# Fast Insight into High-Dimensional Parametrized Simulation Data

Daniel Butnaru, Benjamin Peherstorfer, Hans-Joachim Bungartz
*Technische Universität München*
*85748 Garching, Germany*
*Email: {butnaru,pehersto,bungartz}@in.tum.de*

Dirk Pflüger
*IPVS, Universität Stuttgart*
*70569 Stuttgart, Germany*
*Email: dirk.pflueger@ipvs.uni-stuttgart.de*

***Abstract*—Numerical simulation has become an inevitable tool in most industrial product development processes with simulations being used to understand the influence of design decisions (parameter configurations) on the structure and properties of the product. However, in order to allow the engineer to thoroughly explore the design space and fine-tune parameters, many – usually very time-consuming – simulation runs are necessary. Additionally, this results in a huge amount of data that cannot be analyzed in an efficient way without the support of appropriate tools. In this paper, we address the two-fold problem: First, instantly provide simulation results if the parameter configuration is changed, and, second, identify specific areas of the design space with concentrated change and thus importance. We propose the use of a hierarchical approach based on sparse grid interpolation or regression which acts as an efficient and cheap substitute for the simulation. Furthermore, we develop new visual representations based on the derivative information contained inherently in the hierarchical basis. They intuitively let a user identify interesting parameter regions even in higher-dimensional settings. This workflow is combined in an interactive visualization and exploration framework. We discuss examples from different fields of computational science and engineering and show how our sparse-grid-based techniques make parameter dependences apparent and how they can be used to fine-tune parameter configurations.**

***Keywords*-feature identification; visual analytics; sparse grids; surrogate modeling;**

## I. Introduction

A central task in data mining is feature identification in large collections of multivariate data stemming from observations of discrete events. Another field in which a similar task is lately pursued is simulation technology. Here, each data "point" is a full simulation corresponding to a set of parameters. The data mining task translates into identifying parameters and parameter ranges where the simulation exhibits interesting features. Furthermore, this should be done with as few expensive simulation runs as possible. This requires an appropriate, interactive exploration and investigation environment which helps the engineer to identify important features in the data. We present such an environment based on so-called sparse grids, a numerical discretization technique for high-dimensional problems [1]. They allow us to efficiently substitute the original simulation and offer valuable indicators with respect to interesting parameter constellations and parameter relevance.

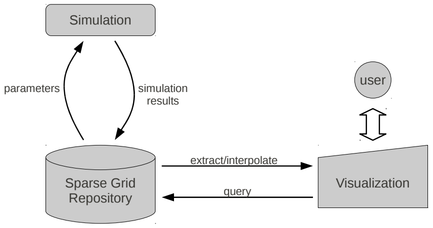


Figure 1. Workflow for interactive exploration and visualization with sparse grid surrogate models.

Figure 1 shows the workflow of our exploration framework. To accomplish an interactive exploration, we cannot run a full numerical simulation for every parameter change. Hence, we employ model reduction and distinguish between Offline and Online phases. In the Offline phase, we pre-compute and store (full) simulation runs $\{u(\mu_1),\ldots,u(\mu_M)\} \subset \mathbb{R}^{\mathcal{N}}$ for several parameter configurations $\mathcal{P}_S = \{\mu_1,\ldots,\mu_M\} \subset \mathcal{P} \subset \mathbb{R}^d$ in a sparse grid repository and build a surrogate model $\tilde{u} : \mathcal{P} \to \mathbb{R}^{\mathcal{N}}$ based on sparse grid interpolation or regression. The dimension $d$ equals the number of parameters and is usually in the range $1 < d < 10$. The surrogate model $\tilde{u}$ is by far cheaper to evaluate than the full model, but we still assume that $\tilde{u}(\mu)$ approximates the full simulation $u(\mu)$ sufficiently well for any $\mu$ in the parameter domain $\mathcal{P}$. In contrast to many other model reduction methods, we do not have to change the underlying simulation code to build our surrogate model. In practice, this is a huge advantage because it cannot be expected that engineers – working years on their simulation code – change them in order to use a visualization framework. In the Online phase, the user has access to the sparse grid surrogate model through the visualization block as shown in Fig. 1. By changing the parameters (e.g. through slide bars), an approximation of the corresponding simulation is immediately computed and displayed.

The sparse grid surrogate model is a linear combination of hierarchical (or multi-level) basis functions. The absolute values of the hierarchical coefficients of this linear combination can be related to the intensity of changes in the simulation entailed by changes in the parameters. By appropriately visualizing these coefficients, we can hint

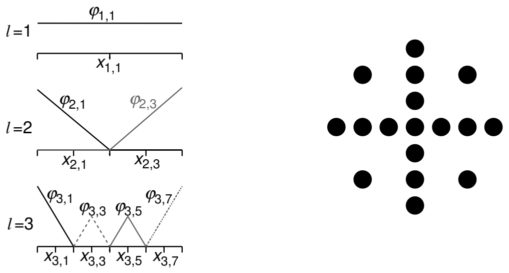


Figure 2. One-dimensional hierarchical basis (left) and regular sparse grid of level 3 (right).

regions in the parameter domain $\mathcal{P}$ where the surrogate $\tilde{u}$ changes heavily. This indicates interesting parameter regions which should be explored in more detail and thus is an important tool for the engineer to better understand the simulation or fine-tune parameter configurations.

In [2] an overview of data mining methods for simulation data can be found. There a bridge simulation is investigated with similar goals – efficiency and insight. It follows a more standard data mining approach while we propose methods from the model reduction community for more robustness and efficiency. In [3], surrogate models are coupled to visual analytics tools but with the purpose of semi-automated optimization.

## II. Sparse Grids

In data mining it is very common to represent a function as a linear combination of kernel functions centered at data points. We employ a grid-based approach, i.e., we represent a function $f_N$ as a linear combination with coefficients $\alpha_i$,

$$f_N(x) = \sum_{i=1}^{N} \alpha_i \phi_i(x)\,, \tag{1}$$

where the basis $\Phi := \{\phi_i\}_{i=1}^N$ stems from a grid and spans the function space $V_N$. Straightforward conventional discretizations with $N$ grid points in each dimension are not feasible in higher dimensions because the number of grid points is of the order $\mathcal{O}(N^d)$, depending exponentially on the dimension $d$. Note that in our case the dimension $d$ equals the number of parameters of our simulation, cf. Sec. I. For sufficiently smooth functions, sparse grids enable one to reduce the number of grid points by orders of magnitude to only $\mathcal{O}(N \log(N)^{d-1})$ while keeping a similar accuracy as in the full grid case [1].

The underlying principle of sparse grids is a hierarchical system of basis functions, see Fig. 2 (left), which is then extended to the $d$-dimensional case via a tensor product approach. A basis function is denoted by $\phi_{l,i}$ where $l$ is the level and $i$ the index. In the $d$-dimensional case the level $l = (l_1, \ldots, l_d)$ and index $i = (i_1, \ldots, i_d)$ become vectors and the corresponding basis function $\phi_{l,i}$ is the product of the one-dimensional basis functions $\phi_{l_1,i_i}, \ldots, \phi_{l_d,i_d}$. This leads to a set of subspaces $W_l$, the hierarchical increments, for which the grid points are the Cartesian product of the one-dimensional ones on the respective one-dimensional levels. We can select only those subspaces that contribute most to the overall solution. The solution of this optimization problem for a certain discretization level $\ell$ is the space

$$V_\ell^{(1)} := \bigoplus_{|l|_1 \leq \ell + d - 1} W_l\,, \tag{2}$$

if the error is measured in the $L_2$- or maximum norm [1] and where $|l|_1$ denotes the sum of the one-dimensional levels. A regular sparse grid of level 3 is shown in Fig. 2 (right).

We can use spatial (local) adaptivity to further reduce the number of unknowns needed to solve a problem up to some required accuracy. We start with a rather coarse sparse grid and use a suitable adaptivity criterion to add points in those regions of the domain that are most important. A simple – though typically very effective – criterion for adaptive refinement is to select the refinement candidates with the highest absolute values of their hierarchical coefficients $\alpha_i$.

## III. Sparse Grid Interpolants as Surrogates

In [4] we proposed to use a sparse grid interpolant as a surrogate model for high-resolution computational steering. This method is non-intrusive in the sense that the underlying simulation code does not have to be changed but is treated as a black box which delivers a snapshot $u(\mu) \in \mathbb{R}^{\mathcal{N}}$ for a requested parameter configuration $\mu \in \mathcal{P}$. The evaluation procedure of the sparse grid interpolant in the Online phase, i.e., when the user works with the model, can be efficiently parallelized and thus the sparse grid surrogate admits an interactive visual exploration.

We discretize the parameter domain $\mathcal{P}$ with a sparse grid and compute the snapshots $u(\mu_1), \ldots, u(\mu_M) \in \mathbb{R}^{\mathcal{N}}$ where the parameters $\mu_1, \ldots, \mu_M \in \mathcal{P}$ have to be the sparse grid interpolation points. Thus, the number of sparse grid points $N$ equals the number of snapshots $M$. We can afford to compute reasonably many snapshots because these computations are done in the Offline phase. We are then looking for a map $\tilde{u} : \mathcal{P} \to \mathbb{R}^{\mathcal{N}}$ such that this surrogate model $\tilde{u}$ approximates the full simulation $u$. Here, the map $\tilde{u}$ is a vector-valued sparse grid function where each component function $\tilde{u}^1, \ldots, \tilde{u}^{\mathcal{N}} : \mathcal{P} \to \mathbb{R}$ can be represented as a linear combination

$$\tilde{u}^i(\mu) = \sum_{j=1}^{N} \alpha_j^i \phi_j(\mu), \tag{3}$$

where $\phi_1, \ldots, \phi_N$ are the hierarchical basis functions and the coefficients $\alpha_1^i, \ldots, \alpha_N^i$ are the hierarchical coefficients corresponding to the $i$-th component function $\tilde{u}^i$. Let $u^i(\mu)$ be the $i$-th node of the snapshot $u(\mu)$ with parameter $\mu$. The $i$-th component function $\tilde{u}^i$ of $\tilde{u}$ is the interpolant of the pairs $\left\{ \left(\mu_1, u^i(\mu_1)\right), \ldots, \left(\mu_M, u^i(\mu_M)\right) \right\}$, where the parameters

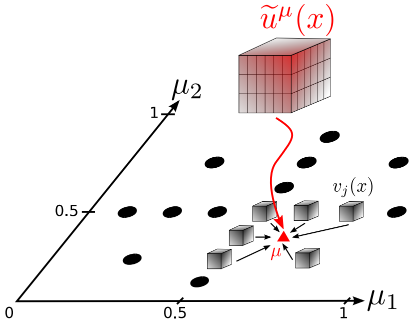


Figure 3. An example for a sparse grid interpolation: the interpolated value at the marked (triangle) parameter $\mu$ is a weighted sum of hierarchical coefficients (small cubes).

$\mu_1, \ldots, \mu_M$ are the sampling points and the values at the $i$-th node of the corresponding snapshots are the function values. This is a multi-dimensional interpolation problem because the parameter domain is usually a subset of $\mathbb{R}^d$ with $d > 1$ and thus sparse grids are required.

In the context of sparse grids, interpolation corresponds to a basis transformation from the ordinary nodal point basis into the hierarchical basis, see Fig. 2. This is done with a process called hierarchisation which is linear in the number of sparse grid points. We perform the hierarchisation process for each component function $\tilde{u}^1, \ldots, \tilde{u}^{\mathcal{N}}$ and obtain the hierarchical coefficients for each component. We refer to [4], [1] for more information about the hierarchisation and its implementation details.

Usually, the hierarchical coefficients give a good indicator where to refine a sparse grid, see Sec. II. Here, however, we have vector-valued sparse grid functions and thus not only one hierarchical coefficient per sparse grid point but a vector of length $\mathcal{N}$ of coefficients. In order to determine which sparse grid point should be refined, we compute the $L_2$-norm of the coefficient vectors and refine the sparse grid point corresponding to the highest value. This step can be repeated until accuracy requirements are satisfied. Our examples in Sec. VI show that this simple refinement criterion works very well.

In the Online phase, we can evaluate the surrogate model $\tilde{u}(\mu)$ at any parameter $\mu \in \mathcal{P}$ and so obtain an approximation of $u(\mu)$. Figure 3 shows the coefficient vectors (smaller cubes) and sparse grid points which need to be collected to interpolate between the snapshots at the evaluation point marked with a triangle.

## IV. Regression-Based Surrogates

In order to employ the sparse grid interpolation as described in the previous section, the sampling points, i.e., the parameters of the snapshots, have to coincide with sparse grid points. This means that we have to know the sparse grid points at the time we set up the data repository. Even though this is usually the case, it might happen that we already have a data repository with snapshots with arbitrary parameter configurations. A straightforward solution is to employ regression instead of interpolation. However, in the context of sparse grids, regression is by far more computationally expensive than interpolation and thus we cannot afford to perform a regression for each node of $u(\mu)$.

We introduce a pre-processing step and do not perform sparse grid regression directly onto the function values but apply Proper Orthogonal Decomposition (POD) (or PCA) first. We then approximate the data as a linear combination of only a couple of POD basis vectors. Let $S \in \mathbb{R}^{\mathcal{N} \times M}$ be the matrix of $M$ snapshots with $\mathcal{N}$ nodes each and $\mathcal{P}_S = \{\mu_1, \ldots, \mu_M\} \subset \mathcal{P}$ the corresponding parameter configurations. The orthonormal POD basis vectors are given by the columns of $V = [v_1, \ldots, v_r] \in \mathbb{R}^{\mathcal{N} \times r}$ of the singular value decomposition of $S = V \Sigma W^T$. Note that it is necessary to ensure that the data matrix $S$ has zero mean. With the coefficients $S^T V = \mathcal{A} = [\alpha^1, \ldots, \alpha^r] \in \mathbb{R}^{M \times r}$ we can represent the snapshots in $S$ as a linear combination of the POD basis $[v_1, \ldots, v_r]$ as

$$u(\mu_j) \approx \sum_{i=1}^{r} \alpha_j^i v_i, \tag{4}$$

where $\alpha_j^i$ is the $j$-th component of the $i$-th column of $\mathcal{A}$. In general, a snapshot $u(\mu)$ is approximated as

$$u(\mu) \approx \sum_{i=1}^{r} f_i(\mu) v_i, \tag{5}$$

where the functions $f_1, \ldots, f_r : \mathcal{P} \to \mathbb{R}$ determine the coefficient of the respective basis vector $v_i$ for a parameter $\mu \in \mathcal{P}$. Here, we choose the functions $f_1, \ldots, f_r \in V_\ell^{(1)}$ to be sparse grid functions. For each POD basis vector $v_1, \ldots, v_r$ we define a training set $T_i := \left\{(\mu_j, \alpha_j^i) | 1 \leq j \leq M\right\}$ which contains $M$ tuples of a parameter $\mu_j$ and the corresponding coefficient $\alpha_j^i$ for the $i$-th POD basis vector. We can easily compute this training set because we have the coefficient matrix $\mathcal{A} = [\alpha^1, \ldots, \alpha^r]$. We then want to find a sparse grid function $f^i = \sum_j \beta_j^i \phi_j$ such that

$$\min_{f^i \in V_\ell^{(1)}} \frac{1}{M} \sum_{k=1}^{M} \left(f^i(\mu_k) - \alpha_k^i\right)^2 + \lambda \sum_j \left(\beta_j^i\right)^2, \tag{6}$$

where $\sum_j \left(\beta_j^i\right)^2$ is a regularization term controlled by $\lambda > 0$ to enforce a smooth solution. The regression boils down to solving the system of linear equations $(BB^T + \lambda I)\beta^i = B^T \alpha^i$ where $\beta^i$ are the coefficients of the sparse grid function $f_i = \sum_j \beta_j^i \phi_j$ and $B_{jk} = \phi_j(\mu_k)$, see e.g. [5]. Note that the matrix $(BB^T + \lambda I)$ is of size $N \times N$ where $N$ is the number of sparse grid points. Thus, the size of the system matrix does neither depend on the number of snapshots $M$ nor on the number of nodes $\mathcal{N}$ of a simulation. The procedure described so far comprises the Offline stage. Hence, the result of the Offline stage are the

POD basis vectors $[v_1, \ldots, v_r]$ and the coefficient functions $f_1, \ldots, f_r : \mathcal{P} \rightarrow \mathbb{R}$.

To obtain an approximated simulation $\tilde{u}(\mu)$ in the Online stage for a parameter $\mu \in \mathcal{P}$, we have to evaluate all functions $f_1, \ldots, f_r$ and form the linear combination (5). It is cheap to evaluate this linear combination, because the sparse grid evaluation does not depend on the number of snapshots $M$ nor on the number of nodes $\mathcal{N}$ of a simulation.

## V. Visual Tools

So far, with the sparse grid surrogate, approximated solutions can be interactively delivered to the exploring engineer. In this section, we address the second issue, namely, which area of the large parameter space should the explorer mostly consider.

By applying the surrogate models based on sparse grid interpolation or sparse grid regression described in the previous two sections, we obtain in the one or other way a function which has a hierarchical structure given as a linear combination of hierarchical basis functions with hierarchical coefficients. A hierarchical coefficient corresponding to a grid point (or basis function) is a crucial piece of information as it indicates how important that grid point is for the function: A small absolute value can only lead to a small change in the function, whereas a large value means that the basis function significantly influences the function. This phenomenon is amplified by the multi-level structure of the hierarchical basis. A through discussion of these properties of the hierarchical basis can be found in e.g. [5], [1].

As presented in Sec. II, the sparse grid can be constructed in an adaptive manner. The adaptivity is based on the hierarchical coefficients and the refinement will focus on areas in the parameter domain of intensive function changes. Hence, the structure of the refined grid contains valuable information about the influence of parameters.

We make use of both indicators – hierarchical coefficients and refinement – in visual analytics tools. First, we employ scatter plot matrices to visualize the multi-dimensional sparse grid. A scatter plot matrix is a scheme of scatter plots where each column contains the same $X$ axis and each row the same $Y$ axis. For a sparse grid each plot is a projection of all high-dimensional grid points onto two dimensions. Thus, due to the refinement criterion, the number and location of the projected points indicates where, with respect to the projected parameters, most changes happen. In Sec. VI we enhance the scatter plot by drawing the grid points as spheres whose sizes correspond to the hierarchical coefficient. This shows the relative importance of each point and the magnitude of change.

Another visual analytics tool we consider is a weighted parallel coordinates plot. To plot a set of $d$-dimensional points, $d$ axes are drawn vertically and equally spaced. A point is represented as a polyline with vertices on the parallel axes; the position of the vertex on the $i$-th axis corresponds to the $i$-th coordinate of the point. In our case, the axes are given by the parameters of the simulation. Each sparse grid point can then be represented by a polyline connecting each axis (dimension) at the value of the corresponding parameter. We further extend this plot by weighting and color coding each polyline with the size of the hierarchical coefficient.

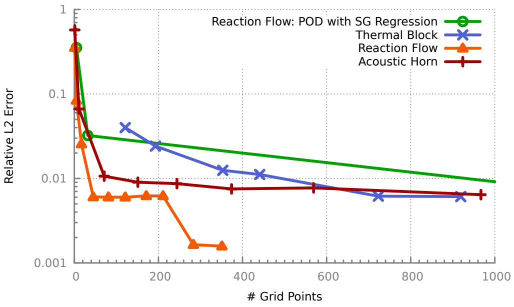


Figure 4. Accuracy of sparse grid surrogates for the thermal block, reaction flow and acoustic horn problems. Also exemplified, accuracy of POD with sparse grid regression for the reaction flow.

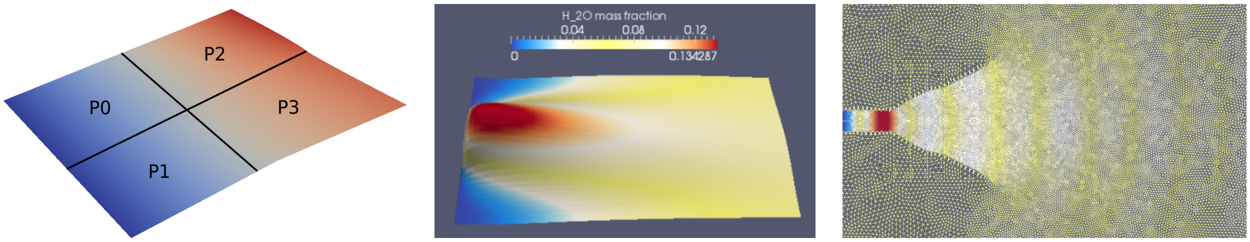
## VI. Applications

We apply the workflow shown in Fig. 1 to three demonstrator applications of computational science and engineering: a heat conduction problem (thermal block), a chemical reaction simulation (reaction flow) and a shape optimization problem (acoustic horn). The purpose is feature detection and exploration for which we employ the sparse grid surrogate models with the visual analytics tools of the previous section. The examples are quite different with respect to the number of parameters (2 - 5) and type of parameters (physical and geometrical).

As far as the accuracy of the sparse grid surrogate models is concerned, we know that for sufficiently smooth functions the $L_2$- and $L_\infty$-error is similar to that of a full grid [1]. Furthermore, we employ adaptive sparse grids and spend only grid points where they are needed which has proven to be very effective [5], [4], [6]. As Fig. 4 shows, for all of the following examples, a relative $L_2$-error of less than one percent is achievable. This is true for the sparse grid interpolation and regression and is sufficient for identifying important parameters or parameter ranges as well as for visual exploration. In the following examples, we employ sparse grid interpolation.

### A. Thermal Block

The thermal block problem describes steady-state heat conduction in a square domain consisting of a regular array of $2 \times 2$ square blocks, i.e., regions of different thermal conductivities, see Fig. 5(a). The problem has four parameters, $\mu \in \mathbb{R}^4$, where each component of $\mu$ is the thermal conductivity of one region. A thorough discussion of the thermal block can be found in [7].



(a) thermal block

(b) reaction flow

(c) acoustic horn

Figure 5. Snapshots of the thermal block, reaction flow and acoustic horn simulations

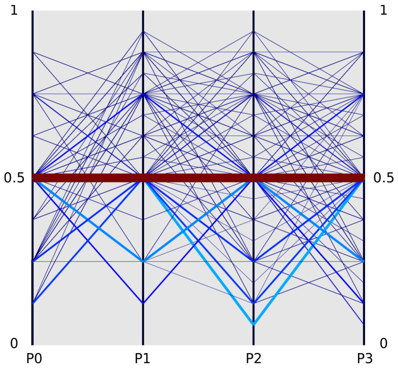


Figure 6. The parallel coordinates for the thermal block show that the center grid points is the most important grid point.

The sparse grid for the thermal block is four-dimensional and consists of 367 grid points. In the weighted parallel coordinates plot in Fig. 6, we see more refinement in the upper half of the parameter ranges which corresponds to faster heat dissipation (high conductivity) and thus more change. In the lower parameter ranges there are a few points with higher hierarchical coefficients. These differ significantly from the central point (on level 1) and are thus candidates for a closer investigation.

## B. Reaction Flow

The reaction flow is a two dimensional $H_2$-Air flame described by the reaction mechanism $2H_2 + O_2 \rightarrow 2H_2O$, see [8]. Basically, hydrogen and oxygen are mixed and ignited by a flame. Fig. 5(b) shows that the concentration of the resulting product (water) is higher when closer to where the fuel and the oxidizer are mixed and ignited. The reaction flow is parameterized by the two parameters $A$ (pre-exponential factor, i.e. rate of molecule collisions) and $E$, energy to overcome in order for the chemical reaction to occur. Of interest in gaining insight in this simulation is the relation of the two parameters to each other and which parameter ranges capture the most changes in the reaction system.

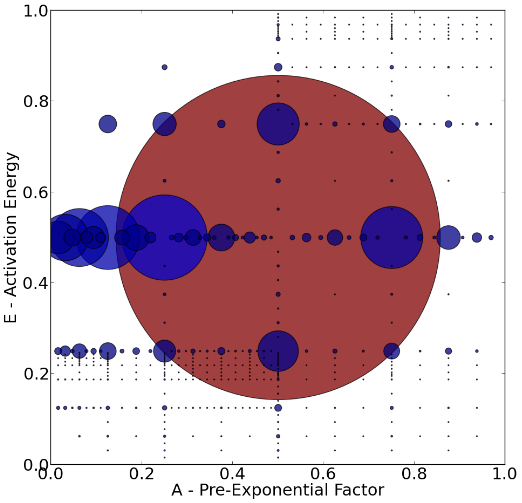


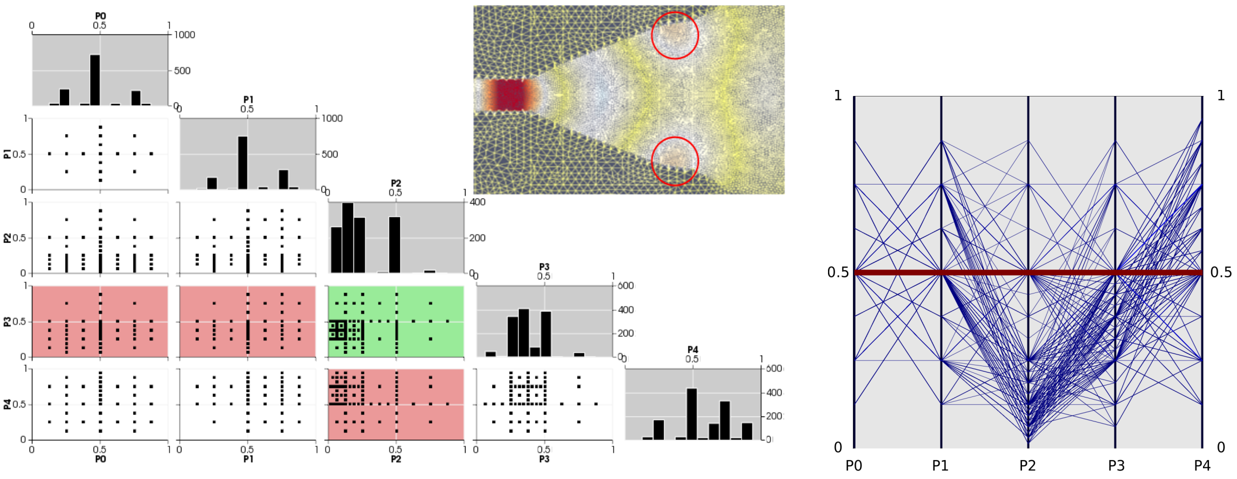
Figure 7. The weighted scatter plot matrix for the reaction flow.

The sparse grid for the reaction flow is two-dimensional and has 445 points. From the weighted scatter plot matrix in Fig. 7 we identify one area (upper left) with high activation energy and low pre-exponential factor where little refinement takes place. This corresponds to the physical interpretation that in the above scenario and within the chosen parameter ranges, the reaction rate is low and thus there is little change in the simulation function. However, if the pre-exponential factor $A$ is increased (molecule collisions are more frequent) then also the reaction rate increases. The weighted parallel coordinates plot in Fig. 8 confirms the above observations and encourages the investigation around $E = 0.2$.

## C. Acoustic Horn

The acoustic horn problem shown in Fig. 5(c) consists of a planar channel, i.e., the waveguide, onto which a conical termination, i.e., the horn, is attached. An incident wave is generated from the far left of the waveguide and propagated through the horn [9]. We are interested in the influence of the shape of the horn onto the wave reflexion. The geometry of the horn is parametrized at five points along its boundaries. With the parameters, we can locally widen and narrow the opening of the horn.

The sparse grid for the acoustic horn is five-dimensional with 1327 points. From the scatter plot matrix in Fig. 9(a) we observe a refinement concentration in the $[P_2, P_3, P_4]$ plots,



(a) Acoustic Horn: (left) weighted scatter plot matrix, (upper right) simulation snapshot shows pressure increase toward the walls.

(b) Acoustic Horn: Parallel Coordinates.

Figure 9. Visual analytics for the acoustic horn example.

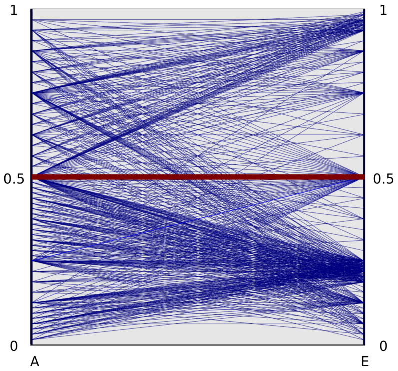


Figure 8. Parallel coordinates for the reaction flow example. We see a clear cluster around 0.2 for the activation energy $E$.

which indeed corresponds to an increased pressure in the neighborhood of the $P_2$ and $P_3$ deformation points on the cone. Fig. 9(b) confirms $P_2$, $P_3$ and $P_4$ cause most changes in the simulation and that, again, the center point captures most of the simulation behavior. Moreover, we observe that the combination of $P_2$ and $P_3$ shifted down and $P_4$ shifted up leads to an increased pressure (thus reflexion). In the upper right corner of Fig. 9(a) a snapshot with parameters from that area is plotted which clearly shows an increased pressure near the wall of the horn (marked with red circles).

## VII. Conclusion

We have described a workflow based on sparse grids for the interactive visualization and exploration of simulation data. By employing sparse grid surrogate models, we achieve quick response and thus interactive exploration. Sparse grids heavily rely on the hierarchical (multi-level) basis. We have exploited the corresponding hierarchical coefficients and refinement patterns in order to reveal parameter dependencies and parameter regions of interest by plotting them with visual analytics tools such as parallel coordinates or scatter plot matrices. The examples from computational science and engineering have shown that the workflow identifies physical properties and provides valuable insight.

## References

[1] H.-J. Bungartz and M. Griebel, “Sparse grids,” *Acta Numerica*, vol. 13, pp. 147–269, 2004.

[2] S. Burrows, B. Stein, J. Frochte, D. Wiesner, and K. Müller, “Simulation data mining for supporting bridge design,” in *Proceedings of the 11th AusDM*, 2011.

[3] K. Matkovic, D. Gracanin, M. Jelovic, and Y. Cao, “Adaptive interactive multi-resolution computational steering for complex engineering systems,” in *EuroVA*, 2011.

[4] D. Butnaru, D. Pflüger, and H.-J. Bungartz, “A parallel and distributed surrogate model implementation for computational steering,” in *In Proceeding of the 11th ISPDC*, 2012.

[5] D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. München: Verlag Dr. Hut, 2010.

[6] D. Butnaru, G. Buse, and D. Pflüger, “Towards high-dimensional computational steering of precomputed simulation data using sparse grids,” *Procedia CS*, vol. 4, pp. 56–65, 2011.

[7] G. Rozza and A. Patera, “The heterogeneous thermal block problems,” MIT, Tech. Rep., 2008.

[8] M. Buffoni and K. Willcox, “Projection-based model reduction for reacting flows,” in *40th Fluid Dynamics Conference and Exhibit*, 2010.

[9] E. Bängtsson, D. Noreland, and M. Berggren, “Shape optimization of an acoustic horn,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 11–12, pp. 1533 – 1571, 2003.