Challenges in Visual Analysis of Ensembles

Patricia Crossno
Sandia National Laboratories

Editor:
Theresa-Marie Rhyne
theresamarierhyne@
gmail.com

Modeling physical phenomena through computational simulation increasingly relies on generating a collection of related runs, known as an ensemble. This article explores the challenges we face in developing analysis and visualization systems for large and complex ensemble data sets, which we seek to understand without having to view the results of every simulation run. Implementing approaches and ideas developed in response to this goal, we demonstrate the analysis of a 15K run material fracturing study using Slycat, our ensemble analysis system.

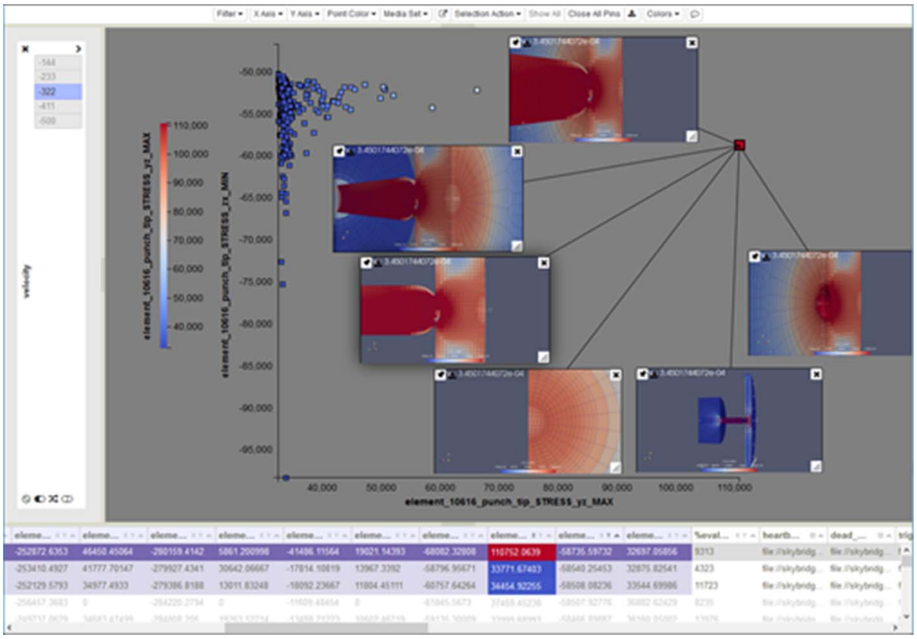
Traditional scientific visualization systems provide tools to explore and compare no more than a handful of simulation results simultaneously. Rather than scaling in the number of results, these systems have been designed to scale relative to problem size, focusing on interactive visualization of extremely large results data. However, there are some important types of questions that cannot be answered with just a few simulations. In particular, sensitivity analysis, uncertainty quantification, and parameter studies all rely on generating a collection of simulation runs, typically referred to as an *ensemble*. An ensemble is a set of related runs, each of which is a sample within a shared problem space. Ensemble data is typically large, consisting of paired sets of input parameters and simulation results. The inclusion of the input parameters (or a set of independent variables) is a key distinction that separates ensemble research from research into techniques for comparing collections of results data.

With increases in the availability and scale of High Performance Computing (HPC) platforms, ensembles are increasingly being used to evaluate and understand computational models. The trend has been towards growth in the numbers of runs within an ensemble, so scalability is a key requirement for designing systems to explore and interpret ensemble results. In Slycat,[1] an ensemble visualization and analysis system that we are developing at Sandia National Laboratories, we routinely work with ensembles containing thousands of runs. In the last year or two, we have started seeing ensembles with tens of thousands of runs.

Our goal in ensemble analysis is to enable an understanding of the ensemble without requiring the user to view each simulation's data. This is analogous to using text analysis to understand the content of a set of documents without reading them all. Beyond scalability issues, there are three

2469-7087/19/$33.00 © 2019 IEEE

aspects of ensemble analysis that make it difficult: the complex nature of the results data, the difficulty of defining analysis approaches that reveal high-level relationships between runs or between variables, and finding suitable visual representations at various levels of abstraction.



eleme...	eleme...	eleme...	eleme...	eleme...	eleme...	eleme...	eleme...	eleme...	eleme...	%eval...	hearth...	dead_...	trig
-252872.6353	46450.45064	-280159.4142	5861.200998	-41486.11564	19021.14393	-68082.32808	110752.0639	-58735.59732	32697.05856	9313	file://skybridg...	file://skybridg...	1
-253410.4927	41777.70147	-279927.4341	30642.06667	-17814.10819	13967.3392	-58796.95671	33771.67403	-58540.25453	32875.82541	4323	file://skybridg...	file://skybridg...	1
-252129.5793	34977.4933	-279386.8188	13011.83248	-18092.23667	11804.45111	-60757.64264	34454.92255	-58508.08236	33544.69986	11723	file://skybridg...	file://skybridg...	1
-256457.3683	0	-284220.2794	0	-11609.48454	0	-61845.5673	37459.45238	-58507.92776	36882.62429	8235	file://skybridg...	file://skybridg...	1
-249737.0629	34683.47499	-284808.205	19363.53774	-13488.21223	10609.46719	-63136.30009	[illegible]	-58466.89687	36160.05507	13976	file://skybridg...	file://skybridg...	1

Figure 1. Parameter Space Model of a 15K run ensemble to study material fracturing, filtered by velocity value on left, showing in situ generated images of an anomalous run, which have been retrieved from a remote computing system.

Depending on the simulation code and the problem being modeled, results data typically include multiple variables saved in a series of temporal snapshots. The variables can consist of scalar or vector values, or they can be a set of values arranged into more complex structures, such as volumes, surfaces, matrices, graphs, or trees. With the growth of in situ visualization, which renders images and videos as part of the simulation, we are seeing media-based results increasingly among the outputs. These results combine decreased capabilities for post-simulation control of views, variable selections, and visualization parameters with increased data fidelity, especially temporal fidelity, during image rendering.

How do we construct a mental model of such a diverse set of results and relate that back to the input parameters? To understand the ensemble as a whole, we need analysis methods that summarize the predominant behaviors of the overall group. This includes relationships between sets of runs, individual runs relative to the majority, correlations between variables, and how set membership or all of the preceding relationships change over time. The analysis techniques selected are highly dependent on the type of results data being compared and the type of relationships being evaluated. For example, Slycat uses hierarchical, agglomerative clustering for discovering set relationships within time series results, and Canonical Correlation Analysis[1] for correlating relationships between input variables and scalar outputs.

The last challenge is the visual representation. Representations of ensemble data are closely tied to the level of abstraction being shown, ranging from highly abstract to concrete. We define four levels of abstraction for viewing an ensemble: the entire ensemble, sets of runs, individual runs, and individual simulation results. A representation at the ensemble level shows trends or behaviors that are exhibited by most of the runs, including relationships between variables and their values. Relationships may include variable correlations, spatial distributions of values, spatial co-occurrences, or temporal shifts in any of the preceding. Abstractions for sets of runs display group relationships, such as similarities between clusters of responses. At the level of individual runs, the abstraction explicitly represents runs as separate entities in the visualization. Similarities and differences in runs relative to the broader group can reveal the robustness of solution spaces, outliers, or anomalies. Finally, the ability to drill-down to the level of an individual run and quantitatively examine its unique input and output data is important for understanding anomalies.

The choice of representation for a particular result type is also tied to the mental model of the user community. For instance, circuit simulations produce sequences of voltage and current values over time as outputs. Electrical analysts want to see and compare waveform shapes as graphs of amplitude over time, so at least one of the abstraction levels should present the data in this familiar form.

Interactions are as important as the representations in designing the interface. Representations should be linked to explore relationships across abstraction levels. At the ensemble level, the visual elements themselves can be used as affordances within the interface, executing operations over sets of simulations or acting as filters to enable rapid reduction in visual clutter. At the level of individual runs, each simulation can be linked to both ensemble level groups and to individual variable values. As information foraging[2] leads the analyst to a reduced set of simulations, group operations can be used to retrieve and compare sets of result variables.

SHARED ATTRIBUTE SPACES

To compare members of an ensemble, we need to find sets of variables shared by all the runs, such as a multi-variate set of scalar outputs, or a finite element mesh of the final time step in the simulation. We will refer to these variables as *shared attribute spaces*. Although it might seem that simulation ensembles could be easily constructed to simply generate common result sets, even outputs as simple as time series data may require transformations prior to comparison. For instance, the set of time series may not all span identical time intervals, sequences may contain differing numbers of values, or values may not be sampled at corresponding times. Consequently, the attribute space is a transformation of the data that is constructed to facilitate comparisons. For more complex output types, attribute spaces based on statistical measures or extracted features are necessary to reduce the dimensionality of the results.

In the research literature, we see that ensemble systems tailor their techniques to the types of results data being analyzed. Some good examples of ensemble analysis systems for various results types include: spatial data,[3] spatiotemporal weather/climate,[4] flow fields over time,[5] shape generation,[6] and visual effects design in movies.[7]

Although exploratory visual analysis may be done to gain insights into a data set, typically ensembles are generated to perform a specific analysis task. That task, combined with the results data type, determines the analysis techniques that are used. In their survey of parameter space analysis research, Sedlmair et al. identified six recurring analysis tasks: parameter optimization, result partitioning, fitting measured data, identifying outliers, quantifying uncertainty (results reliability), and sensitivity analysis of the results relative to changes in the inputs.[8]

Data fitting and uncertainty are especially difficult due to the challenges in constructing a shared attribute space between simulation data and measured real-world data. For example, the input parameters used by the simulation may not correspond to the set of independent variables in the experimental data, there may be calibration differences in measured data, or experimental results may consist of a set of photographs created under unknown conditions (lighting, lens characteristics, positioning). There is great demand for comparing simulation and real-world data, so we expect this will be an important research area in the near future.

SLYCAT MODELS

Although for scientists, each simulation run is a *model* of real-world phenomena given certain conditions, we use the term *model* to refer to our modeling of the ensemble data, not the physics. Within Slycat, a *model* encapsulates an attribute space, an analysis method, a set of visual representations, and a set of interactions between those representations.

Slycat models are each tailored to address specific results types, so different model types can provide complementary perspectives of data features when analyzing the same data set. Insights gained from each model combine to create a more complete mental model of the underlying problem domain.

We currently have models providing correlation analysis, parameter exploration, and time series clustering. Correlation analysis looks exclusively at scalar-valued table data and addresses the tasks of sensitivity analysis and outlier identification. Parameter exploration combines table data and media data, facilitating tasks of parameter optimization and outlier finding. Time series clustering operates on tables and temporal sequences, targeting the tasks of result partitioning and outlier detection. Because we want to map results back to input parameters, table data is a component of all Slycat model types.

The Slycat architecture integrates data management into our models. Typically, results consist of many files per run, distributed across multiple directories, stored on a remote file system. Analysis of even a single result variable across all runs can entail significant data management issues due to the size, locality, and organization of the results data. Minimizing data movement helps to mitigate these issues. Analysis is performed in parallel by remotely launching jobs through a Slycat Agent on the same HPC where the ensemble was created, as shown in Figure 2. This reduces the scale of data moved from the HPC to that of a much smaller set of analysis artifacts and a handful of results from selected sets of runs. Results, such as images or movies, are interactively pulled from the HPC and viewed remotely on the analysts' desktops via commodity web clients.

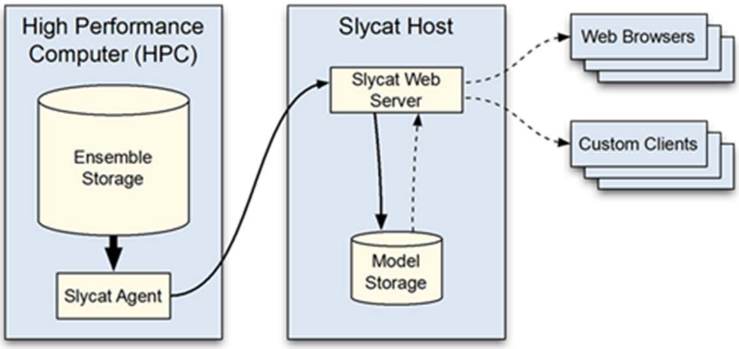


Figure 2. Slycat web-based architecture, designed for ensemble analysis and visualization. Ensemble data is stored on the HPC, where analysis is performed in parallel to reduce computation time and avoid data movement. Analysis results are sent to the Slycat server, where they form the basis of a Slycat model. Models combine analysis of specific results types with multiple visual representations, each with a different level of abstraction. Users remotely visualize and interact with these models through a standard web browser. Drill-down to view individual results data, such as images or movies, is interactively managed by the model through the Slycat Agent.

Prior to beginning this work, we interviewed analysts at Sandia to evaluate workflows, analysis needs, and available tools.[1] Slycat resulted from our identification of an unmet need for ensemble analysis. Our choices in targeted results data types, analysis tasks, and visual interfaces were made in response to requests from, and in collaboration with, our users.

MATERIAL FRACTURING PROBLEM

We use Slycat to analyze a solid mechanics problem, consisting of a 15K run ensemble. This ensemble explores changes in material responses as a punch impacts a plate (shown in Figure 3) under various conditions. The analysts are interested in understanding how different parameter combinations affect stresses and material fracturing around the point of impact. Over the course of the simulation, as stresses distort mesh cells beyond certain limits, those cells are removed from the mesh and marked as dead. The missing cells form the fracture.

Sierra/SolidMechanics[9] and ParaView Catalyst[10] are the simulation and in situ codes, respectively, used to generate the ensemble. Given the large number of simulation runs, we did not save volumetric results, choosing to save in situ generated outputs instead. For each of the 15K runs, the ensemble consists of 8 input parameters and 38 outputs (12 scalar results, 16 variables changing over time, 6 event-triggered images, and 4 movies of 1000 frames each). Although the total data size was small, only about a terabyte, the complexity is derived from the sheer number and variety of results. This ensemble is used in generating all the examples in the remainder of the paper.

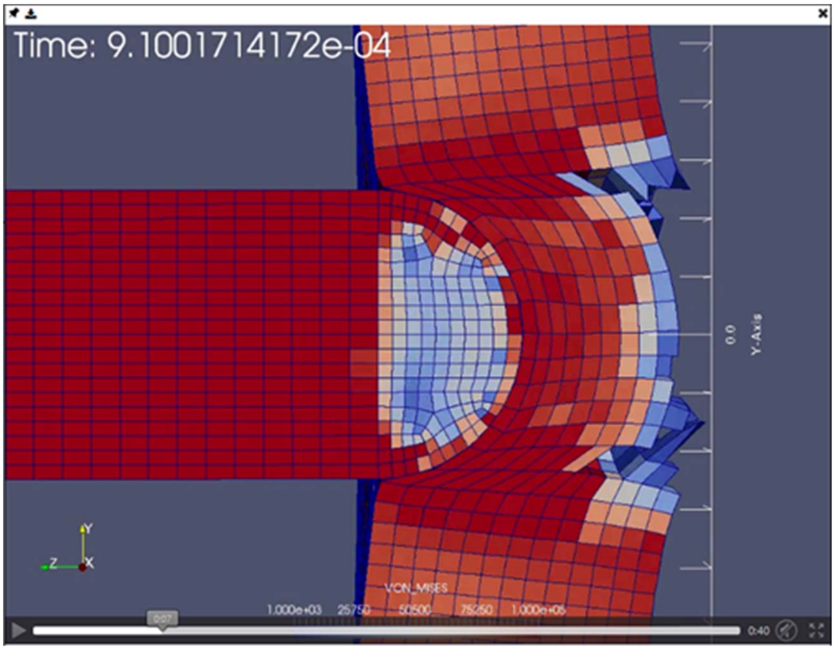


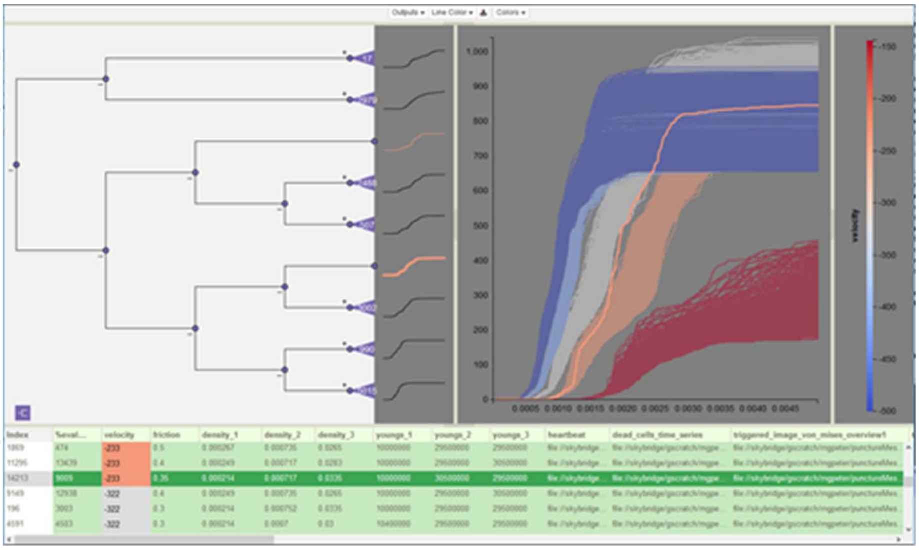
Figure 3. In situ generated movie frame showing material fracturing as a punch impacts a plate. Cells are colored by Von Mises stress. Dead cells have been removed from the mesh of the plate (near the blue region on the right side of the image).

We want to answer the following questions. Are there groups of runs that share fracturing behaviors? How many groups? Which runs are in each group? Do runs change groups over time? Do specific input values correspond to group membership? How do cell behaviors near the point of impact differ relative to group membership?

TIME SERIES MODEL

Since cell death is tightly coupled with fracturing behavior, we want to understand how cell death evolves over time. We use Slycat's Time Series Model to analyze a temporal result containing running totals for the number of dead cells, sampled at each time step (Figure 4). Note that the lines are color-coded by initial velocity values, which we chose because the results of analyzing the parameter table with Slycat's Canonical Correlation Analysis (CCA) Model showed that this input parameter has the strongest correlation with output metrics for stresses (see below).

Looking at the individual-run-level representation in the line plot view (Figure 4, upper right), where each line represents cell deaths for a single run, we see that there are four or five distinct groups. The red lines represent runs with the smallest initial velocities and the dark blue lines are those with the largest. We see that cell death, and hence fracturing, increases with the initial velocity of the punch.



index	%eval...	velocity	friction	density_1	density_2	density_3	youngs_1	youngs_2	youngs_3	heartbeat	dead_cells_time_series	triggered_image_von_mises_overview1
1869	474	-233	0.5	0.000267	0.000735	0.0265	10000000	29500000	29500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...
11295	13439	-233	0.4	0.000249	0.000717	0.0283	10000000	29500000	30500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...
14213	9009	-233	0.35	0.000214	0.000717	0.0336	10000000	30500000	29500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...
9149	12938	-322	0.4	0.000249	0.000735	0.0265	10000000	29500000	30500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...
196	3003	-322	0.3	0.000214	0.000752	0.0336	10000000	29500000	29500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...
4591	4503	-322	0.3	0.000214	0.0007	0.03	10400000	29500000	29500000	file://skybridge...	file://skybridge/gscratch/mgpe...	file://skybridge/gscratch/mgpeter/punctureMes...

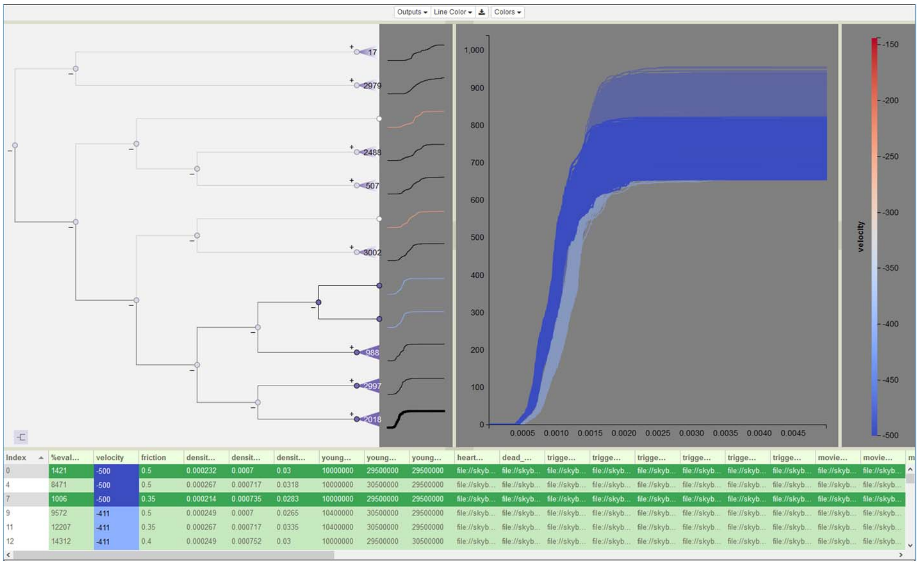
Figure 4. Time series analysis of cell deaths over time in 15k run fracture ensemble. Each run is color-coded by initial velocity value. In the line plot view on the right, the *x*-axis is simulation time and the *y*-axis is cumulative cell deaths. An anomalous run in orange is highlighted. Initially a member of the orange group, this run later shifts into the white group.

Changing to the Parameter Space Model to view videos from each group, we discover that a high-speed impact stretches and tears the plate cells, leading to earlier and greater numbers of cell deaths, while a slower impact does less cell damage (sometimes not even penetrating the plate), leading to fewer and later cell deaths (see Figure 7).

To better evaluate whether the dark and light blue runs form distinct groups, we switch to the ensemble-level abstraction (Figure 4, upper left), which uses agglomerative hierarchical clustering. We chose this analysis method because it does not require prior knowledge of the number of clusters, and the dendrogram representation because it mirrors the technique. The dendrogram is initially only drawn to a few levels, with most of the tree hidden in subtree icons, which are the purple triangles annotated with the number of the hidden leaves. Each subtree or leaf node is followed by a sparkline,[12] which is an exemplar of the plot shape for runs in that subtree. Dendrogram components act as controls, not only in the display of the tree itself, but also in the line plot and the table (Figure 5).

By highlighting one of the two dark blue subtrees (clicking on a sparkline selects its subtree members in both the line plot and the table), we can see that although the two groups initially have similar plots, the highlighted set of dark blue runs levels off with fewer overall deaths. As the dendrogram shows, the highlighted runs more closely match the light blue group (whose subtree is immediately above the highlighted group) than the non-highlighted dark blue set. By interactively manipulating the view, we can determine that there are only four unique groups of responses. We use the highlighted rows in the table to identify group members through their ID number.

The dendrogram can also be used to find outliers and anomalous runs. In Figure 5, two light blue outliers appear in the dendrogram above the three subtrees that we previously discussed. In Figure 4, the highlighted anomalous run in orange initially aligns with the orange group, then transitions to more closely match the behaviors of the white group. The dendrogram shows it as a lone run (fourth sparkline from the bottom) and links it with the white group's subtree below.



Index ▲	%eval...	velocity	friction	densit...	densit...	densit...	young...	young...	young...	heart...	dead_...	trigge...	trigge...	trigge...	trigge...	trigge...	trigge...	movie...	movie...	m
0	1421	-500	0.5	0.000232	0.0007	0.03	10000000	29500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	
4	8471	-500	0.5	0.000267	0.000717	0.0318	10000000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	
7	1006	-500	0.35	0.000214	0.000735	0.0283	10000000	29500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	
9	9572	-411	0.5	0.000249	0.0007	0.0265	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	
11	12207	-411	0.35	0.000267	0.000717	0.0335	10400000	30500000	29500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	
12	14312	-411	0.4	0.000249	0.000752	0.03	10000000	29500000	30500000	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	file://skyb...	

Figure 5. The darkened portion of the dendrogram represents the visible runs. The line plot and table views are reduced to show only the cell death evolution for the visible subtree. Within those runs, the bottom subtree with the darker sparkline has been selected, highlighting its member runs in both the line plot and the table.

CANONICAL CORRELATION ANALYSIS MODEL

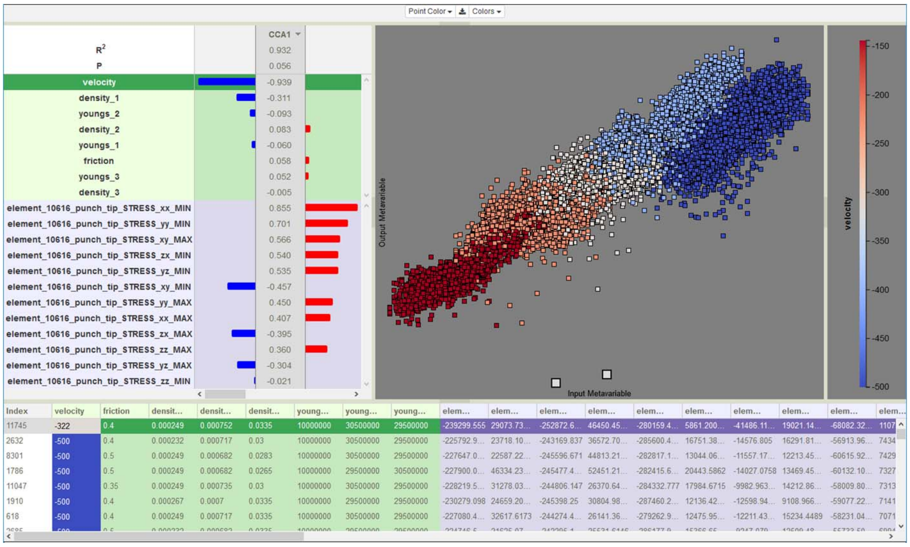
It is not enough to cluster temporal results in the Time Series Model, we want to understand how differences in the input parameters are related to changes in the responses. Although we could have stepped through color-coding the line plots by each of the input variables to discover the relationship between initial velocity and cell deaths, a more efficient approach is to use the Canonical Correlation Analysis (CCA) Model to explore relationships between variables.[11]

CCA operates on tables of scalar values, providing correlations between two sets of multivariate data, such as between sets of input parameters and output metrics, which form the shared attribute space. We selected this analysis method to perform sensitivity analysis because it maps well to the duality of our data space and provides a many-to-many analysis of the variables, i.e. it simultaneously correlates all the inputs to all the outputs (see our earlier paper for more detail).[1]

CCA can be used for sensitivity analysis, determining which input variables are the primary drivers of the results, the strength of the correlations, and whether the correlations are negative or positive. The correlations between variables form the ensemble-level abstraction and are shown in the bar chart in the upper left of Figure 6.

We chose a bar chart for the representation because we wanted to encode both the magnitude and type of correlation so they could be understood pre-attentively. Although the weight values are written in the central column, they are also encoded in the bar length. Color and orientation of the bars redundantly encode the positive or negative correlation between rows of variables, with shared color between input and output variable rows representing positive correlations and differing color showing negative correlations. Variables are sorted in order of decreasing significance (weight) in each of the input and output sets.

In Figure 6, we can see at a glance that there are only two significant input parameters, velocity and density_1, which are negatively correlated with most of the output metrics (i.e. axis-aligned minimum and maximum stress values at the punch tip).



	CCA1
R^2	0.932
P	0.056
velocity	-0.939
density_1	-0.311
youngs_2	-0.093
density_2	0.083
youngs_1	-0.060
friction	0.058
youngs_3	0.052
density_3	-0.005
element_10616_punch_tip_STRESS_xx_MIN	0.855
element_10616_punch_tip_STRESS_yy_MIN	0.701
element_10616_punch_tip_STRESS_xy_MAX	0.566
element_10616_punch_tip_STRESS_zx_MIN	0.540
element_10616_punch_tip_STRESS_yz_MIN	0.535
element_10616_punch_tip_STRESS_xy_MIN	-0.457
element_10616_punch_tip_STRESS_yy_MAX	0.450
element_10616_punch_tip_STRESS_xx_MAX	0.407
element_10616_punch_tip_STRESS_zx_MAX	-0.395
element_10616_punch_tip_STRESS_zz_MAX	0.360
element_10616_punch_tip_STRESS_yz_MAX	-0.304
element_10616_punch_tip_STRESS_zz_MIN	-0.021

Index	velocity	friction	densit...	densit...	densit...	young...	young...	young...	elem...	elem...	elem...	elem...	elem...	elem...	elem...	elem...	elem...	elem...
11745	-322	0.4	0.000249	0.000752	0.0335	10000000	30500000	29500000	-239299.555	29073.73...	-252872.6...	46450.45...	-280159.4...	5861.200...	-41486.11...	19021.14...	-68082.32...	1107
2632	-500	0.4	0.000232	0.000717	0.03	10000000	30500000	29500000	-225792.9...	23718.10...	-243169.837	36572.70...	-285600.4...	16751.38...	-14576.805	16291.81...	-56913.96...	7434
8301	-500	0.5	0.000249	0.000682	0.0283	10000000	30500000	29500000	-227647.0...	22587.22...	-245596.671	44813.21...	-282817.1...	13044.06...	-11557.17...	12213.45...	-60615.92...	7429
1786	-500	0.5	0.000249	0.000682	0.0265	10000000	29500000	30500000	-227900.0...	46334.23...	-245477.4...	52451.21...	-282415.6...	20443.5862	-14027.0758	13469.45...	-60132.10...	7327
11047	-500	0.35	0.000249	0.000735	0.03	10000000	30500000	29500000	-228219.5...	31278.03...	-244806.147	26370.64...	-284332.777	17984.6715	-9982.963...	14212.86...	-58009.80...	7313
1910	-500	0.4	0.000267	0.0007	0.0335	10000000	29500000	29500000	-230279.098	24659.20...	-245398.25	30804.98...	-287460.2...	12136.42...	-12598.94...	9108.966...	-59077.22...	7141
618	-500	0.4	0.000249	0.000717	0.0335	10000000	30500000	29500000	-227080.4...	32617.6173	-244274.4...	26141.36...	-279262.9...	12475.95...	-12211.43...	15234.4489	-58231.04...	7071

Figure 6. Correlation model for 15K run fracture problem. Ensemble level variable correlations are shown in the bar chart (shared bar color between variables indicates positive correlation, differing color is a negative correlation; bar length shows correlation strength). The scatterplot provides individual level relationships between each run and the weightings in the bar chart, with two outliers in white at the bottom. Runs are color-coded by initial velocity value.

The scatterplot provides the individual-run level of abstraction with each point representing a run. This view shows the correspondence between the bar chart's high-level relationships and each simulation run. The coordinate space is highly abstract, with the x-axis representing a metavariable that is the sum of all the input parameters and the y-axis a metavariable of all the scalar outputs. The x and y coordinates for each run are computed as the weighted sums of input and output values, respectively. Runs whose variable relationships exactly match the CCA weightings will lie along the diagonal. Runs whose variable relationships do not match will appear off the diagonal, such as the two white-colored outliers in Figure 6.

This view can be used detect runs whose variable values or relationships are outliers or anomalous. Because the point coordinates depend upon all variables simultaneously, this view reveals outliers resulting from interactions between multiple variables, which would not be discovered by simple thresholding.

The table at the bottom provides drill-down to variable values within individual runs. All three views are linked by selected variable (bar chart row, scatterplot color, and table column). The scatterplot and table are linked by selected run (point and table row). These interactions facilitate searching for patterns and relationships, such as identifying that the anomalous points in Figure 6 represent simulations with extreme values at the punch tip, one in a maximum stress variable, and the other in a minimum stress variable.

PARAMETER SPACE MODEL

The previous two models have provided very abstract representations of simulation results. Although image and movie outputs provide familiar representations of volumetric data, without the context of the associated simulation parameters and the ability to simultaneously view and compare multiple instances, media results rapidly become incomprehensible. The Parameter Space Model provides this context.

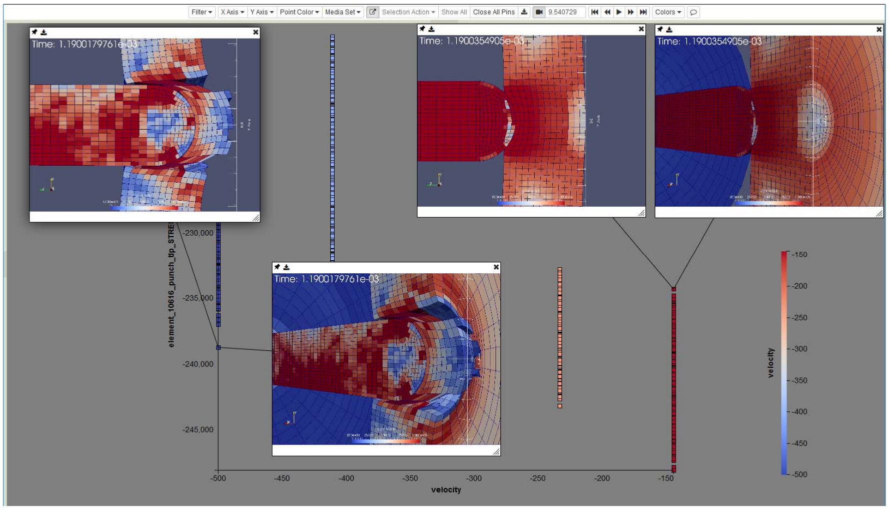


Figure 7. Parameter Space Model with synched videos to examine differences between high initial velocity (linked to blue point on the left) and low initial velocity (linked to red point on the right). Cells in the high velocity video are stretched and dying. The plate is not being penetrated in low velocity case.

Like CCA, the Parameter Space Model operates on table data, but it additionally provides filtering and remote access to in situ generated media results, along with group-based operations for interacting with them. The shared attribute space is the parameter table combined with media types. This model can be used for tasks such as parameter optimization, result partitioning, and identifying outliers. Other than gathering summary statistics for the variables, no analysis is done by the system. Instead, this model is an exploratory interface to assist the user in finding patterns.

Filters are the ensemble level representation (Figure 1), showing either a set of buttons, if the variable values are discrete as in Latin hypercube sampling, or a range slider. Summary information is only displayed if a filter is active. Filters are predominantly scatterplot controls to reduce visual clutter.

The individual-simulation-level abstraction embodied in the scatterplot is the core representation. We chose scatterplots because we wanted to interact with media from large numbers of runs, and points use very few pixels. Hovering over individual points (or selecting groups of points to enable a group interface) retrieves images and movies from remote data stores, providing rapid viewing of media outputs that are linked to parameter information. Although overplotting can be an issue with this representation, the scatterplot axes and point color-coding are interactively selectable, so the plot can be manipulated to expose interesting runs. Since outliers are often targets for drill-down, their unique position or color can be used to further down-select runs.

Interesting media can be *pinned* in the view to perform comparisons (Figure 7). For multiple pinned videos, *video synch* enables shared group controls for synchronized playback and single-stepping of all visible videos.

FINAL THOUGHTS

Currently, the application of multiple Slycat models to analyze the same ensemble is a manual operation. Each model must be independently created and operations are not shared between models. Given the power of combining multiple perspectives on a shared problem, we are working on mechanisms to link selection or visibility between models. The challenge is to do this in a web-based framework.

An additional issue is that although Slycat provides the data management for retrieving and viewing multiple videos using synchronized playback, screen real estate limits viewing to a handful of videos at a time. What we really need is a new model that will abstractly show video similarities over time, so that we can see the shifting relationships between video results without having to view them all.

ACKNOWLEDGEMENTS

I want to thank the Slycat team, since the system described in this paper is due to all their hard work: Warren Hunt, Shawn Martin, Alex Sielicki, Matt Letter, and Jaxon Gittinger. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

1. P.J. Crossno et al., "Ensemble Analysis of Electrical Circuit Simulations," *Topological and Statistical Methods for Complex Data*, Springer, pp. 279–294, 2015.
2. P. Pirolli and S. Card, "Information Foraging," *Psychological Review*, vol. 106, no. 4, 1999, pp. 643–675.
3. I. Kolesár et al., "A Fractional Cartesian Composition Model for Semi-Spatial Comparative Visualization Design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, 2017, pp. 851–860.
4. J. Wang et al., "Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, 2017, pp. 81–90.
5. H. Obermaier et al., "Visual Trends Analysis in Time-Varying Ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 10, 2016, pp. 2331–2342.
6. M. Beham et al., "Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 1693–1702.
7. S. Bruckner and T. Möller, "Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, 2010, pp. 1468–1476.
8. M. Sedlmair et al., "Visual Parameter Space Analysis: A Conceptual Framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 2161–2170.
9. S.S.M. Team, *Sierra/SolidMechanics 4.22 User's Guide*, technical report SAND2011-7597, Sandia National Laboratories, 2011.
10. U. Ayachit et al., "ParaView Catalyst: Enabling In Situ Data Analysis and Visualization," *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization* (ISAV 15), 2015, pp. 25–29.
11. H. Hotelling, "Relations between Two Sets of Variables," *Biometrika*, vol. 28, 1936, pp. 321–377.
12. E. Tufte, *Beautiful Evidence*, Graphics Press, pp. 46–63, 2006.

ABOUT THE AUTHOR

Patricia Crossno is a Principal Member of Technical Staff at Sandia National Laboratories. Contact her at pjcross@sandia.gov

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.

This article originally appeared in IEEE Computer Graphics and Applications, *vol. 38, no. 2, 2018.*