now is to identify the two regions in design space corresponding to the two clusters. Since our design space is two-dimensional we just check the scatter plot in the K_S-F plane. In Figure 10 we see a picture of the design space where the boundary between the two regions is approximated with a straight line.

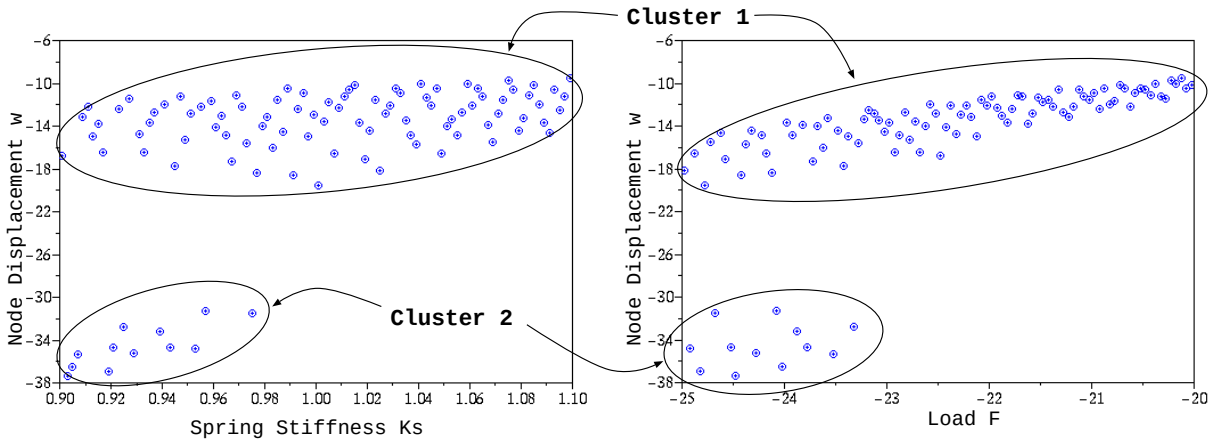

Figure 9: Clustering of the samples for the model problem. The graphics show the projection on the two coordinate planes K_S-w and F-w respectively. Cluster 2 correspond to snap-through.

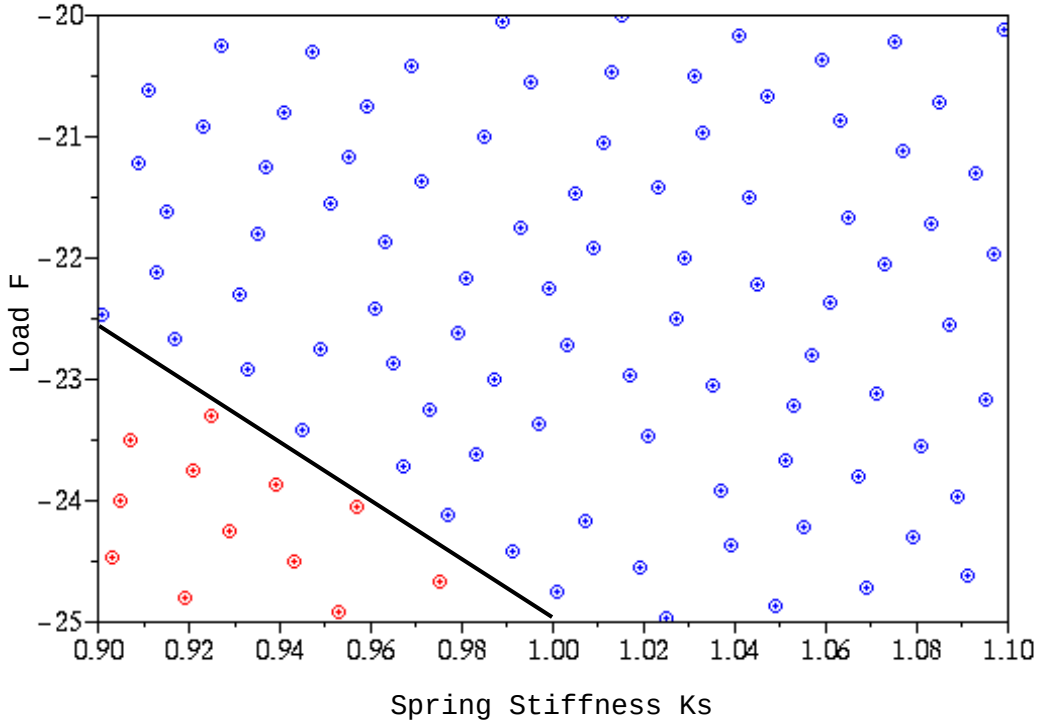

Figure 10: The boundary between the two regions in design space corresponding to the two clusters in Figure 9. For parameter values in the smaller region (lower left corner) we have snap-through behavior and in the other we do not.

This model problem illustrates some nonlinear phenomena in a very simple context. We knew the qualitative picture beforehand with the two possible behaviors.

The following difficulties did not occur in this problem but are present in most of the real life problem and especially the stochastic analysis of crash simulations

- Design space with more than two space dimensions, i.e. more than two variables. In the case of two variables it is very simple to identify interesting regions as we did in Figure 9. For three variables it is still possible, but much more complicated. When the number of variables is four or higher it is, of course, impossible to visualize the complete situation.
- The separation into two behaviors was extremely clear in our model problem. In general there will be many different behaviors and often not so clearly separable. Furthermore the distribution of samples may be such that we have only very few in some of the regions and thus there is not sufficient information.
- We must of course mention the high cost of crash computations. This makes it very expensive, sometimes prohibitively so, to obtain sufficiently many samples to identify the different regions.

3.2 A rear frame crash

In this section we study an example of the rear frame of a car crashing into a wall. This is a Radioss simulated problem. The rear frame finite element model is shown in Figure 11.

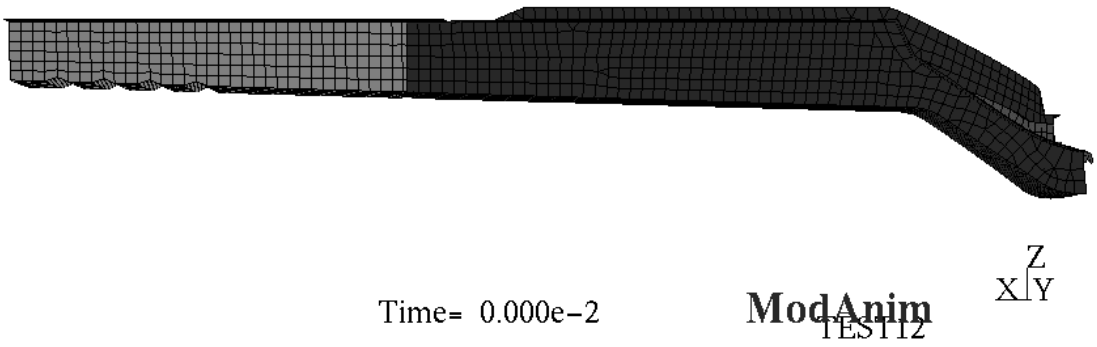

Figure 11: The rear frame finite element model. The rear frame is fixed to the right and a wall moves in from the left with a constant speed of 35 km/hour. The part with lighter color to the left is denoted part 1. The darker part to the right is part 2. Finally inside part 2 there is a smaller part denoted part 3. It is possible to see a few elements of it in lighter color inside the frame to the far right of the picture.

The rear frame consists of, in total, approximately 6000 elements. We have done a statistical investigation with 300 samples, i.e. 300 crash simulations have been performed. The samples were generated by the OLH. The time interval for simulation is from contact of the moving wall and the frame until 0.04 seconds has passed. During this time the wall moves approximately 40 centimeters which corresponds to about 1/3 of the total length of the frame.

In this situation the frame will be deformed by a combination of compression and buckling. In Figure 12 we see the result of two crash simulations with different parameter values for the beam. In the first picture we

see a deformation which is dominated by compression and in the picture to the right the buckling is important. In the statistical test we describe below we investigate the influence of some properties of the beam on the deformation behavior, in particular if buckling occurs. Generally, in the design for crashworthiness of cars, compression is considered the good behavior. In case of compression the energy absorption is high and the transversal displacements are relatively low. On the contrary, buckling results in large and potentially dangerous displacements in the structure and represents poor energy absorption.

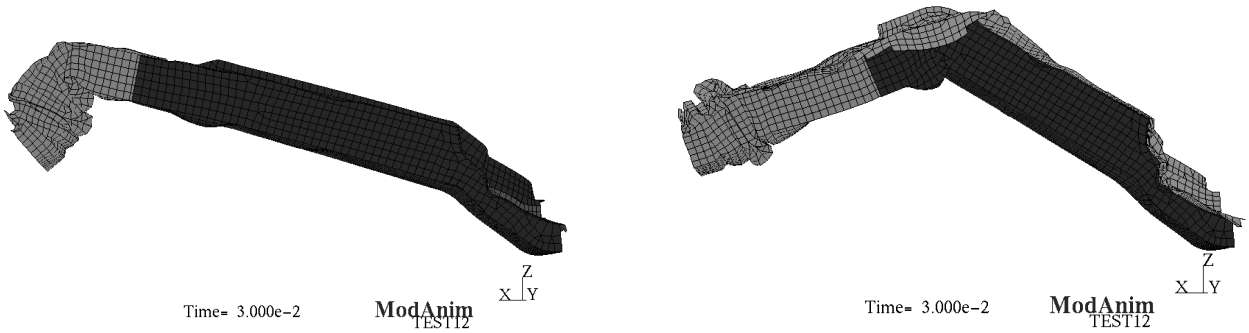

Figure 12: Two types of behavior: compression and buckling. Depending on the parameters the rear frame is deformed in different ways. In the left picture compression is dominant while in the right the frame is buckled in the middle. The rigid wall which moves in from the left and causes the deformation is in contact with the left side of the frame, but it is not shown in the pictures.

To define a statistical test we take six variables in total from the three parts discussed in the caption of Figure 11. These three parts consist of shell elements of an elastic plastic material with a piecewise linear constitutive relation. For each part we take its thickness as a variable and a dimensionless parameter α in the constitutive relation. The stress σ depend on both the plastic strain ε_p and the strain rate $\dot{\varepsilon}$

$$\sigma = \alpha f(\varepsilon_p, \dot{\varepsilon}).$$

Here we find our variable α as a scaling parameter.
We take these six variables to be uniformly distributed in the following intervals

- Thickness, part 1, t_1 in [1.02mm,1.38mm]
- Thickness, part 2, t_2 in [1.36mm,1.84mm]
- Thickness, part 3, t_3 in [1.7mm,2.3mm]
- Material parameter, part 1, α_1 in [0.7,1.3]
- Material parameter, part 2, α_2 in [0.7,1.3]
- Material parameter, part 3, α_3 in [0.7,1.3]

For responses we have made the following choices. We have chosen four nodes. One in the middle of each part and the fourth at the interface of part 1 and 2. For these nodes we take the three components of the displacement and the three components of the velocity as responses. Furthermore, for each part we take the

three components of the average velocity and also the internal energy of each part. Finally we take the global energy and the components of the global velocity as responses. This amounts to a total of 40 responses.

We start the analysis of the results by looking for clustering of the samples, i.e. different physical behaviors. In Figures 13 and 14 we have two scatter plots which display the division of the samples into two clusters, furthermore it is the same clustering found in the two pictures. To discover buckling, it is natural to try a plot such as Figure 13 with the z-displacement of a node in part 2.

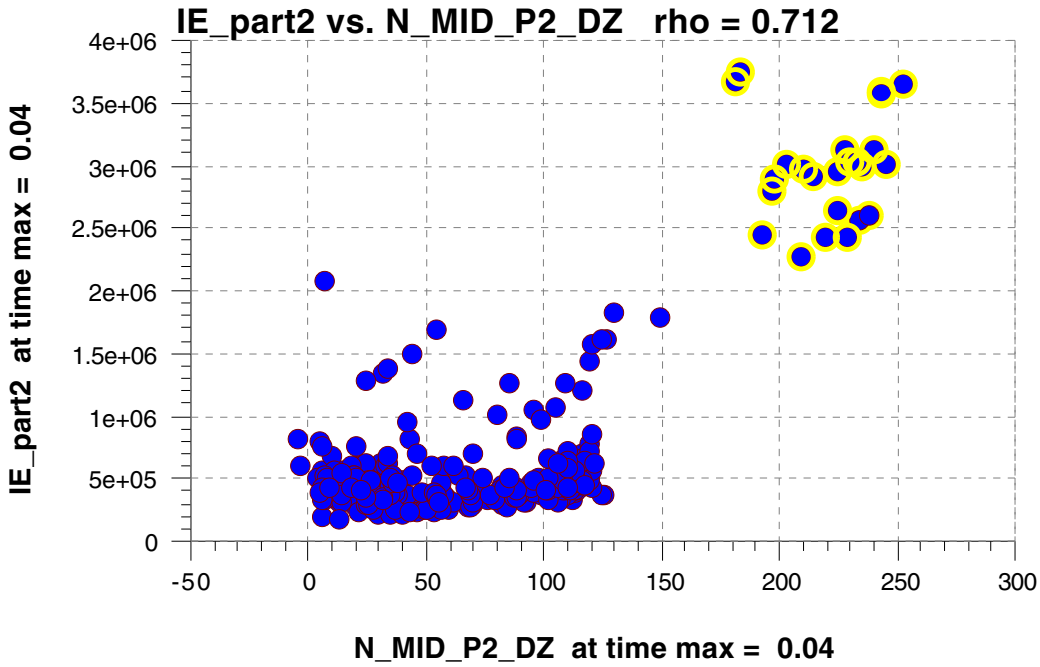

Figure 13: Scatter plot at time 0.04 seconds, i.e. the stopping time of the computation. On the horizontal axis we have the z-displacement of a node in the middle of part 2 measured in millimeters. On the vertical axis is the internal energy of part 2. The cluster in the upper right corner correspond to the buckling behavior.

For this problem it is more difficult to find the regions in design space which correspond to the two cases. The first possibility is to look at the projection of the samples into a two dimensional subspace parallel to the coordinate axes, and use coloring to identify the samples corresponding to the two clusters. With six variables there are 21 different such subspaces. None of them show a separation of the clusters.

The cluster separation algorithm described in section 2.5 however gives the solution to the problem. Thus the following equation for a separating hyperplane is obtained automatically.

$$-0.70t_1 - 0.38\alpha_1 + 0.43t_2 + 0.42\alpha_2 + 0.02t_3 + 0.05\alpha_3 = 0.21 \tag{14}$$

In Figure 15 the solution is illustrated by a scatter plot in a plane orthogonal to the hyperplane of (14).

The hyperplane divides the design space into two parts. On one side we have the designs which buckle and on the other the compression behavior. Using the left hand side in (14) we can evaluate other designs, which were not among our samples, without performing a FE calculation. If the expression in the left hand side is greater than 0.21 then we almost certainly have compression and if it is below 0.21 we have buckling. We have thus obtained one convenient design criterion.

A heuristic way to analyze equation 14 is to first neglect the influence of t_3 and α_3 since their coefficients are small. Then we group the remaining four terms in the left hand side in the following way to get an equation for

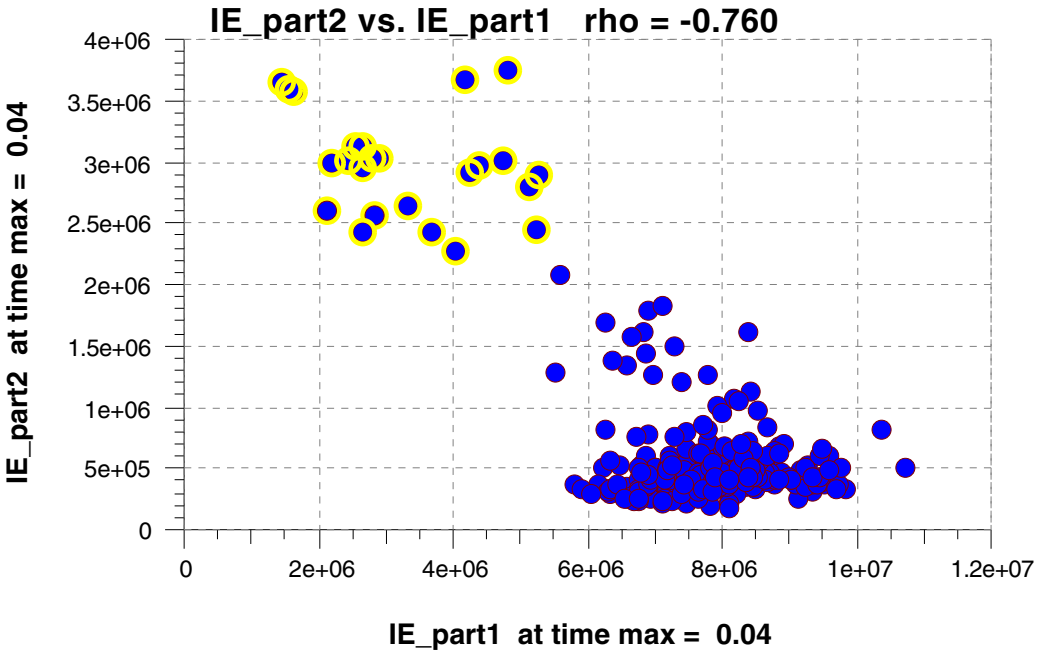

Figure 14: Scatter plot at time 0.04 seconds. Here the coordinates of the samples are the internal energies of part 1 and 2 respectively. The same clustering is visible as in Figure 13. The cluster in the upper left corner correspond to the buckling behavior.

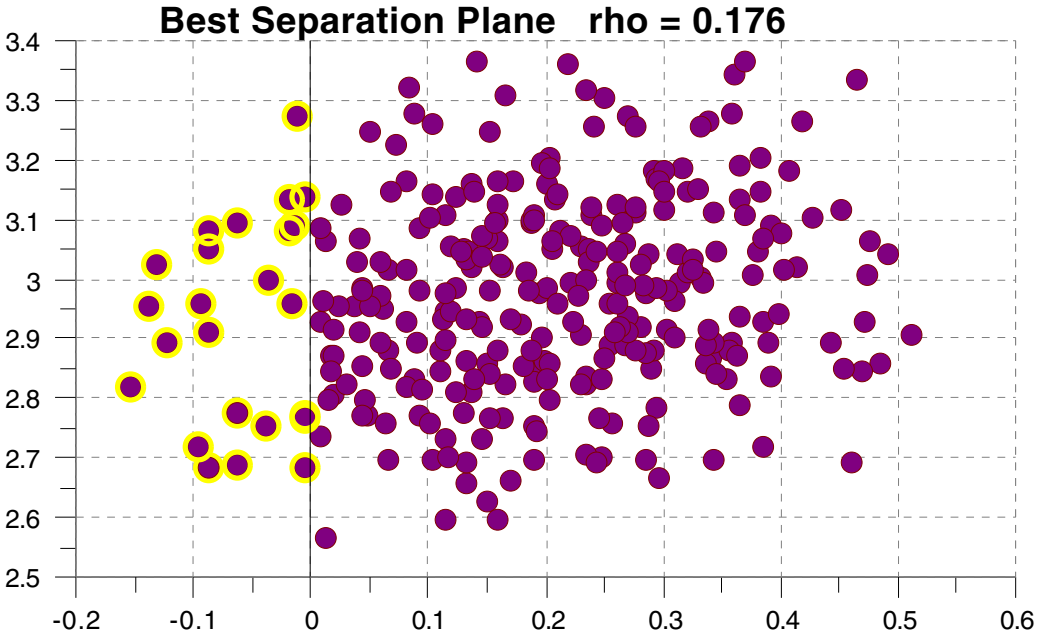

Figure 15: Here we see the separation of the two clusters by the hyperplane of equation (14). The plane of the scatter plot is spanned by the normal vector of the hyperplane and an arbitrarily chosen vector orthogonal to this normal vector. We emphasize that this separation is not possible to see in any plane spanned by the original coordinate axes.

a hyperplane close to the one determined above.

$$-0.4(2t_1 + \alpha_1) + 0.4(t_2 + \alpha_2) = 0.2 \tag{15}$$

Here we have rounded the numbers to one significant digit. Now we observe that the quantity $S_1 = 2t_1 + \alpha_1$ clearly correspond to the 'global' stiffness of part 1. If S_1 is large, then the shell elements are thick and the material is stiffer. In the same way we have $S_2 = t_2 + \alpha_2$ as a measure of the global stiffness of part 2. Inserting S_1 and S_2 in equation (15) and rearranging leads to the expression

$$S_2 = S_1 + 0.5$$

This expression is very easy to interpret physically. If $S_2 > S_1 + 0.5$, then part 2 is so stiff that it does not bend. Instead the deformation begins with a compression of part 1. On the other hand, if the inverse inequality holds, then part 1 is so stiff that we will have bending of part 2 instead of compression of part 1.

The presented approach proved to be useful in many application. However, the relative position, the shape and the number of clusters can make the problem much more difficult. An increasing number of variables of course also complicates the process.

4 Conclusion

In this paper we present M-Xplore which is a new module of the Radioss software for crash simulations. It provides facilities to perform statistical investigations with the principal aim of design for crashworthiness.

In the module, variables and responses can be defined in a user-friendly way using new features of the FE preprocessor. Then a (statistical) task can be defined, (distribution law of each variables, sample size, sampling method), and the computations can be automatically launched either locally or remotely on a supercomputer. Advanced post processing facilities are also available for the user to explore the behavior of the design.

In this paper we also analyze two example problems. By this we illustrate typical problems in the domain, the techniques used and also how to work with the software. We emphasize the problem of finding failure modes and design criteria to avoid them.

In this challenging area there is no universal method which will give all information of all the models. The diversity of the problems require the use of many different approaches. The goal of M-Xplore is to incorporate many efficient methods for exploration and fit them into a convenient tool to facilitate the investigation of crashworthiness problems.

Acknowledgments

The work has been supported by Marie Curie Fellowship of the European Community programme GROW under contract number G3TR-CT-2000-00038. This support is gratefully acknowledged. The author R. Stocki would also like to gratefully acknowledge the partial support from The Foundation of Polish Science (FNP nr 4/2001).

References

[1] Mecalog SARL, 2 Rue de la Renaissance 92160 Antony, France, RADIOSS Input Manual, Version 4.2 (2000).

[2] W. Li, Optimal design using CP algorithms, Proceedings for 2nd world conference of the international association for statistical computing (1997) 130–139.

[3] M. Liefvendahl, An implementation and evaluation of the CP-algorithm for optimization of latin hypercubes, Tech. rep., Mecalog (2002).

[4] P. Audze, V. Eglais, New approach to planning out of experiments, in: Problems of dynamics and strength, Vol. 35, 1977, pp. 104–107, (in Russian).

[5] J.S.H.M. Wismans, Injury Biomechanics, Eindhoven University of Technology, 1994.

[6] K. Schittkowski, A Fortran subroutine for solving constrained nonlinear programming problems, Annals of Operations Research 5 (1985) 485–500.

[7] M. Crisfield, Non-linear finite element analysis of solids and structures. vol. 1: Essentials, John Wiley & sons, 1991.