

# Analysis of Bulky Crash Simulation Results: Deterministic and Stochastic Aspects

Tanja Clees, Igor Nikitin, Lialia Nikitina, and Clemens-August Thole

Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven  
53754 Sankt Augustin, Germany  
{Tanja.Clees, Igor.Nikitin, Lialia.Nikitina,  
Clemens-August.Thole}@scai.fraunhofer.de

**Abstract.** Crash simulation results show both deterministic and stochastic behavior. For optimization in automotive design it is very important to distinguish between effects caused by variation of simulation parameters and effects triggered, for example, by buckling phenomena. We propose novel methods for the exploration of a simulation database featuring non-linear multidimensional interpolation, tolerance prediction, sensitivity analysis, robust multiobjective optimization as well as reliability and causal analysis. The methods are highly optimized for handling bulky data produced by modern crash simulators. The efficiency of these methods is demonstrated for industrially relevant benchmark cases.

**Keywords:** Surrogate Models, Bulky Data, Multiobjective Optimization, Stochastic Analysis.

## 1 Introduction

Simulation is an integral component of virtual product development today. The task of simulation consists mainly in solution of physical models in the form of ordinary or partial differential equations. From the viewpoint of product development the real purpose is product optimization, and the simulation is "only" means for the purpose. Optimization means searching for the best possible product with respect to multiple objectives (multiobjective optimization), e.g. total weight, fuel consumption and production costs, while the simulation provides an evaluation of objectives for a particular sample of a virtual product.

The optimization process usually requires a number of simulation runs, the results form a simulation dataset. To keep simulation time as short as possible, "Design of Experiments" (DoE, [1]) is applied, where a space of design variables is sampled by a limited number of simulations. On the basis of these samples, a surrogate model is constructed, e.g. a response surface [2], which describes the dependence between design variables and design objectives. Modern surrogate models [3, 4, 12-15] describe not only the value of a design objective but also its tolerance limits, which allow to control precision of the result. Moreover, not only scalar design objectives but whole simulation results, even highly resolved in space/time, ("bulky" dataset) can be modeled [12-15].

In this paper we will concentrate on the stochastic aspects of simulation processes. Industrial simulations, e.g. virtual crash tests, often possess a random component, related to physical and numerical instabilities of the underlying simulation model and uncertainties of its control parameters. Under these conditions the user is interested not only in the mean value of an optimization criterion, e.g. crash intrusion, but also in its scatter over simulations. In practice, it is required to fulfil optimization objectives with a certain confidence, e.g. 6-sigma. This task belongs to the scope of robustness or reliability analysis.

Often, the confidence intervals are so large that one has to reduce scatter before optimization. There is a part of scatter deterministically related to the variation of design variables, which can be found by means of sensitivity analysis. The other part is purely stochastic. It can be triggered by microscopic variations of design variables and - even if they are fixed - by the numerical process itself, e.g. by random scheduling in multiprocessing simulation. These microscopic sources are then amplified by inherent physical instabilities of the model related e.g. to buckling, contact phenomena or material failure. Stochastic analysis allows to track the sources of scatter, to reconstruct causal chains and to identify hidden parameters describing chaotic behavior of the model. If uncontrolled, these parameters propagate scatter to the optimization objectives. The challenge is to put them under control, at least partially, e.g. by predeformation of buckling parts, adjustment of contact conditions, placement of additional welding points etc. In this way the scatter of simulation can be suppressed and optimization becomes more robust.

In Sec.2 we will overview the methods for metamodeling of bulky simulation results; in Sec.3 we describe stochastic methods for reliability and causal analysis; Sec.4 presents applications of these methods to real-life examples in automotive design. The approaches presented in this paper have been implemented in software tools DiffCrash [9-11] and DesParO [12-15] and have been subjects of international patent applications (DPMA 10 2009 057295.3 and PCT/ EP2010/061439).

## 2 RBF Metamodel

Numerical simulations define a mapping  $y=f(x): \mathbb{R}^n \rightarrow \mathbb{R}^m$  from n-dimensional space of simulation parameters to m-dimensional space of simulation results. In crash test simulation the dimensionality of simulation parameters  $x$  is moderate ( $n \sim 10-30$ ), while simulation results  $y$  are dynamical fields sampled on a large grid, typically containing millions of nodes and hundreds of time steps, resulting in values of  $m \sim 10^8$ . High computational complexity of crash test models restricts the number of simulations available for analysis (typically  $N_{exp} < 10^3$ ) which is preferred to be as small as possible.

Metamodeling with radial basis functions (RBF) is a representation of the form

$$f(x) = \sum_{i=1..N_{exp}} c_i \Phi(|x-x_i|), \quad (1)$$

where  $\Phi()$  are special functions, depending only on the Euclidean distance between the points  $x$  and  $x_i$ . The coefficients  $c_i$  can be obtained by solving a linear system

$$y_i = \sum_j c_j \Phi(|x_i - x_j|), \tag{2}$$

where  $y_i = f(x_i)$ . The solution can be found by direct inversion of the moderately sized  $N_{exp} \times N_{exp}$  system matrix  $\Phi_{ij} = \Phi(|x_i - x_j|)$ . The result can be written in a form of weighted sum  $f(x) = \sum_i w_i(x) y_i$ , with the weights

$$w_i(x) = \sum_j \Phi^{-1}_{ij} \Phi(|x - x_j|). \tag{3}$$

A suitable choice for the RBF, providing non-degeneracy of  $\Phi$ -matrix for all finite datasets of distinct points and all dimensions  $n$ , is the multi-quadric function [5]  $\Phi(r) = (b^2 + r^2)^{1/2}$ , where  $b$  is a constant defining smoothness of the function near data point  $x = x_i$ . RBF interpolation can also be combined with polynomial detrending, adding a polynomial part  $p(x)$ :

$$f(x) = \sum_{i=1..N_{exp}} c_i \Phi(|x - x_i|) + p(x). \tag{4}$$

This allows reconstructing exactly polynomial (including linear) dependencies and generally improving precision of interpolation. The precision can be controlled via the following cross-validation procedure: the data point is removed, data are interpolated to this point and compared with the actual value at this point. For an RBF metamodel this procedure leads to a direct formula [13-15]

$$err_i = f_{interpol}(x_i) - f_{actual}(x_i) = -c_i / (\Phi^{-1})_{ii}. \tag{5}$$

**Specifics of Bulky Data:** although RBF metamodel is directly applicable for interpolation of multidimensional data, it contains one matrix-vector multiplication  $f(x) = yw(x)$ , comprising  $O(mN_{exp})$  floating point operations per every interpolation. Here  $y_{ij}$ ,  $i=1..m$ ,  $j=1..N_{exp}$  is the data matrix, where every column forms one experiment, every row forms a data item varied in experiments.

Dimensional reduction technique applicable for acceleration of this computation is provided by principal component analysis (PCA). At first, an average value is row-wise subtracted, forming centered data matrix  $dy_{ij} = y_{ij} - \langle y_i \rangle$ . For this matrix a singular value decomposition (SVD) is written:  $dy = U\Lambda V^T$ , where  $\Lambda$  is a diagonal matrix of size  $N_{exp} \times N_{exp}$ ,  $U$  is a column-orthogonal matrix of size  $m \times N_{exp}$ ,  $V$  an orthogonal square matrix of size  $N_{exp} \times N_{exp}$ :

$$U^T U = 1, V^T V = V V^T = 1. \tag{6}$$

A computationally efficient method [14] for this decomposition in the case  $m \gg N_{exp}$  is to find Gram matrix  $G = dy^T dy$ , to perform its spectral decomposition  $G = V\Lambda^2 V^T$ , and to compute the remaining  $U$ -matrix with post-multiplication  $U = dy V \Lambda^{-1}$ . The  $\Lambda$  values are non-negative and sorted in non-ascending order. If all these values in the range  $k > N_{mod}$  are omitted (set to zero), the resulting reconstruction of  $y$ -matrix will have a deviation  $\delta y$ .  $L_2$ -norm of this deviation gives

$$err^2 = \sum_{ij} \delta y_{ij}^2 = \sum_{k > N_{mod}} \Lambda_k^2 \tag{7}$$

(Parseval's criterion). This formula allows controlling precision of reconstructed  $y$ -matrix. Usually  $\Lambda_k$  rapidly decreases with  $k$ , and a few first  $\Lambda$  values give sufficient

precision. The result of interpolation is represented as a product  $df = \Psi g$  of SVD modes  $\Psi = U\Lambda$  (principal components) to SVD-transformed RBF weights  $g = V^T w$ . Finally one has  $f(x) = \langle y \rangle + df(x)$ , computational cost of interpolation is reduced to  $O(m N_{\text{mod}})$ , plus once-charged  $O(m N_{\text{exp}}^2)$  cost of SVD. This method is convenient when interpolation should be performed many times ( $\gg N_{\text{exp}}$ ), e.g. for interactive exploration of database.

More generally, for representation of bulky data one can use clustering techniques [14]. They also decompose bulky data over a few basis vectors (modes) and accelerate linear algebra operations with them.

### 3 Reliability Analysis

The purpose of reliability analysis is an estimation of confidence limits (CL) for simulation results:  $P(y < CL) = C$ , where  $P$  is probability measure and  $C$  is a user specified confidence level. For example, median corresponds to 50% CL, i.e.  $P(y < \text{med}) = 0.5$ ; while 68% CL corresponds to confidence interval  $[CL_{\text{min}}, CL_{\text{max}})$ , where  $P(y < CL_{\text{min}}) = 0.16$ ,  $P(y \geq CL_{\text{max}}) = 1 - P(y < CL_{\text{max}}) = 0.16$ ; etc. Several methods for solution of this task are available.

#### 3.1 First Order Reliability Method (FORM)

FORM is applicable for linear mapping  $f(x)$  and normal distribution of simulation parameters:

$$\rho(x) \sim \exp(-(x-x_0)^T \text{cov}_x^{-1} (x-x_0)/2). \tag{8}$$

Here  $x_0 = \langle x \rangle$  is mean value of  $x$  and

$$(\text{cov}_x)_{ij} = \langle (x-x_0)_i (x-x_0)_j \rangle \tag{9}$$

is covariance matrix of  $x$ . In this case  $y$  is also normally distributed, with mean value

$$y_0 = \langle y \rangle = \text{med}(y) = f(x_0) \tag{10}$$

and covariance matrix

$$\text{cov}_y = J \text{cov}_x J^T, \tag{11}$$

where  $J_{ij} = \partial f_i / \partial x_j$  is Jacoby matrix of  $f(x)$ , called also sensitivity matrix. The diagonal part of  $\text{cov}_y$  gives standard deviations  $\sigma_y^2$  directly defining  $CL(y)$ , e.g.

$$CL_{\text{min/max}}(68\%) = \langle y \rangle \pm \sigma_y, \tag{12}$$

$$CL_{\text{min/max}}(99.7\%) = \langle y \rangle \pm 3\sigma_y. \tag{13}$$

In particular, when simulation parameters are independent random values,  $\text{cov}_x$  becomes diagonal:  $\text{cov}_x = \text{diag}(\sigma_x^2)$ , and

$$\sigma_{y_i}^2 = \sum_{j=1..n} (\partial f_i / \partial x_j)^2 \sigma_{x_j}^2. \tag{14}$$

A finite difference scheme used to compute Jacoby matrix of  $f(x)$  requires  $N_{exp}=O(n)$  simulations, e.g.  $2n$  for central difference scheme plus one experiment at  $x_0$ ,  $N_{exp}=2n+1$ . The algorithm possesses computational complexity  $O(nm)$  and can be implemented efficiently as reading data from  $N_{exp}$  simultaneously open data streams and writing CL to a single output data stream. In this way the memory requirements can be minimized and parallelization can be done straightforwardly.

**3.2 Second Order Reliability Method (SORM)**

SORM is applicable for slightly non-linear mapping  $f(x)$ , which can be approximated by quadratic functions. The distributions  $\rho(x)$  are normal or can be cast to normal ones by a suitable transformation of parameter. For quadratic approximations CL can be explicitly computed [6] using main curvatures in the space of normalized variables  $z_i=(x-x_0)_i/\sigma_{x_i}$ , i.e. eigenvalues of Hesse matrix  $H^1_{jk}=\partial^2 f_i / \partial z_j \partial z_k$ . These eigenvalues can be also used to estimate non-linearity of the mapping  $f(z)$ , by maximizing the 1st and 2nd Taylor's terms over a ball of radius  $R$ :

$$\max_{|z| \leq R} Jz = |J|R, \quad \max_{|z| \leq R} |z^T H z / 2| = H_{max} R^2 / 2, \tag{15}$$

so that the linear term prevails over quadratic one, in this ball, iff  $|J| \gg H_{max} R/2$ . Here

$$J_{ij} = \partial f_i / \partial z_j, \quad |J| = (\sum_j J_{ij}^2)^{1/2} \tag{16}$$

and  $H_{max}$  is maximal absolute eigenvalue of  $H$ . Both this criterion and estimation of main curvatures require full Hesse matrix, i.e.  $N_{exp}=O(n^2)$  simulations. Practically, the usability of SORM is limited, because strongly non-linear functions would involve higher order terms and because distributions of simulation parameters can strongly deviate from normal ones.

**3.3 Confidence Limits Determination with Monte Carlo Method (CL-MC)**

In the case of non-linear mapping  $f(x)$  and arbitrary distribution  $\rho(x)$  general Monte Carlo method is applicable. The method is based on estimation of probability

$$P_N(y < CL) = \text{num.of } (y_n < CL) / N \tag{17}$$

for a finite sample  $\{y_1, \dots, y_N\}$ . By the law of large numbers [7],  $F_N = P_N(y < CL)$  is consistent unbiased estimator for  $F = P(y < CL)$ , i.e.  $F_N \rightarrow F$  with probability 1, when  $N \rightarrow \infty$  and  $\langle F_N \rangle = F$  for all finite  $N$ . By the central limit theorem [7], the error of such estimation  $\text{err}_N = F_N - F$  at large  $N$  is distributed normally with zero mean and standard deviation  $\sigma \sim (F(1-F)/N)^{1/2}$ . Algorithmically the method consists of three phases:

- (CL1) generation of  $N$  random points in parameter space according to user specified distribution  $\rho(x)$ ,
- (CL2) numerical simulations for given parameter values,
- (CL3) determination of confidence limits by one-pass reading of simulation results, sorting  $m$  samples  $\{y_1, \dots, y_N\}$  and selection of  $k$ -th item in every sample with  $k = [(N-1)F + 1]$  as a representative for CL.

The analysis phase of the algorithm possesses computational complexity  $O(mN \log N)$  and can be efficiently implemented using data stream operations similar to FORM. Precision of the method is estimated using standard deviation formula above. Remarkably, the precision depends neither on dimension of parameter space  $n$ , nor on the length of simulation result  $m$ , but only on sample size  $N=N_{exp}$  and user-specified confidence level  $F=C$ . For instance, CL determination at the level 68% ( $F=0.16$ ) with 4% precision requires  $N_{exp}=84$ , while for  $68\% \pm 1\%$  one needs  $N_{exp}=1344$ .

### 3.4 Monte Carlo Combined with RBF Metamodel (MC-RBF)

Large sample size is required for precise determination of CL with Monte Carlo method. To reduce the number of required simulations, RBF metamodel can represent the mapping  $f(x)$  during analysis phase of CL-MC. While a metamodel can be constructed using a moderate number of simulations, e.g.  $N_{exp} \sim 100$ , determination of CL can be done with  $N \gg N_{exp}$ . Application of RBF metamodel for CL computation proceeds similarly to CL-MC. The only difference is that  $N_{exp}$  parameter points generated at phase (CL1) are used as input for the metamodel. They should not necessarily possess user specified distribution, but one providing better precision of metamodel, i.e. better covering "the corners" of parameter space. It is especially important for populating tails of distribution, corresponding to high confidence e.g. 99.7% CL. Uniform distribution is suitable for this purpose. Then, after numerical simulations at phase (CL2), and after filtering out failed experiments, the actual distribution  $\rho(x)$  is used to generate  $N$  parameter points, and construct RBF weight matrix  $w_{ij}=w_i(x_j)$ ,  $i=1..N_{exp}$ ,  $j=1..N$ . This matrix is used in phase (CL3) for multiplication with simulation results  $y_{ik}$ ,  $k=1..m$ , comprising  $O(m N N_{exp})$  operations, which usually prevails over  $O(m N \log N)$  operations needed for sorting of interpolated samples.

## 4 Causal Analysis

Causal analysis is determination of cause-effect relationships between events. In context of crash test analysis, this usually means identification of events or properties causing the scatter of the results. This allows to find sources of physical or numerical instabilities of the system and helps to reduce or completely eliminate them.

Causal analysis is generally performed by means of statistical methods, particularly, by estimation of correlation of events. It is commonly known that correlation does not imply causation (this logical error is often referred as "cum hoc ergo propter hoc": "with this, therefore because of this"). Instead, strong correlation of two events does mean that they belong to the same causal chain. Two strongly correlated events either have direct causal relation or they have a common cause, i.e. a third event in the past, triggering these two ones. This common cause will be revealed, if the whole causal chain i.e. a complete sequence of causally related events will be reconstructed. Practical application of causal analysis requires formal methods for reconstruction of causal chains.

A practical problem of causal analysis in crash-test simulations is often not a removal of a prime cause of scatter, which is the crash event itself. It is more an observation of propagation paths of the scatter, with a purpose to prevent this propagation, by finding regions where scatter is amplified (e.g. break of a welding point, pillar buckling, slipping of two contact surfaces etc). Since a small cause can have large effect, formally earliest events in the causal chain can have a microscopic amplitude ("butterfly effect"). Therefore it is reasonable to search for amplifying factors and try to eliminate them, not the microscopic sources.

As input for causal analysis the centered data matrix  $dy_{ij}$ ,  $i=1..m$ ,  $j=1..N_{exp}$  is used. Here every column forms one experiment, every row forms a data item varied in experiments, and the mean value  $\langle y \rangle$  is row-wise subtracted from the matrix. Then every data item is transformed to a z-score vector [8]:

$$z_{ij} = dy_{ij} / |dy_i|, |dy_i| = \sqrt{\sum_j dy_{ij}^2}, \tag{18}$$

or by means of the equivalent alternative formula

$$z_{ij} = dy_{ij} / (s(y_i)(N_{exp})^{1/2}), s(y_i) = (\sum_j dy_{ij}^2 / N_{exp})^{1/2}. \tag{19}$$

Here  $s(y_i)$  is the root mean square deviation of the  $i$ -th data item, which can serve as a measure of scatter. In this way the data items are transformed to  $m$  vectors in  $N_{exp}$ -dimensional space. All these  $z$ -vectors belong to an  $(N_{exp}-2)$ -dimensional unit-norm sphere, formed by intersection of a sphere  $|z|=1$  with a hyperplane  $\sum_j z_{ij}=0$ . The scalar product of two  $z$ -vectors is equal to Pearson's correlator of data items:

$$(z_1, z_2) = \sum_j z_{1j} z_{2j} = \text{corr}(y_1, y_2). \tag{20}$$

An important role of this representation is the following. Strongly correlated data items correspond either to coincident ( $z_1=z_2$ ) or opposite ( $z_1=-z_2$ )  $z$ -vectors. If not the sign but only the fact of dependence is of interest, one can glue opposite points together formally considering a sphere of  $z$ -vectors as projective space. Using this representation, one can apply [13,14] general purpose clustering methods such as  $k$ -means to group data items distributed on this sphere to a few strongly correlated components.

In spite of their numerical efficiency, these clustering methods neglect temporal ordering of events, while in causal analysis the task is to find an earliest physically significant event in the causal chain. In crash test simulation such events correspond to bifurcation points, where the scatter appears "ex nihilo". Such points are clearly visible as spikes in dynamical scatter plots  $s(y)$ , the problem is that there are too many of them. Although decision between potential candidates by a formal algorithm can be difficult, an engineering knowledge allows narrowing the search to significant parts where scatter propagation can be really initiated by physical effects, such as buckling of longitudinal, break of welding point etc. The other problem is that in bifurcation points new scatter is just appeared and it is generally hidden under the consequences of previous effects. At first one needs to separate scatter contributions.

Considering two data items  $dy(a)$  and  $dy(b)$ , one can define contribution relevant to the data item (a) in (b) as follows:

$$dyl_a(b) = \text{corr}(a,b)s(b)z(a). \tag{21}$$

After subtraction of this contribution a residual  $dy(b)-dyl_a(b)$  does not correlate with  $dy(a)$ , and the scatter

$$s^2_{l_a}(b) = \langle (dy(b)-dyl_a(b))^2 \rangle = s^2(b)(1-\text{corr}^2(a,b)) \tag{22}$$

does not increase.

Armed with this subtraction procedure, we propose the following algorithm for causal analysis.

*Temporal clustering:*

- (T1) visualize scatter state-by-state;
- (T2) isolate bifurcation point;
- (T3) subtract its contribution from the scatter in consequent states;
- (T4) if scatter is still remaining, goto (T1).

Here subtraction of scatter from previous bifurcations reveals new bifurcations hidden under the consequences of previous ones. The remaining scatter monotonously falls during the iterations, and the iterations can be stopped when the scatter becomes small everywhere or in the regions of interest.

The geometrical meaning of subtraction procedure:  $b-bl_a = b-(a,b)/(a,a)a$  is an orthogonal projection in the space of data items and the whole sequence is Gram-Schmidt (GS) orthonormalization procedure applied in the order of appearance of bifurcation points  $a_i$ . The obtained orthonormal basis  $g_i = GS(a_i)$  can be used for reconstruction of all data by the formula:

$$dy = \sum_i \Psi_i g_i + \text{res}, \Psi_i = (dy g_i). \tag{23}$$

The norm of residual is controlled by remaining scatter, which is small according to our stop criterion:

$$|\text{res}|^2 / N_{\text{exp}} = s_{\perp}^2(y) = s^2(y) - \sum_i \Psi_i^2 / N_{\text{exp}}. \tag{24}$$

Algorithmically every  $i$ -th iteration one computes a scalar field  $\Psi_i$  describing contribution of  $i$ -th bifurcation point to scatter of the model and a scalar field  $s_{i\perp}^2(y)$  used for determination of the next bifurcation point  $a_{i+1}$ , or for stop criterion  $s_{i\perp}^2(y) < \text{threshold}$ . This requires  $O(mN_{\text{exp}})$  floating point operations per iteration.

Matrix decomposition of the form  $dy = \Psi g$  is similar to PCA described above, with the other meaning of the modes  $\Psi$ . Like in PCA,  $\Psi$  are scalar fields distributed over dynamical model which are common for all experiments. They have the other temporal profile than PCA modes, reflecting causal structure of scatter: they start at corresponding bifurcation points and propagate forward in time. Differently from PCA modes, they are not orthogonal columnwise, i.e. with respect to scalar product over the model.  $g$ -coefficients form  $N_{\text{exp}} * N_{\text{mod}}$  columnwise orthonormal matrix. Like corresponding matrix in PCA, they define an orthonormal basis in the space of experiments, with respect to the scalar product coincident with Pearson's correlator.

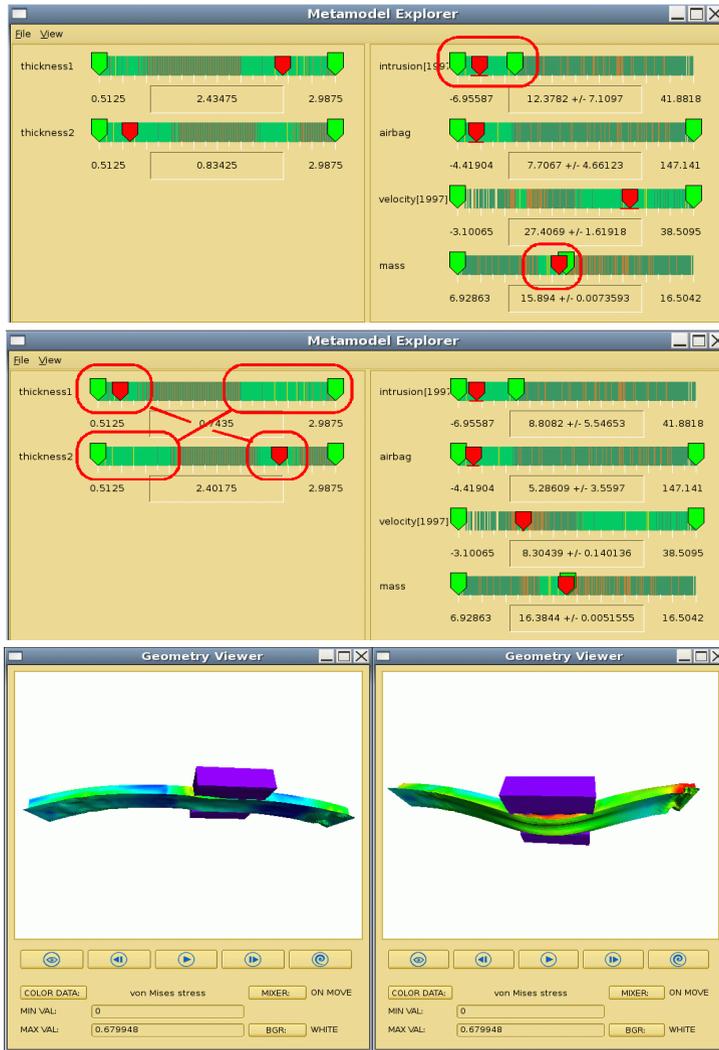
The scatter associated with design variables can be treated by the same method, if one puts data items containing variation of design variables as the first candidates for bifurcation points. The corresponding  $\Psi$ -modes will represent sensitivities of simulation results to variation of parameters. The remaining scatter represents indeterministic part of the dependence. The corresponding  $\Psi$ -modes are bifurcation profiles and their g-coefficients are those hidden variables which govern purely stochastic behavior of the model. One can either take hidden variables into account when performing reliability analysis, or try to put them under control for reducing scatter of the model.

## 5 Examples

### 5.1 Audi B-Pillar Crash Test

The model shown on Fig.1 contains 10 thousand nodes, 45 timesteps, 101 simulations. Two parameters are varied representing thicknesses of two layers composing a part of a B-pillar. The purpose is to find a Pareto-optimal combination of parameters simultaneously minimizing the total mass of the part and crash intrusion in the contact area. To solve this problem, we have applied the methods described in Sec.2, namely RBF metamodeling of target criteria for multiobjective optimization and PCA for compact representation of bulky data. Based on these methods, our interactive optimization tool DesParO supports real-time interpolation of bulky data, with response times in the range of milliseconds. As a result, the user can interactively change parameter values and immediately see variations of complete simulation result, even on an ordinary laptop computer.

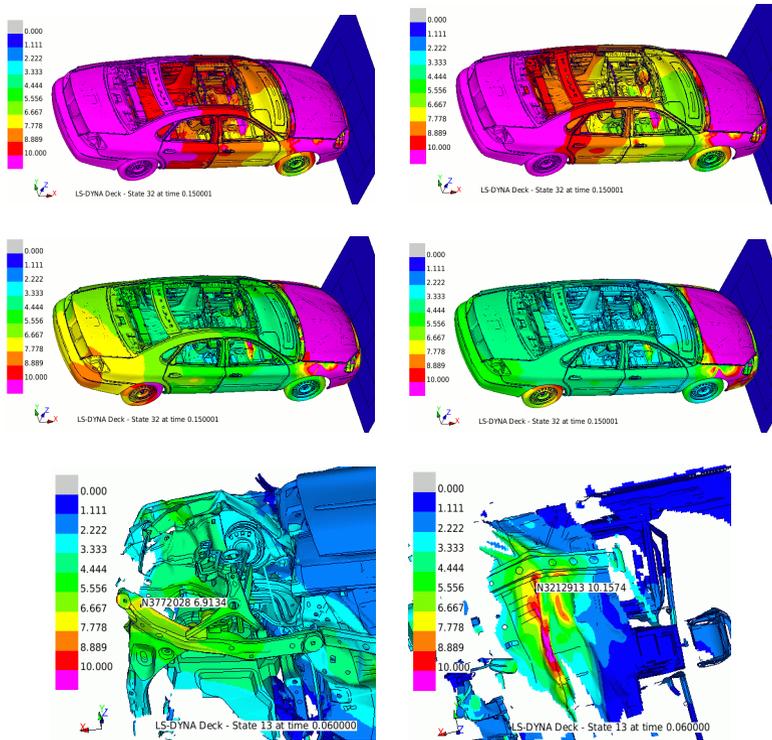
In more details, Fig.1 shows the optimization problem loaded in the Metamodel Explorer, where design variables (thicknesses1, 2) are presented at the left and design objectives (intrusion and mass) at the right. First, the user imposes constraints on design objectives, trying to minimize intrusion and mass simultaneously, as indicated by red ovals on Fig.1 (upper part). As a result, "islands" of available solutions become visible along the axes of design variables. Exploration of these islands by moving corresponding sliders shows that there are two optimal configurations, related cross-like, as indicated on Fig.1 (middle). For these configurations, both constraints on mass and intrusion are satisfied, while they correspond to physically different solutions, distinguished by an auxiliary velocity criterion. For every criterion also its tolerance is shown corresponding to 1-sigma confidence limits, as indicated by horizontal bars under the corresponding slider as well as +/- errors in the value box. This indication allows satisfying constraints with 3-sigma (99.7%) confidence, as shown on the images. The Geometry Viewer, shown at the bottom of Fig.1, allows to inspect the optimal design in full details.



**Fig. 1.** Audi B-Pillar crash test in DesParO Metamodel Explorer. (top): constraints on intrusion and mass are imposed. (center): two optimal designs are found. (bottom): inspection of optimal design in DesParO Geometry Viewer.

E.g. on the two images at the bottom, one can see the difference between small and large thickness values resulting in softer or stiffer crash behavior.

While performing constraint optimization, the user immediately sees how small mass solutions disappear when intrusion is minimized. This gives an intuitive feeling for the trade-off (Pareto behavior) between optimization objectives. With these capabilities and complementary information such as auxiliary criteria and interactive interpolation of bulky simulation results, “the” optimal solution, i.e. a single representative on the Pareto front, can be selected by a user decision.



**Fig. 2.** Temporal clustering of Ford Taurus crash test using DiffCrash. (*upper left*): original scatter in mm. (*upper right – center left – center right*): consequent iterations of scatter subtraction. (*bottom*): two major bifurcations found.

## 5.2 Ford Taurus Crash Test

The crash model shown on Fig.2 contains 1 million nodes, 32 timesteps, 25 simulations. Processing of this model with the temporal clustering algorithm described above has been performed on a 16-CPU Intel-Xeon 2.9GHz workstation with 24GB main memory. It required 3min per iteration and converged in 4 iterations.

Crash intrusions in the foot room of the driver and passenger are commonly considered as critical safety characteristics of car design. These characteristics possess numerical uncertainties, the analysis of which falls in the subject of Sec.3-4. The upper left part of Fig.2 shows the scatter measure  $s(y)$ , in mm, distributed on the model. The scatter in the foot room is so large ( $>10\text{mm}$ ) that direct minimization of intrusion is impossible. Temporal clustering allows to identify sources of this scatter and to subtract relevant contributions. Further images show how the scatter decreases in these subtractions. After the 4<sup>th</sup> iteration the scatter in the foot room reaches a safe level ( $<3\text{mm}$ ). Several bifurcation points have been identified and subtracted per iteration; in this way the performance of the algorithm has been optimized. The two major bifurcations found are shown on the bottom part of Fig.2. They represent

buckling phenomena on the left longitudinal rail and a fold on the floor of the vehicle, which appear in earlier time steps. In total, 15 bifurcation points have been identified, representing statistically independent sources of scatter. The whole scatter in the model can be decomposed over the corresponding basis functions  $\Psi(y)$ . In this way the dimensionality of the problem is reduced to 15 variables (g-coefficients) completely describing the stochastic behavior of the model.

## 6 Conclusions

We have presented and discussed methods for nonlinear metamodeling of a simulation database featuring continuous exploration of simulation results, tolerance prediction and rapid interpolation of bulky FEM data. For the purpose of robust optimization, the approach has been extended by the methods of reliability and causal analysis. The efficiency of the methods has been demonstrated for several application cases from automotive industry.

Further plans include to use the results of causal analysis as a basis for modifications of a simulation model for improving its stability. We also plan to consider non-linear relationships between stochastic variables. Linear methods such as PCA and GS determine only a linear span over principal components, while some stochastic variables can become non-linear functions of others. For determination of such dependencies the methods of curvilinear component analysis (CCA) can be applied.

**Acknowledgements.** Many thanks to the team of Prof. Steve Kan at the NCAC and the Ford company for providing the publically available LS-DYNA models and to Andreas Hoppe and Josef Reicheneder at AUDI for providing PamCrash B-pillar model. The availability of these models fosters method developments for crash simulation. We are also grateful to Michael Taeschner and Georg Dietrich Eichmueller (VW) and to Thomas Frank and Wolfgang Fassnacht (Daimler) for fruitful discussions.

## References

1. Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley, London (1997)
2. Donoho, D.L.: High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality, Lecture on August 8, 2000 to the American Mathematical Society "Math Challenges of the 21st Century" (2000), <http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/AMS2000.html>
3. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Opt.* 13, 455–492 (1998)
4. Keane, A.J., Leary, S.J., Sobester, A.: On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. *J. Glob. Opt.* 33, 31–59 (2005)
5. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations. Cambridge University Press, Cambridge (2003)

6. Cizelj, L., Mavko, B., Riesch-Oppermann, H.: Application of First and Second Order Reliability Methods in the Safety Assessment of Cracked Steam Generator Tubing. *Nucl. Eng. Des.* 147, 359–368 (1994)
7. van der Vaart, A.W.: *Asymptotic Statistics*. Cambridge University Press, Cambridge (1998)
8. Larsen, R.J., Marx, M.L.: *An Introduction to Mathematical Statistics and Its Applications*. Prentice Hall, Upper Saddle River (2001)
9. Thole, C.-A., Mei, L.: Reason for Scatter in Simulation Results. In: 4th European LS-DYNA User Conference, vol. B-III, pp. 11–20. Dynamore Press, Ulm (2003)
10. Thole, C.-A.: Compression of LS-DYNA Simulation Results. In: 5th European LS-DYNA User Conference, vol. 6b, pp. 86–91. Dynamore Press, Birmingham (2005)
11. Thole, C.-A., Mei, L.: Data Analysis for Parallel Car-Crash Simulation Results and Model Optimization. *Simulation Modelling in Practice and Theory* 16(3), 329–337 (2008)
12. Stork, A., Thole, C.-A., Klimenko, S., Nikitin, I., Nikitina, L., Astakhov, Y.: Towards Interactive Simulation in Automotive Design. *Vis. Comput. J.* 24, 947–953 (2008)
13. Nikitin, I., Nikitina, L., Clees, T., Thole, C.-A.: Advanced Mode Analysis for Crash Simulation Results. In: 9th LS-DYNA User Forum, vol. S11, pp. 11–20. Dynamore Press, Bamberg (2010)
14. Nikitin, I., Nikitina, L., Clees, T.: Stochastic analysis and nonlinear metamodeling of crash test simulations and their application in automotive design. In: Browning, J.E., McMan, A.K. (eds.) *Computational Engineering: Design, Development and Applications*. Nova Science, New York (2011) (to be publ.)
15. Nikitin, I., Nikitina, L., Clees, T.: Nonlinear metamodeling, multiobjective optimization and their application in automotive design. In: Günther, M., Bartel, A., Brunk, M., Schoeps, S., Striebel, M. (eds.) *Progress in Industrial Mathematics at ECMI 2010*, vol. 17. Springer (2010) (to be publ. 2012)