ENABLING THE ANALYSIS OF FINITE ELEMENT SIMULATION BUNDLES

Rodrigo Iza Teran

Fraunhofer Institute for Scientific Computing, 53757 Sankt Augustin, Germany

Original Manuscript Submitted: 05/20/2012; Final Draft Received: 11/29/2012

We propose a methodology capable of allowing a fast evaluation of thousands of finite element design variants simultaneously. This approach uses a high dimensional analysis concept, namely diffusion maps, that has been in use successfully for years in many areas of science. Using feature vectors from a bundle of finite element simulations containing information on the design variables on different mesh sizes and applying this analysis concept is the purpose of this paper. Applying this approach enables the identification of a set of parameters (reduction coordinates) along which (i) geometrical variants can be identified and (ii) for random time-dependent problems, slow variables can be identified that show the variables with the most significant impact on a design. We demonstrate the application of this approach in several industrial examples in the areas of metal forming and vibration analysis as well as vehicle crash simulation, which is a noisy stochastic process. Finally, we show per example that this approach can identify and expound on the occurrence of a bifurcation point, a very important issue in vehicle design.

KEY WORDS: diffusion maps, dimension reduction, finite element simulation

1. INTRODUCTION

The use of fast finite element solvers on high performance computers as well as the use of three-dimensional (3D) visualization software for engineering analysis of the solutions (simulations) has enabled the rapid modeling of many product variants resulting in gigantic repositories of simulation bundles.

Engineers generate several variants of a specific model that need to be simulated. Each simulation can take several hours even with hundreds of processing cores; this design workflow generates thousand of results that have to be saved in very large archives.

Finite element simulation of physical processes is an essential tool in the development and improvement of industrial products [1]. This is achieved via a lumped approximate representation (a mesh) whereby the mathematical model of the physical process is numerically solved. For an accurate description, such numerical lumped approximations are very fine (several million mesh points and elements) and the resulting data sizes are often huge.

Our paper concentrates on two aspects of the virtual product development process, both involving the analysis of simulation bundles. The first aspect is the realization of many variants of a model where small modifications are carried out, changing material parameters or conducting geometrical modifications. The objective is to find a design that minimizes some criteria such as costs while taking some constraints into account (such as safety regulations), which is very time consuming. Postprocessing software tools are able to display the geometrical information of the model and the results of the physical simulation. Using engineering intuition, the engineer can analyze and compare several simulations simultaneously; nevertheless, this is generally limited to the analysis of only a few simulations at a time. This is because analyzing simulations using postprocessing software is based on the 3D visualization of the geometry and associated design variables of each configuration and can only be achieved in a sequential manner, that is, taking groups of simulations and discarding poor designs from each group. This process involves displaying
several 3D representations at the same time on a monitor and trying to identify similarities or differences in the values defined on the finite element mesh.

The next aspect involves the reuse of previous design information. Simulation repositories have a great deal of valuable information about design changes from other projects, and engineers like to take advantage of them. However, in most cases this information is not easily accessible in database queries because of its size and/or because the information is not consistently saved.

The above-mentioned problems in the product development process can be dealt with by using dimensionality reduction. Simulation bundles consist of several thousand simulations corresponding to a specific product development phase that contain many correlations since they originate from similar models. Therefore, it is reasonable to assume that embedding the data in a lower dimensional space would allow the identification of the so-called intrinsic geometrical structure of the data. The engineer can describe a number of simulations simultaneously using these structures and in addition is able to search the reduced dimensional spaces far more efficiently than in the original representation space.

Dimension reduction methods assume that a high dimensional dataset has actually an intrinsic lower dimensionality, i.e., the unknown lower dimensional information is embedded in a higher dimensional space, and this is the one we can actually observe. The method of dimension reduction finds a low dimensional representation that minimizes information loss when embedding the high dimensional information in lower dimensions. According to the criteria used to estimate this loss, it is possible to define a classification of different methods. For details about this taxonomy of the methods, please refer to [2].

In this paper, we deal specifically with the method of dimension reduction diffusion maps [3] as a way of finding geometrical descriptions (geometry of the clusters) of datasets. This approach is a kernel method that uses eigenvectors of a Markov matrix defining a random walk (diffusion) on the data to obtain a new description of a data set using a mapping toward a low dimensional Euclidean space. The distance in this space describes the relationships between data points (the geometry or connectivity of the data).

Based on the assumption of the low dimensionality of simulation bundles, we demonstrate a novel methodology for its analysis. An assembly of many simulations is represented in the Euclidean space of low dimension. The analysis process consists of data extraction, preprocessing, dimension reduction, and exploration. The application of this process is shown in three application areas, namely metal forming, NVH (noise vibration and harshness), and crash simulation. We can identify relevant parameters of the simulation and can show the possible utility of the approach from an engineering point of view.

In Section 2 we will review relevant concepts of diffusion maps and describe the proposed analysis methodology for simulation bundles in Section 3. The application of the approach for simulation bundles starts in Section 4, the first application in metal forming is described in Section 4.1 followed by the analysis of a real simulation bundle provided by AUDI for noise vibration harshness in Section 4.2. This dataset contains geometrical as well as material parameter modifications. Section 4.3 takes into consideration the analysis of a frontal crash simulation bundle, random variations of a material parameter are applied and in this case we will demonstrate that it is possible to identify a slow variable that describes a bifurcation point in a stochastic crash simulation.

Diffusion maps are used in this application paper as a representative method of the more general class of kernel methods that use a solid mathematical background [4]. All of these methods construct a similarity matrix based on some distance measure, once specified, a different nonlinear method can be found. For example, the use of a geodesic distance identifies the method Isomap; the use of a Euclidean distance and other normalization of the similarity matrix allows the specification of Laplacian-Eigenmaps and so on, details about this correspondence can be found in [5].

Diffusion maps have also been shown to be useful in applications [6–11] in particular because of its organizational power allowing a parametrization of datasets and the fact that it allows the identification of so-called slow variables, a concept that is very important in the context of simulation data, as will be verified in the following sections. Last but not least we would like to mention that we have tested some of the examples in this paper with other kernel methods: Isomap, Laplacian-Eigenmaps, local linear embeddings, and linear tangent space alignment (see [2]) for details about each method; the obtained results were similar in the experiments to the one obtained by diffusion maps. A detailed systematic comparison will be the subject of a further publication.

International Journal for Uncertainty Quantification
2. DIFFUSION MAPS

Diffusion maps construct a weighted graph $G = (X, W)$, the nodes in the graph are the $M$ observed data points in $X \subset \mathbb{R}^n$. The weight $W$ which is assigned to the edges between two data points is given by $w(x, y) = e^{-\Delta(x,y)\epsilon}$ (the kernel) where $\Delta(x,y)$ is an application-specific, locally defined distance. The value of $\epsilon$ controls the neighborhood size. A Markov random walk is depicted on the graph and the transition probability for going from $x$ to $y$ is proportional to $p(x, y) = w(x, y) / \sum_{z \in \Omega} w(x, z)$. If the points are similar, then this probability will be high and vice versa. The chain can be iterated in $t$ time steps and a so-called “diffusion distance” can be defined as follows:

$$D^2_t(x,y) = \sum_{z \in \Omega} (p_t(x,z) - p_t(y,z))^2/\phi_0(z),$$

where $\phi_0(z)$ is the so-called stationary distribution, found by iterating the chain until no more changes in the transition probability are apparent, and $p_t(x,y)$ is proportional to the transition probability for going from $x$ to $y$ in $t$ time steps.

The actual dimension reduction is based on mapping the high dimensional data set, but doing this in a way that still approximates the diffusion distance. With this objective in mind, it is shown [12] that the data set can be embedded into a low dimensional space $d$ carrying out the transformation,

$$\Psi_t: x \rightarrow [\lambda^1_t \psi_1(x), \lambda^2_t \psi_2(x), ..., \lambda^n_t \psi_n(x)],$$

where $\lambda_j$ and $\psi_j$ are the eigenvalues and right eigenvectors of the similarity matrix $P = p_t(x,y)_{x,y}$. The first eigenvector (for $j = 0$) is per construction constant and is not taken into account. It has been shown that [3]

$$D^2(x, y) = \sum_{j=1}^{M-1} \lambda^2_j (\psi_j(x) - \psi_j(y))^2 \approx \sum_{j=1}^{d} \lambda^2_j (\psi_j(x) - \psi_j(y))^2 = ||\Psi_t(x) - \Psi_t(y)||^2.$$  

Comparing (1) with (3) shows that one can approximate the distance between transition probabilities using the Euclidean distance in the reduced space. The amount of terms that are used for the embedding, the value of $d$, depends on the kernel and the value of $\epsilon$. If only the first eigenvalues are significant and the rest are small (presence of an spectral gap) it is possible to achieve a good approximation of the diffusion distance with only a few terms. For details refer to [3].

There are several aspects that should be taken into consideration in order to apply this method for analysis of simulation data. Through simple examples, we describe the type of information that can be obtained from the diffusion maps coordinates and its contributions toward the identification of slow variables in a stochastic dynamical system, even when only a small number of simulations are available.

2.1 Diffusion Coordinates for a Simplified One-Dimensional Simulation Bundle

Suppose we are given a mathematical model (a differential equation) for a physical system defined on a discrete one-dimensional (1D) uniform mesh $\{r_1, r_2, ..., r_n\}$; we additionally assume that this model depends on only one, let us say material parameter, $\alpha$. We can solve this mathematical model for arbitrary realizations of $\alpha$, as the result $M$ simulations are to be generated, and each of them can then be described in the following way:

$$x_i = (S_{\alpha_i}(r_1), S_{\alpha_i}(r_2), ..., S_{\alpha_i}(r_n)), \ i = 1, ..., M.$$  

The value $S_{\alpha_i}(r_j)$ is the value of the simulation for the parameter $\alpha_i$ at point $r_j$ on the mesh. Although the ambient space is $\mathbb{R}^n$, the points are located on a 1D curve $\Lambda$ parametrized by $\alpha$ such that

$$\Lambda = \{y(\alpha) = (S_{\alpha}(r_1), ..., S_{\alpha}(r_n)) | \alpha \in [0, \alpha_{max}]\}.$$  

The points are brought to $\mathbb{R}^n$ by means of a mapping $i : \mathbb{R} \to \mathbb{R}^n$. Even though the data points have dimension $n$, in reality all simulation points are on the curve $\Lambda$ that is parametrized by $\alpha$. We know in this case the intrinsic dimension ($d = 1$) and we would like to know the type of information that is recovered by the diffusion coordinates.
The method constructs a similarity matrix of size $M \times M$, with kernel $w(x_i, x_j)$, and from this matrix we can calculate the first nontrivial eigenvectors. In the continuous scenario we can consider a corresponding random walk integral operator on the data set $X$

$$Pf(x) = \int_X p(x, y)f(y)d\mu(y). \quad (6)$$

If the probability density $\mu(y)$, from which the data are sampled, is uniform, and in the limit $\epsilon \to 0$, a differential operator (corresponding to the Laplace-Beltrami operator) is obtained. For our example it is the second derivative with respect to $\alpha$, and the eigenvalue problem is determined as

$$\frac{d^2 f}{d\alpha^2} = \lambda f. \quad (7)$$

For a nonclosed curve with two end points and length $\alpha_{\text{max}}$, the first nontrivial eigenfunction is of the form $\cos(\alpha/\alpha_{\text{max}})$ [12]. The eigenvectors of the similarity matrix approximate the eigenfunctions of the limit operator (6) and therefore, we can approximately recover the parameter $\alpha$ that generated the simulation (up to a transformation given by the cosine function).

### 2.2 Stochastic Dynamical Systems and Slow Variables for Simulation Bundles

Since real data are not uniformly sampled, there is a specific probability distribution $\mu$, mostly unknown, from which the data are sampled. Analyzing the effect of this distribution has been the subject of intensive research [13]. As described above in the case of uniform sampling the limit operator is the Laplace-Beltrami operator and for non-uniform sampling the limit operator is the backward Fokker-Planck operator [14].

Given a transition probability density $q(x, t|y, s)$ of finding the system at location $x$ at time $t$, the distribution of this density at a later time $s > t$ obey the backward Fokker-Planck equation [15]

$$\frac{\partial q}{\partial s} = \Delta q - \nabla q \cdot \nabla U, \quad (8)$$

where $U(x) = -\log(\mu(x))$ and $\mu$ is the probability density.

We mention this theoretical result because of its relevance when a weighted graph is constructed from simulation bundles generated from stochastic processes. The Fokker-Planck equation describes the time evolution of the probability density function of a stochastic mechanical, physical, chemical, or biological system. In the presence of a spectral gap the long-term behavior of the system is approximately described using only the first eigenfunctions of this operator. In other words for a stochastic system with many random variables, the first eigenfunctions will describe the behavior of the slow variables (or principal trends), corresponding to the conformational state changes. For example, a system can have many variables that oscillate very rapidly around a state as the system progresses in time, only after a long time there are changes to a new state (conformational change). For many applications, the slow variables are precisely the interesting ones. Having enough data sets, we can construct a discrete approximation of this operator, and its eigenvectors represent the slow variables of an stochastic process.

Crash simulation is a stochastic process where variability is induced naturally on input parameters such as material, geometry, or even numerical noise in the finite element calculation. In such processes it is important, at least from the point of view of the engineer, to study and identify so called conformational changes. In structural dynamics this could correspond to a change in the bending state of a beam. This effect produces a bifurcation in the deformations that propagate to the overall structure. We assume that for this stochastic process, there is a corresponding Fokker-Planck operator from which we can extract the first eigenfunctions. According to the theory, the slow variables of the system will describe the appearance of a bifurcation point in the deformation (buckling mode) and this behavior can be recovered by the eigenvectors of a discrete approximation of this operator. In the next subsection we would like to illustrate these ideas with a specific example.
2.2.1 Slow Variable for a Simple Simulation Bundle

In this subsection we deal with a specific example that, for illustration purposes, can be evaluated analytically. In this case we assume that the probability density $\mu$, normally unknown, is given. We keep for the purpose of simplicity a 1D formulation that we extend to two-dimensional (2D) thanks to the specially chosen density.

We evaluate the integral operator (6); from it we can explicitly find its eigenfunctions (denoted by $\phi$) that, according to the theory in the limit $\epsilon \to 0$ for a nonuniform probability, corresponds to the eigenfunctions (denoted by $\psi$) of a backward Fokker-Planck differential operator (8) (once multiplied by a density-dependent normalization factor). A separable 2D extension of the probability density allows us to write directly the corresponding eigenfunctions for the 2D case. We start with a discrete data set obtained by sampling from the given 2D probability density. Using this sampled data we construct the similarity matrix (see Algorithm 1) and calculate the first nontrivial eigenvector (ordered according to the value of the eigenvalues). According to the theory those eigenvectors approximate the first eigenfunctions of the backward Fokker-Planck operator. We show at the end of the section that we are able to approximate the first eigenfunction through the eigenvector of the matrix and finally analyze the effect of the number of data sets on the shape of this eigenvector. In general a simulation bundle can contain a hundred simulations and in very large repositories around thousand a simulations. A very important aspect for the intended application is to know whether we could still expect to obtain some dependable results for the case where only such few simulations are available.

In the context of Eq. (8), consider the potential $U(x) = x^2/2\tau$ with the probability density $\mu = e^{-U}/\sqrt{2\pi\tau}$. We assume that a simulation bundle is obtained by sampling from this density (Brownian movement of particles in a potential field [14]). Using this density it is also possible to define the continuous integral random walk operator

$$\tilde{P}_c \phi(x) = \int_{-\infty}^{\infty} \tilde{p}_c(x,y)\phi(y)\,\mu(y)\,dy,$$

where the following kernel normalization has been used:

$$\tilde{p}_c(x,y) = \frac{k_c(x,y)}{\sqrt{\mu_c(x)\sqrt{\mu_c(y)}}}$$

and $k_c(x,y) = \exp(-||x-y||^2/\epsilon)$, $\mu_c = \int k_c(x,y)\,\mu\,dy$. The eigenvalues of the integral operator (9) are $\lambda_k = (\tau/(\tau + \epsilon)) < 1$, with corresponding eigenfunctions $\phi_k(x) = \tau_k(x)\exp(-x^2/4(\tau + \epsilon))$, $\tau_k$ is a polynomial of degree $k$.

In the limit $\epsilon \to 0$ this integral operator converges to the backward Fokker-Planck operator (8), and the corresponding eigenfunctions $\psi(x)$ of such an operator can be obtained by normalization of the ones of (9) with $\sqrt{\mu(y)} = C_\epsilon \exp(-y^2/(4\tau))$ (see [14] for details).

Just for building this example we consider an extension of this type of potential to two dimensions in the form $U(x_1,x_2) = x_1^2/(2\tau_1) + x_2^2/(2\tau_2)$ and choose $\tau_1 \gg \tau_2$ so that $x_1$ is the slow variable; this specially chosen probability density has a separable structure $\mu(x_1,x_2) = \mu_1(x_1)\mu_2(x_2)$. The eigenfunctions and eigenvalues are given by

$$\phi_{i,j}(x_1,x_2) = \phi_{i,1}(x_1)\phi_{2,j}(x_2), \quad \lambda_{i,j} = (\tau_1/(\tau_1 + \epsilon))^{i}(\tau_2/(\tau_2 + \epsilon))^{j}. \quad (11)$$

Since we set $\tau_1 \gg \tau_2$, ordering the eigenfunctions by decreasing eigenvalue, the first nontrivial eigenfunctions are $\phi_{1,0}, \phi_{2,0}$. Since the order of the polynomial is $i$ and $j$, it can be seen that

$$\phi_{1,0}(x_1,x_2) = \phi_{1,1}(x_1)\phi_{2,0}(x_2) = Cx_1 \exp\left(-\frac{x_1^2}{4\tau_1}\right) \exp\left(-\frac{x_2^2}{4\tau_2}\right). \quad (12)$$

After conjugation (dividing by $\sqrt{\mu}$ for each variable) it can be seen that this first eigenfunction depends linearly only on $x_1$.

$$\psi_1(x_1,x_2) = C x_1. \quad (13)$$
Our next task is to verify this analytical result. In order to do that, we will generate several data sets sampled from the given 2D probability distribution $\mu(x_1, x_2)$ (see Fig. 1 using $\tau_1 = 1$, $\tau_2 = 25$). Notice that the specific value of $\tau$ is not essential in this analysis, but the fact that $\tau_1 \gg \tau_2$. Using these data sets we build a similarity matrix, normalize it [according to (10)] and obtain the first diffusion coordinate (first eigenvector). Plotting the values of $x_1$ against the first non-constant eigenvector $\psi_1$ shows that, for $M = 1000$, a linear dependence is obtained except at the boundaries [Fig. 2(a)]. Next we test the approach after the data sets are reduced drastically for example to only 100 data points; the dependence is still clearly seen even for this very small amount of data sets [Fig. 2(b)]. Aiming to theoretically support the assumption that even with a very small number of points (number of datasets) the diffusion coordinates can still be useful is in itself difficult. In order to approximate the eigenfunctions of the continuous Laplace-Beltrami operator or the Fokker-Planck operator, we need several thousand datasets. In the case of the Gauss kernel with a fixed value of $\epsilon$, it is known (see [16]) that the rate of convergence of the eigenvectors is of the order $O(1/\sqrt{M})$. It is also known that eigenvectors obtained from a reduced similarity matrix obtained by sampling from a finer one can approximate the eigenvectors of the original matrix (using the Nyström approximation; see [17]). A formal analysis of the convergence properties for very few data sets is certainly needed; nevertheless, the actual application in Section 4 seems to confirm that even for a small number of simulation datasets, useful results can still be obtained.

Choosing the value of $\epsilon$ is still a very important issue in using eigenvectors for embedding a dataset and implementing a Gauss kernel as in our application. Since most of the time only a limited amount of datasets are available, all proposed methods for setting $\epsilon$ are not practical. Therefore, an empirical way of setting it consists of observing the obtained clusters or low dimensional structures for several values of $\epsilon$. For all cases we have dealt with in this paper, we have observed that for range of values of $\epsilon$ the same shapes of the clusters are obtained.

2.2.2 Diffusion Map Methodology for a Simulation Bundle

In this section we provide details of the application of diffusion maps to a simulation dataset in a algorithmic way (see Algorithm 1). The training data set is saved on files in binary proprietary format. So-called postprocessor software [18, 19], can be used to read these data and extract all or part of them into for example ASCII format. We use the software Animator® [18] to extract specific components of a car or structure. The components we choose are the ones that are critical for the engineer in the sense of structural behavior under crash or vibration response under excitation.

FIG. 1: Data sampled from a bivariate distribution.
Notice that in our analysis we use simulation data on a finite element mesh directly or extract some response curves for vibration analysis again using Animator. A finite element mesh contains nodes and elements. We assume that the mesh connectivity is the same and use only the values defined at the nodes on the mesh as data set. If the mesh connectivity is different, a reference mesh can be used to map the simulation values to it.
To find the reduced representation for a new finite element data set that is not available in the training set we use a Nyström method (Algorithm 2).

**Algorithm 2: Extension using Nyström**

**Input:** $Q$ finite element test data sets containing a variable $\hat{x} \in \mathbb{R}^n$, where $n$ is the number of nodes in a finite element mesh or the number of points on a curve for vibration response at a point

**Output:** Reduced representation $[\hat{\lambda}_1^t \hat{\psi}_1(\hat{x}), \hat{\lambda}_2^t \hat{\psi}_2(\hat{x}), ..., \hat{\lambda}_d^t \hat{\psi}_d(\hat{x})] \in \mathbb{R}^d$, where $d \ll n$

1: foreach $j$ in $M$ do 
2:     foreach $j$ in $Q$ do 
3:         Calculate $D_{ext}(i,j) = ||\hat{x}_i - \hat{x}_j||$; 
4:     end 
5: end 
6: $D_{sum} = D_{sum} + \min(\text{nonzeros}(D(i,:)), i \in M)$; 
7: $e = D_{sum}/M$; 
8: Form the matrix $K_j = \exp(-D/e)$; 
9: $p = K_1 \ast 1$ where $1 = (11...1)'$; 
10: $K_2 = K_1 ./ (p \ast p')$; 
11: $v = \sqrt{\text{diag}(K_2 \ast 1)}$; 
12: $K = K_2 ./ v \ast v'$; 
13: Diagonalize $K$ by $[U, S, V] = \text{svd}(K)$; 
14: The eigenvalues are $\lambda = \text{diag}(S)$; 
15: The eigenvectors are $\psi = U$; 
16: The first eigenvalue $\lambda_0$ is 1 the following eigenvalues are $\lambda_i < 1$: 
17: Take $\lambda_1^t \ast U_1, \lambda_2^t \ast U_2, ..., \lambda_d^t \ast U_d$ as the first $d$ spectral coordinates; 
18: The value of $d$ is taken so that $d = \max\{j: \lambda_j > 0.1\}$

International Journal for Uncertainty Quantification
3. METHODOLOGY

We describe a general methodology that allows analysis of finite element simulations bundles. Four steps are identified in it:

- **Extraction**
  The variables for the analysis are obtained from the simulation. These variables can be of a different type such as scalars, vectors, or tensors defined on nodes or elements of a finite element mesh.

- **Preprocessing**
  This step is necessary to cope with the problem of the data size (a million nodes and elements). The preprocessing contains simple subsampling, that is to just take consistently only subsets of the data. A further substep that is important to mention is the generation of compact signatures. The data can be transformed in different ways in order to obtain a compact representation; examples of such a transformation include principal component analysis (PCA) or even using diffusion maps. Another effective way of dealing with the high dimensionality problem is to measure the response of the simulation at specific points in the finite element structure instead of the overall simulation data (see NVH application in Section 4.2).

- **Dimension reduction**
  In this step a low dimensional representation is obtained from the dataset that parametrizes the information, allowing the identification of intrinsic parameters. We use the concept of diffusion maps as a dimensionality reductive framework. Notice that other dimension reduction methods can be used at this point, without giving a complete justification; we do mention that to our knowledge the connection to stochastic dynamical systems and the identification of slow variables provided by this method is not yet available using other methods. This is a key issue when dealing with simulations, as will be demonstrated later.

- **Exploration**
  This step can be done as long as the simulation variables are organized along the low dimensional embedding space obtained in the dimension reduction step. All simulations can be represented in this low dimensional setting and not only that, since diffusion maps are able to identify intrinsic parameters from the data, we can explore all the dataset in a simplified setting along such parameters. The engineer can obtain an overview of all his design variants in this way and the effect of the design variations can be explored along the intrinsic parameters.

4. ANALYSIS OF SIMULATION BUNDLES

In the next subsections the application of the proposed methodology as highlighted in Section 3 is presented for three engineering application areas. For all cases in addition to the trivial eigenvalue by 1, only the first few eigenvalues are significant; correspondingly, we use the second and third eigenvectors ($\Psi_1, \Psi_2$) as diffusion coordinates. We note that in the practice, we do not expect that a high dimensional simulation data set can always be reduced to two dimensions. Nevertheless, as mentioned before, simulation data can be considered as a high dimensional dynamical system, and using diffusion maps we can extract the slow variables of the system in such a way that the first nontrivial eigenvector corresponds to the first slow variable, the second to the next slow variable, and so on. We could have more slow variables but we can be sure that by projecting to 2D, we will at least be able to see the first one. Another aspect is the point of view of clustering and classification; the second eigenvector is very informative (spectral clustering). Organizing the simulations according to the order given by the second eigenvector shows a way to parametrize or cluster all the data. The clustering obtained in this way is already very useful in practice, since the engineer can automatically see which of the variables have a larger impact on his design. This fact has been shown in all practical examples presented in this paper.
4.1 Analysis Case I

We will consider material parameter changes that produce variations in a plastic strain on a plate (a quarter disc). In the metal forming simulation, this is an interesting output variable that evaluates the permanent (inelastic) distortion of metals under applied stresses that strain the material beyond its elastic limit. The ability of metals to flow in a plastic manner without fracture is the fundamental basis for all metal forming processes.

The material law considers the effect of elastic-plastic hardening with a Swift Law \( \sigma = K(\varepsilon_0 + \varepsilon)^n \) as well as anisotropies, defined by several parameters \( r_{00}, r_{45}, r_{90}, r_m \) and \( \Delta \). For this specific application we have varied the parameters \( K \) and \( r_{00} \). Each parameter varies randomly between 0 and 3 times the nominal value. A total of 80 simulations were performed using LSDYNA® [20].

After each simulation, we extracted the plastic strain (extraction step) to obtain a set of vectors \( M = 80 \) of dimension \( n = 1780 \). No preprocessing is necessary in this case because of the reduced size of \( n \). The general procedure for nonlinear dimension reduction with diffusion maps is used building a similarity matrix with weights \( w_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{\epsilon}) \). The value of \( \epsilon \) is chosen according to the criteria given in Section 2.2.1.

Since the first eigenvector is constant we use the second and third ones for embedding purposes. The result can be seen in Fig. 3, where in addition the value of the corresponding parameter \( K \) is displayed as a color for each point, corresponding to each simulation.

FIG. 3: Metal forming embedding in 2D; the color insert represents the value of the parameter \( K \).
In Fig. 3, by analyzing the information from this 2D embedding (exploration step), there are several observations that can be made:

- The simulations have been categorized according to the intensity of the plastic strain. This can be verified for example by observing the corresponding strain distributions for a number of simulations on the left compared with the ones on the right of the 2D embedding.
- The intensity of the plastic strain is almost constant on the uppermost left point. On the contrary it has its biggest variation on the uppermost right point.
- Observing the simulations toward the middle ($x = 0.05$), one can recognize points where the strain distributions show a more local distribution on the disc, whereas the ones on the right branch of points shows a plastic strain distributed around the border of the disc.
- The values of the stiffness factor $K$ have been used as color for the points in the 2D embedding. As can be seen from the figure, the method identifies this parameter as a function of the changes on the plastic strain for all simulations.

The 2D embedding clearly suggest a clustering where the strong strain distributions are separated from the corresponding weaker ones. The material parameter $K$ is also shown to be separated according to its value. A decision about a specific material parameter combination that produces a specific strain distribution can be reached based on these results.

### 4.2 Analysis Case II

Noise, vibration, and harshness (NVH) is a normal analysis task that should be performed for the development of complex structural components subjected to dynamic loads. A car is an example of such structure because it is a composition of several components (the doors, roof parts, console, lateral stiffeners, support structure, etc.). A real example from a car company, used in this analysis, contains around 1000 components and each of them is represented using a very fine mesh structure with different types of elements (shells, volumes, beams). For this discretized structure a dynamic simulation is performed that calculates the behavior under simulated road operating conditions. For the engineers it is important to simulate the response of the structure to specific excitations. This response is calculated with the dynamic simulation software NASTRAN®. The results are saved as simulation data in a specific data format for graphic postprocessing. Based on this, a comparison can be made between several design variations where parameters such as thickness, material, or geometrical form are changed. The decision about an ideal configuration is normally a conflicting objective, subject to several design restrictions. For example, a structural component should be, from the point of view of the dynamics, stiff but at the same time this implies an increase in weight.

The response of the structure is analyzed at several points of interest in the structure. Figure 4(a) shows a typical vibration mode (using a model developed by the National Crash Analysis Center of George Washington University, USA for visualization [21]), the response [see Fig. 4(b)] to a specific excitation at points of interest (for example at the console front) shows the engineer how well a design fulfills the dynamical requirements for comfort and vibration, and can be evaluated from these vibration modes.

In the EU project SIMDAT (IST-2004-511438), the AUDI-AG have completed a series of model variations, with changes in the thickness of some of the components, changes in the mass of the cooler, or introduction of small structural changes such as small stiffeners at the doors. A total of 39 models were simulated using NASTRAN® software, each one of them representing a specific change.

The variable of interest here is the response of the structure at specific points. We could directly use the high dimensional displacement vectors ($n = 500,000$) of the vibration mode to do our analysis but we think it is not necessary to deal with such big vectors. A preprocessing step can be applied here based on the observation that for analysis purposes we can evaluate the response of the structure at specific points (at the console front) and use such smaller vectors for our purposes. We can extract this scalar quantity (extraction step) and obtain a set of 39 vectors of length ($n = 500$). We use the general procedure for nonlinear dimension reduction using diffusion maps,
building a similarity matrix with $w_{ij} = \exp[-((x_i - x_j)/\epsilon)^2]$ as before. The second and third eigenvectors have been chosen for the embedding purposes. The result can be seen in Fig. (5), where in addition to the response curves, the change in some of the car models (changes schematically represented using a model developed by the National

FIG. 5: 2D embedding showing a parametrization of vibration response curves obtained from 39 NVH simulations. Some geometrical or material parameter changes that were made in the input model are shown together with the corresponding simulation response curves.

International Journal for Uncertainty Quantification
Crash Analysis Center of George Washington University [21]) is superposed graphically. Each response curve has been given a number and this number is displayed in Fig. (5). Analyzing the information from this 2D embedding (exploration step), we can make the following observations:

- The different model simulations are clustered (visually) in at least four different groups.
- There is a specific shape of the response curve that can be associated with a specific car model parameter change for the mass, thickness, stiffness, and for the introduction of ribs at the doors. An engineer can of course reach this conclusion, but this demands a time-consuming process of visually analyzing each of the 39 simulation results and deciding which curves can be allotted to a specific cluster. The method we propose identifies this classification automatically.
- The cluster of results for the response curve that are associated with changes in thickness is very dense. The response curve for this cluster is correspondingly very similar. It can be stated that modifications in the car design based on the thickness variations that have been done, do not have a strong effect on the shape of the response curve.
- The introduction of changes in the cooler mass or on the cooler stiffness have a strong influence on the shape of the response curve as well as changes in the mass distributions. The introduction of doors-stiffener changes also has a significant influence.

The 2D embedding provides an overview of all design changes according to its effect on the vibration response of the structure. A decision about a specific design can be taken based on this analysis.

4.3 Analysis Case III

In developing a new car model, many variants of it are constructed; those variants are obtained by making alterations of different types such as its geometry or material. Each variant is calculated using the state-of-the-art finite element software that solves a mathematical model for very large plastic deformations. The calculation time for each variant is very high (several hours on a many-core system). The engineer conducts several trials to reduce construction costs that at the same time adhere to strict safety regulations and constructive functional constraints.

A simplified car model with around 65,000 nodes and corresponding elements developed by the National Crash Analysis Center of George Washington University (see [21]) is considered. The simulation of the model impacting frontally a barrier using 17 time steps (physically representing milliseconds) has been performed, a total of 9 thickness changes ($T_1 \ldots T_9$) were changed (±30%) randomly, and a total of $M = 132$ simulations were computed using the LSDYNA® software [20].

The variable we chose for this analysis is the total deformation difference with respect to a reference model. It is a scalar field defined at each node of a 65000 unstructured grid. The procedure highlighted in Section 2 is used, with the deformation considered as a vector in $\mathbb{R}^n$ where $n$ is the number of nodes of each simulation. An $M \times M$ matrix is obtained and we focus on the second and third eigenvectors. We used as preprocessing step the uniform sub-sampling, reducing the information by a factor of 4.

Each of the 132 simulations is represented as a point in the plot. A 2D parametric representation of the 132 simulations is obtained in this way (see Fig. 6). For some points in the 2D plot we superpose the plot of the deformation differences. Based on this representation, it is possible to compare all the simulations in a very easy way (exploration step). In Fig. 6 it can be clearly seen that the simulations are categorized according to the intensity of the deformation difference. On the sides of the curve the deformations increase sequentially and achieve a maximum at the right uppermost point and a minimum at the left uppermost point. The distribution of the values for the deformation in each branch differs; a closer look at the frontal part of the car shows two types of bending (buckling mode) of a structural beam (see inserted closer view in the upper part of Fig. 6). The color coding in the 2D plot corresponds to the value of the thickness of the structural beam and can be used as an indicator of the buckling behavior. Such a result is provided by the method and can be very useful from the application point of view, because the engineer can actually identify critical structural parts and the effect of the changes of the input parameter on the crushing behavior.
FIG. 6: Second vs. third eigenvector plot showing displacements differences and thickness (color coding) of a structural beam showing a bifurcation (see upper inset).

We wish to underpin here the expected result in this case, from a probabilistic point of view, calculating the first eigenvectors of the matrix approximates the eigenfunctions of a Fokker-Planck operator in the limit of infinitely many simulations. The first eigenfunctions identify the slow variables in a stochastic dynamical system. In our example there are just 132 simulations, nevertheless we are still able to recover approximately a slow variable corresponding to the thickness of a structural beam. Such a variable is shown to describe a buckling mode of the crash simulation. A more detailed analysis describing the input variable dependence will be presented in a future publication.

5. CONCLUSIONS

In this paper, we introduced a methodology for analyzing finite element simulations by identifying low dimensional structures using diffusion maps. We demonstrated the application of this methodology for three engineering applications and organized the data along such structures in a step-by-step method where simultaneous analysis of many simulations is made possible.

In the section on metal forming, we analyzed a structure under deformation loads, where two material parameter changes were carried out. Using the second and third diffusion map coordinates, the intensity of the distribution of the plastic strain is parametrized accordingly along those coordinates.

We presented results of a real scenario in the context of noise-vibration-hardness analysis. We concentrated on frequency response curves of a car structure at a specific point and obtained diffusion coordinates from 39
simulations. Such simulations were generated by conducting several parameter changes around a nominal configuration. We were able to identify several groups of points suggesting clusters for the shape of the response curves, along those coordinates. Such clusters can be linked to a specific change in the car design such as thickness, mass, or stiffness.

Finally, we analyzed a dataset from a crash simulation whereby nine parameters were randomly varied and thus demonstrated that the diffusion coordinates enabled the identification of a buckling behavior in a structural beam of the car undergoing a frontal crash. A parametrization of all simulations according to the thickness of one of the beams shows this phenomenon, whereby the thickness is a slow variable in the stochastic framework of diffusion maps.

We have focused first on the parametrization step since this already provides interesting practical applications. Nevertheless, a very important aspect is also the reconstruction of a high dimensional simulation from a low dimensional representation, with certainly very useful applications. We notice that using only the reduced representation, one can find the high dimensional corresponding representation. For doing that, one can use an interpolation based approach (doing an interpolation which uses coefficients calculated with nearest neighbors in diffusion space) like the one used in [22]. The approach, its validation, and the comparison with other approaches is the focus of a further publication.

ACKNOWLEDGMENTS

The author would like to thank the Fraunhofer Institute SCAI, especially Mr. Clemens-August Thole, and the Mathematical Institute of the University of Cologne, especially Professor Dr. Caren Tischendorf for the support during the time of the realization of the internal project RIEMANNS where this research work was done.

REFERENCES