In Situ Solutions with CinemaScience

David H. Rogers, Soumya Dutta, Divya Banesh, Terece L. Turton, Ethan Stam, James Ahrens

Abstract As simulations move to exascale computing, the dominant data analysis and visualization paradigm will shift from primarily post hoc processing to in situ approaches in order to meet I/O bandwidth constraints. One such approach is Cinema, a flexible in situ visualization ecosystem. Cinema combines data extracts with viewers and analysis capabilities to support in situ, post hoc and hybrid approaches for data processing. With data extracts that include metadata, images, meshes, and other data types, Cinema databases generated in situ are a central component of post hoc analysis workflows. These workflows support visualization and exploration of the data, verification and validation tasks, and leverage computer vision and statistical techniques for post hoc analysis. This chapter describes the Cinema approach, the database specification, and demonstrates its use through example workflows.

David H. Rogers
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: dhr@lanl.gov

Soumya Dutta
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: sdutta@lanl.gov

Divya Banesh
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: dbanesh@lanl.gov

Terece L. Turton
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: tlturton@lanl.gov

Ethan Stam
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: stam@lanl.gov

James Ahrens
Los Alamos National Lab, Los Alamos, NM, USA, e-mail: ahrens@lanl.gov

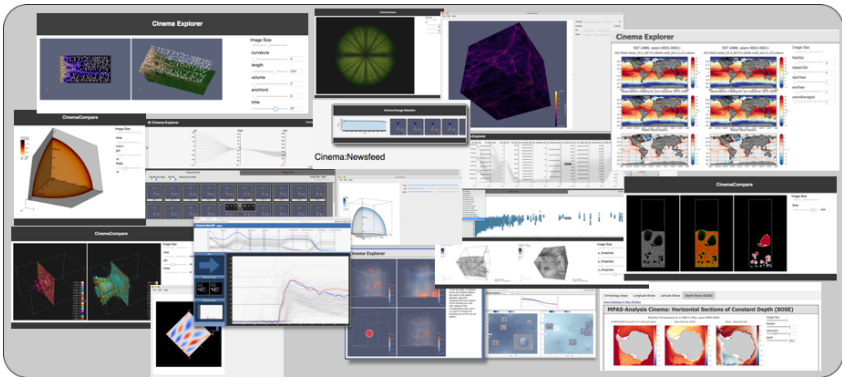

Fig. 1 Cinema is a novel way to capture, store and interact with data extracts from a wide variety of sources. It is well suited for hybrid in situ coupled with post hoc data workflows, and has been integrated into the most common visualization and analysis applications in use at extreme scale.

1 Introduction

Data analysis and visualization workflows are traditionally based on a post-processing model. Data is saved at regular intervals and then visualized post hoc with a standard visualization application. With this model, data sizes – over all and for a single time step – can be quite large. This impacts the amount of data that can be saved, usually requiring significant temporal down-sampling, and limits the ability of the scientist to effectively render and explore the data in a post hoc workflow.

As discussed in the introductory chapter, in situ is a system. Each component of the system contributes in some way to solving the concurrency challenges of large scale computing. The in situ infrastructures in Part II of the Introduction removes the human-in-the-loop, moving analysis algorithms and visualization into the in situ workflow. Data reduction techniques in Part III further downsize the data being saved to disk.

One consequence of moving analysis algorithms and data reduction into the in situ workflow is that the data output may change as a result of an in situ computation. In particular, in situ workflows may result in the generation of small subsets of data – *data extracts*. For example, if an algorithm finds an interesting feature at a specific time and location, the algorithm may create a data extract that allows closer inspection of a relevant portion of the data. That data extract could be a visualization, a subset of the data, a statistical representation of the data, or some other form of data extract.

In contrast to long-standing practice where a simulation outputs a standard data format (which may be specific to that simulation), these data extracts can be widely different from each other in scope (spatial dimensions, variables saved, mesh-based vs image-based, etc.) and time. Thus, output from algorithm-driven in situ pipelines can be a heterogeneous set of data extracts. A key component of the in situ system is the ability to generate these data extracts and therefore complementary post hoc

analysis tools and pipelines are needed that bring the human-in-the-loop back into the system, enabling the domain scientist to generate scientific insights from these reduced data extracts. This is the central problem that the Cinema project addresses.

Cinema provides a novel way of interacting with related sets of data extracts produced in situ and the resultant smaller output size enables the possibility of higher temporal resolution. This approach, first introduced in [2], is a way of capturing, recording, analyzing and interacting with related sets of extracts from scientific data. A Cinema workflow is based on a well-defined database for these extracts, as detailed in the Cinema specification [10]. Cinema databases provide a very compact data representation that can provide many of the benefits of interacting with extremely large data, while also defining a flexible infrastructure for analyzing and interacting with the data. Cinema databases work together with Cinema writers, Cinema viewers and Cinema based analysis algorithms to form the Cinema ecosystem.

The goal of this chapter is to motivate the use of Cinema as part of an in situ workflow and to describe it sufficiently so as to allow the interested reader to understand which aspects of the Cinema ecosystem best fit their needs. This will be accomplished by a survey of Cinema functionality starting with an overview of the Cinema ecosystem, Section 2. The Cinema database is described in Section 2.2 with an overview of alternate data types in Section 2.5. The set of writers available for in situ export of Cinema databases is discussed in Section 2.3. Standard viewers can be found in Section 2.4 along with prototype work.

Analysis capabilities are discussed in Section 3, showcasing examples leveraging computer vision techniques, Section 3.1, and statistical methods, Section 3.2. A task-based workflow, typical for in situ production of Cinema databases, is described in Section 4.

Although Cinema is a general approach to data capture and analysis applicable to both experimental and simulated data, this chapter focuses on the application of Cinema to scientific simulation data, related to the context of this book.

The Cinema ecosystem is constantly evolving and adding new functionality to meet the needs of scientists. Throughout the paper, examples and references to publications demonstrate how scientists are leveraging the Cinema ecosystem to enable scientific insight. A full list of the many publications utilizing Cinema can be found on the CinemaScience website [11].

2 The Cinema Ecosystem

Cinema is an ecosystem of capabilities organized around a database of extracts. The Cinema ecosystem, shown in Figure 2, organizes material from various sources into a database that associates data parameters with data extracts. A Cinema database includes metadata about data written to permanent storage. These databases can be written by any application, operated on by algorithms, and interactively viewed with a set of viewers and viewer components. The simplicity and flexibility of the

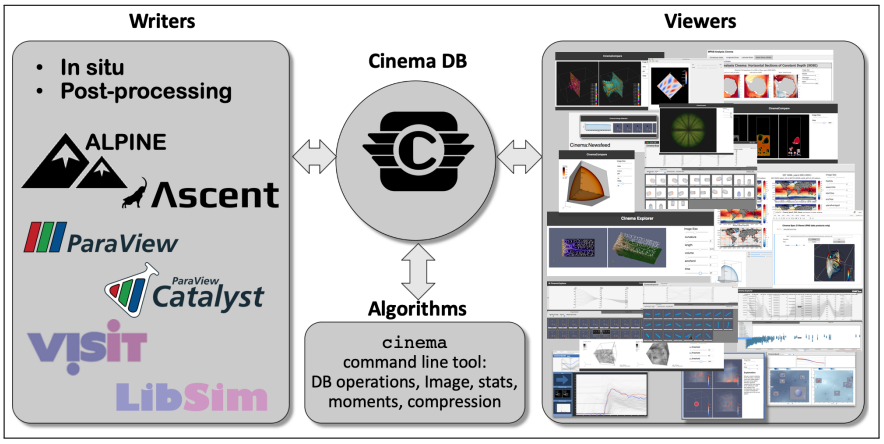

Fig. 2 Cinema is an ecosystem of capabilities organized around a database of extracts. This includes metadata about data written to permanent storage. Most common is a database of images that captures a set of rendered views of the data. These databases can be written by any application, operated on by algorithms, and interactively viewed with a set of viewers and viewer components. The simplicity and flexibility of the database means that it is easy to get started and take advantage of some of the power of Cinema. (Versions of this image have been previously used in internal ECP [14] documents and, for example, in [27] and [8].)

database means that it is quite easy to integrate and experiment with Cinema as an in situ analysis workflow.

Cinema provides the following capabilities:

- A novel **image-based data extract**, the *cinema composable image* [10], which provides interactive data exploration for extreme-scale data. Interactions include: smoothly varying camera positions, so the user can interact with the data, the ability to interactively compose images so that elements can be turned on and off, and interactive post hoc recoloring of data to tune visualizations after the simulation is complete. More details can be found in Section 2.1.
- **Writers** that extract data to a well-specified database format, discussed in Section 2.2.
- **Viewers** that allow interaction with data in a movie-like or interactive application-like manner. This is shown in Section 2.4.
- **Algorithms** that query, analyze and filter sets of results in entirely new ways, discussed in Section 3.

2.1 Simple Use Case: Cinema Image Databases

The application of Cinema as an image database approach is the most simple use case of this ecosystem to illustrate its functionality. Image representations of extreme-size simulations can be rendered as a projection of the simulation to a plane representing

the location of a camera. For example, a user might render images from specific camera positions for all time steps of the simulation. These images provide a compact form by which to interact with the larger data sets. Cinema can then be used to organize, and interact with these sets of images through its viewers. For example, the simulation can be 'rotated' through a series of images captured at different spatial locations. The progression of the simulation over time can be viewed through images captured temporally. Figure 3 shows the results of exporting a Cinema image database from ParaView and viewing the images with a Cinema viewer. Sliders in the viewer allows users to rotate the data and scroll through time interactively.

The *cinema composable image*, an extension of rendered images, allows a viewer to interactively compose elements of an image and recolor them post hoc. Therefore, in addition to manipulating parameters such as camera position, a Cinema viewer can also turn elements on and off, and interactively recolor images. Figure 4 shows how different elements of a cinema composable image can be combined together to provide a more complete picture of a data set while allowing components to be viewed and hidden interactively.

The following sections of the chapter delve more deeply into the database concept, writers, and viewers that comprise the Cinema system.

2.2 The Cinema Database

At a high level, Cinema maps metadata to data extracts saved on disk or other permanent storage. A Cinema database does not encode specific meaning in the metatdata - that is left up to the application. This was a deliberate choice when creating the specification, to retain simplicity at Cinema's core, and allow flexibility for user-written applications. For consistency, tools (viewers, writers) operate on parameters and extracts across the ecosystem.

Cinema's database specification provides simplicity and adaptability. Rows represent database entries, e.g. time steps in a simulation, and columns are the parameters and data extracts available for each entry.

The minimal Cinema database is a directory that contains a required `data.csv` file. Optional data files and directories can include data extracts such as images, small mesh files, other CSV files, text files, or other relevant files. The specification allows for other files to be present in the main directory, so that applications can add additional information that core Cinema tools will ignore. Typically, each row in a Cinema database maps a set of parameters to one or more data extracts. The database specification, [10], contains full details of the current specification for the interested user.

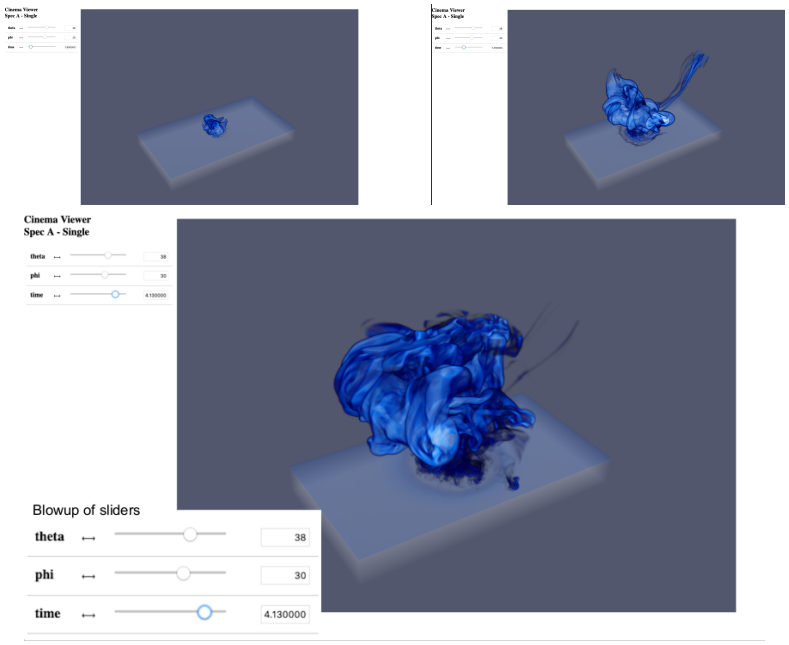

Fig. 3 A Cinema web-based viewer showing a Cinema database of images from the Deep Water Impact Ensemble Data Set [22] to provide interactive exploration of data. In these images, the user operates a time-based slider to scroll through time steps of the simulation. At the top is an image from the beginning and middle of an asteroid impact simulation. At the bottom is an image from late in the simulation. Due to constraints on loading large data sets, this type of interaction would be impossible with full sized data. This interaction is a powerful feature of Cinema, demonstrating that even extreme-sized data can be explored interactively.

2.2.1 Fully Populated vs. Sparse Metadata

One possibility for a Cinema database is that the metadata has a fully populated cross-product of values. This can be achieved by, for example, writing an image from a set of camera positions for each time step. Table 1 is an example of a fully populated metadata database. A data set such as shown in Figure 3 is an example of a fully populated cross-product Cinema database that can be read by a Cinema viewer designed to view a fully populated metadata database, such as Cinema:View.

A second option is that the columns are not full cross-products. This is a frequent occurrence in scientific data sets, where all values for all parameters may not be defined, and all combinations of the variables may not be defined. Consider this example, in which the variable `isovar` is defined on `time step 1`, but not `time step 0`. In this case, the Cinema database would look like the one in Table 2. The Cinema ecosystem provides a set of standard viewers (Section 2.4) to handle either case – sparse or full cross-product databases.

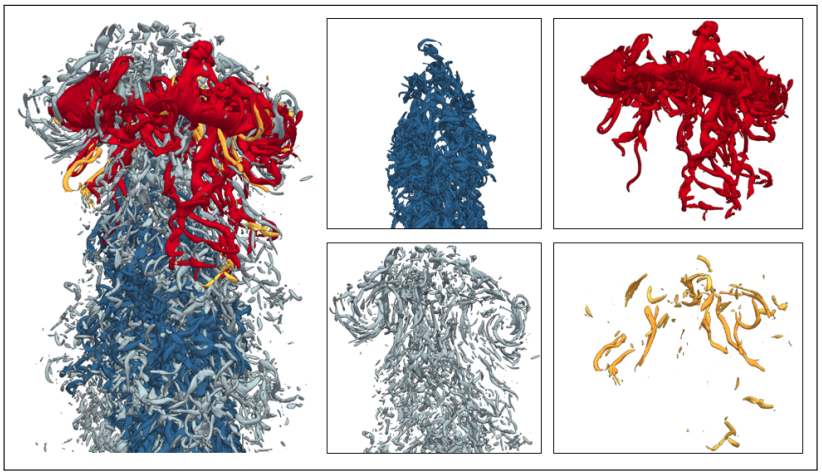

Fig. 4 These images represent elements of a cinema composable image, demonstrating how these can be composited by a Cinema viewer. The renderings show four clusters of roughly 5,000 highly turbulent regions of a computational fluid dynamics simulation. On the left is a composite image showing all four regions together. On the right are the four separate regions, each stored in a different component of a cinema composable image. Each of the clusters can be viewed independently, but also, through the cinema composable image specification, they can be combined together (composited) to show a complete picture of the data. These components can be interactively composited, producing the effect of turning elements 'on' and 'off' in any view of the data. This makes the resulting Cinema database highly interactive and explorable (Images courtesy of J. Lukasczyk, Arizona State University).

time	phi	theta	FILE
0	0	45	000.png
0	0	90	001.png
0	45	45	002.png
0	45	90	003.png
1	0	45	004.png
1	0	90	005.png
1	45	45	006.png
1	45	90	007.png

Table 1 This example `data.csv` file is for a Cinema database containing images for a simple phi/theta camera move over two time steps. Time varies over `[0,1]`, phi over `[0,45]` and theta over `[45,90]`, and the PNG images each have a unique filename.

2.3 Cinema Writers

As part of an in situ workflow, Cinema image database export is available in common open source scientific visualization applications and infrastructures. ParaView [1] provides post-processing Cinema export and in situ export is available through the ParaView Catalyst [4, 15] in situ library. VisIt [9] also provides post-processing ex-

time	isovar	isovalue	FILE
0			000.png
1			001.png
2	temperature	100.0	002.png
3	temperature	150.0	003.png

Table 2 This is a sparse database, in which data values (columns) may be missing, valued as `Nan`, or duplicates of other values. This is commonly seen in tables of values from, for example, experimental scientific data sets. Cinema viewers are still able to view data sets like this, through UI widgets other than continuous sliders.

port capability. Ascent [17], a new flyweight infrastructure under development as part of the Exascale Computing Project [14], also contains Cinema export functionality. As an example, the MPAS-Ocean [24] simulation was instrumented with Catalyst's in situ capability to export a Cinema database [20]. ParaView was used post-processing to generate Cinema databases from the Nyx [3] cosmology simulation, used in [8] and as seen in Figure 5 and Figure 7.

Common to each of these export capabilities is the ability to choose a static view, i.e., a single camera angle, or a `phi-theta` view with a user-specified number of steps in (ϕ, θ). By default, the ϕ and θ steps are regularly spaced.

It is useful to note that Cinema databases can also be built post hoc. Simple Python or bash scripts can be used to organize already existing images or output plots into Cinema databases. While this chapter focuses on the in situ use of Cinema, Cinema databases built post hoc have been used for both simulation and experimental data sets. The Foresight framework [16] has used Cinema databases to allow scientists to interactively explore the impact of compression on simulation data. Cinema has also been used for a variety of experimental analysis workflows that output images. These include shock physics experiments [21], experimental diffraction images [29], and Bragg peak detection and tracking as discussed in [27].

Lastly, we note that most while most of the in situ writers work in a tightly coupled mode, Cinema can also be implemented in a loosely coupled or in transit environment such as described in various chapters in this book.

2.4 Cinema Viewers

A core concept in Cinema is the *viewer* — flexible applications that can read in any specification-compliant Cinema database and enable analysis workflows. Cinema includes three standard viewers, Cinema:Scope, Cinema:View and Cinema:Explorer that will meet the needs of most users. There is also a library of individual viewer components that can be used to build workflow-specific viewers. This section overviews each of the standard viewers, providing example use cases. The Cinema ecosystem constantly evolves to meet user needs and prototype viewer development is also discussed.

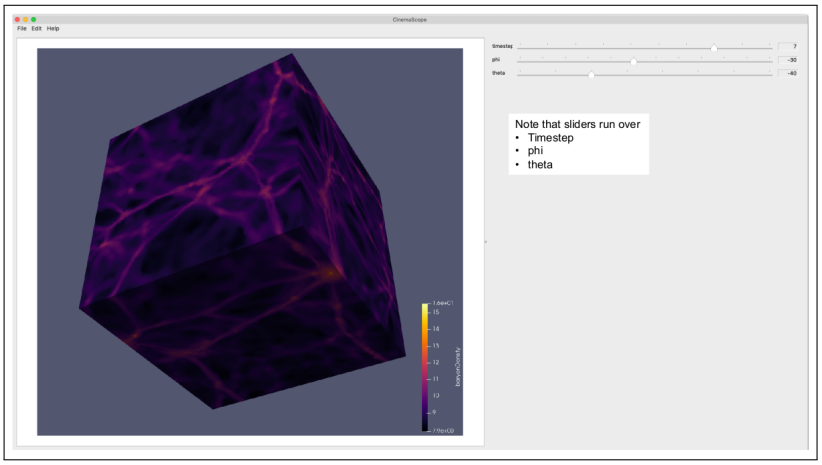

Fig. 5 CinemaScope is used to view a Nyx cosmology simulation showing the formation of dark matter halos over time. The three sliders, timestep, phi, theta, are mapped to the mouse controls to enable intuitive movement through the Cinema image database.

2.4.1 Cinema:Scope

Cinema:Scope is a cross-platform application built on Qt and C++. It has slider controls mapped to the database parameters. By default, Cinema:Scope loads the first set of images in a Cinema database and provides intuitive mouse controls that are mapped to `phi` and `theta`. The mouse control mapping and image set can be changed within the application. Figure 5 shows an example of a cosmology simulation [3] viewed within Cinema:Scope. The database parameters are `time`, `phi`, and `theta` where `phi` and `theta` are mapped to the mouse controls.

This functionality is one of the attractive features of Cinema:Scope. It gives the user the feel of using a full visualization application such as ParaView or VisIt but without the overhead of rendering each image. For simulations where the domain scientist already knows the visualizations needed, that rendering can be done as part of the in situ workflow. Cinema:Scope provides the post hoc interactive exploratory functionality needed by the scientist while avoiding the computationally expensive part in the post hoc workflow.

2.4.2 Cinema:View

Cinema:View is a basic browser-based viewer used to visually explore images in a Cinema database. Cinema:View is based on JavaScript and D3. It features slider controls and can be used to view a single Cinema image database or multiple databases with common parameter sets. Figure 6 uses Cinema:View to view three ways to detect voids or bubbles in an MFiX-Exa [26] bubbling fluid bed simulation.

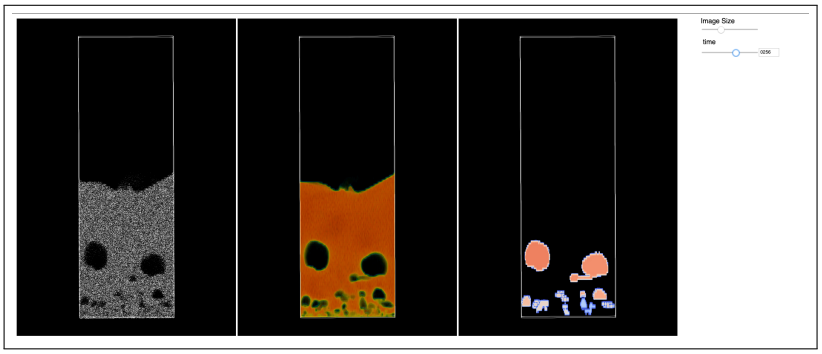

Fig. 6 Cinema:View is used to view the detection of voids (bubbles) in an MFiX-Exa bubbling fluid bed simulation. From left to right, a subset of the original data, downsampled to 5% of the data (see the chapter on *Sampling In situ*); a density field is used to find bubbles; a bubble finding algorithm using the density field calculation shows only the voids. The three views of the bubbling bed simulation can be compared over time using the slider. Image size can also be changed to fit the user browser. MFiX-Exa data courtesy of A. Almgren and J. Blaschke, Lawrence Berkeley National Laboratory; bubble images courtesy of A. Biswas, Los Alamos National Laboratory.

2.4.3 Cinema:Explorer

Also browser-based using JavaScript and D3, Cinema:Explorer leverages parallel coordinates to explore data within a Cinema database. The use of parallel coordinates for Cinema databases was first explored in [31]. Parallel coordinates are a common approach to exploring high dimensional data.

Cinema:Explorer includes a parallel coordinates view, an image spread view, and a scatterplot view. The view panel information is linked so that a selection in the parallel coordinates panel brings up the associated images in the image spread view and the scatterplot. This can be used, for example, to identify outliers or explore correlations in large data sets.

The screenshot in Figure 7 shows Cinema:Explorer being used to query a large image database using the parallel coordinates interactive view. For this example, a Nyx cosmology Cinema database has had computer vision algorithms applied to generate image-based statistics. Cinema:Explorer displays the database parameters and statistical quantities in parallel coordinates. Standard parallel coordinate techniques can be used to hide/show axes or select ranges on the axes. In this example, the user has selected images early in time and with low entropy. Cinema:Explorer shows all images in the database that match this query. Cinema:Explorer allows scientists to quickly view queries on data ranges across all samples extracted, making it possible to explore and compare large sets of data very quickly.

This is a typical Cinema enabled workflow: saving parameters and data extracts such as pre-rendered visualizations in situ, extending the analysis post hoc, and using the viewer to select, query, and explore the information within the database.

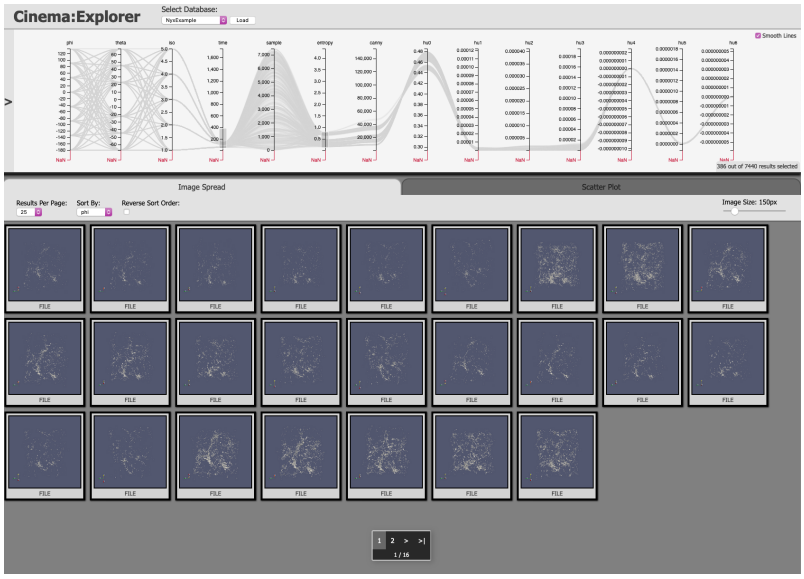

Fig. 7 In this image, Cinema:Explorer is used to query a large image database, using the parallel coordinates interactive view. For this example, the Nyx cosmology Cinema database has had computer vision algorithms applied to generate image-based statistics. Cinema:Explorer displays the database parameters and statistical quantities in the parallel coordinates view. Standard parallel coordinate techniques can be used to hide/show axes or to select ranges on the axes. Using the axes, the user has selected images early in time and with low entropy. The results of the query can be seen in the image spread view.

2.4.4 Jupyter-Based Viewers

Many simulation scientists use the Python scientific analysis stack and notebooks are becoming a more common approach to the analysis workflow for simulations. NERSC has a dedicated JupyterHub to connect notebooks to HPC resources. To accommodate Python users, a prototype Jupyter notebook-based viewer is available at `https://github.org/cinemascience/cinema_jnc`. Figure 8 shows a WarpX [28] simulation of a plasma-driven accelerator in the Cinema:JNC viewer. Currently, this has similar functionality to Cinema:View, allowing the user to use sliders to view a Cinema database through time and spatial angles. This viewer could be included in a Jupyter-based workflow, leveraging Python's data analysis capabilities for post hoc analysis. Including multiple data types in a Cinema database, as described in Section 2.5, allows workflows that, for example, use Python/VTK [25] based pipelines within a Jupyter notebook-based analysis framework.

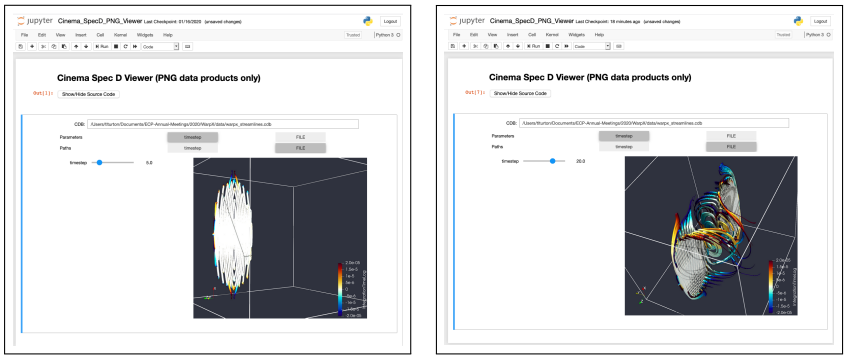

Fig. 8 Two views showing field streamlines in a WarpX plasma accelerator simulation. The prototype Jupyter notebook based viewer, Cinema:JNC, is used to explore the WarpX Cinema database. An early time step is on the left, evolving to a later time step on the right. Cinema database courtesy of R. Bujack, Los Alamos National Laboratory.

2.4.5 Cinema Components

In addition to the standard viewers, Cinema also has a library of components that can be combined to create a viewer specific to the needs of the scientist.

These components include a parallel coordinates plot, image spread, scatter plot, query generator, and glyph-style plots. An example use case of an analysis-specific viewer built with individual components (beyond those used in Cinema:Explorer) can be found in [29]. In addition to these components, users can develop and combine new components as needed. Cinema:Bandit [21] is one such instance of a viewer built for a specific scientific application.

2.5 Data Types Beyond Images

Cinema works on any data type and supports multiple data extracts and mixed data types for each parameter set (row in the database). For example, an in situ analysis could identify a specific feature of interest. The Cinema database export might save both a visualization of that feature and a small VTK-based mesh containing that feature. This is an effective way to downsample the data by not saving the full simulation mesh. That VTK-based file is then included in the Cinema database, associated with the same set of parameters (e.g., view angles and time) that identify the corresponding image.

An example of multiple data types can be seen using the Cinema:Explorer viewer. In Figure 9, a Cinema database of a simple sphere contains the visualization of the sphere for different ϕ and θ values. Some of the rows also have an associated VTK or PBD file. Clicking on file types brings up the vti or pbd data for interactive viewing

Fig. 9 A simple sphere Cinema database demonstrates the multiple data types in Cinema:Explorer. For each [ϕ-θ] set of parameters, there is an image. Additionally, some of the [ϕ-θ] parameter sets have other data types such as a vtk file or pbd file. Clicking on the thumbnail for each extract (vtk or pbd file) will bring up the data in a viewer for that specific data type, Figure 10.

Fig. 10 Different file types: ParaView pbd file on the left and a ParaView vti file on the right. These are displayed in a modal view when selected by clicking on the file name in the Cinema:Explorer image spread.

in a modal view, Figure 10. Rows where the vti or pbd datafile is not available have an informational message displayed.

3 Analysis Algorithms

As discussed in this chapter's introduction, Cinema is a hybrid approach that produces heterogeneous data extracts in situ. These extracts can become the input for post hoc analysis workflows. The light-weight nature of the Cinema database approach enables flexible real-time exploration. This exploration may be conducted through the Cinema:View viewer, the parallel coordinates plot in Cinema:Explorer and in the case of image databases with associated saved parameters, through computer vision

and statistical algorithms applied to the images and saved data parameters. Large image databases with several columns of associated numerical information can easily be produced through the methods discussed in Section 2.3. Standard tools to help users explore and understand these data products are an essential facet of the Cinema ecosystem. Two examples of algorithms for Cinema image database exploration – a computer vision framework and a set of statistical methods – illustrate how users may employ such techniques for in-depth analysis.

As shown in Figure 2, the image-based *Analysis Algorithms* are part of an *iterative workflow*. The results of analysis algorithms, either computer vision or the statistical methods, are collated back into the Cinema database as additional images or columns of numerical data. Therefore, a user who needs to compute a series of steps on their data can accomplish this goal through a sequential set of commands.

It is important to note that when analyzing images, the format in which the image is saved becomes highly pivotal. For RGB color-mapped images, the user must be aware of the potential effects of the colormap on their data and subsequent post-processing analyses. Alternatively, the user may opt to save their data as *cinema composable images*, where the simulation data is directly projected into the 32-bit pixels of the image. This allows for a more direct application of the Analysis Algorithms to the underlying data.

3.1 Computer Vision Framework

There exists a wide body of image-based computer vision techniques that can be exploited in a Cinema database workflow. However, a computer vision framework for data analysis and visualization must ensure that each component of the framework helps the user identify and and examine features that directly correspond to attributes of the simulation. Therefore, the methods currently included in this framework help to identify common physical properties of interest in scientific data. For example, a gradient-based edge-detection algorithm is included to identify regions with sharp discontinuities. An examination of the Western Boundary of the Gulf Stream discussed in [27] is an example of an application of this edge-detection technique.

Another capability of the computer vision framework is contour detection, to locate closed regions at and above (or below) a given threshold, i.e., superlevel or sublevel sets. These extracted features may then be matched temporally or spatially based on attributes and metrics such as location, pixel intensities, area or derived quantities such as Hu moments. Given a particular matching metric, these matched features can be tracked temporally to identify events such as splits, merges, deaths or births. An example of this computer vision based workflow is discussed below and the interested reader can find further details in [6] and [27].

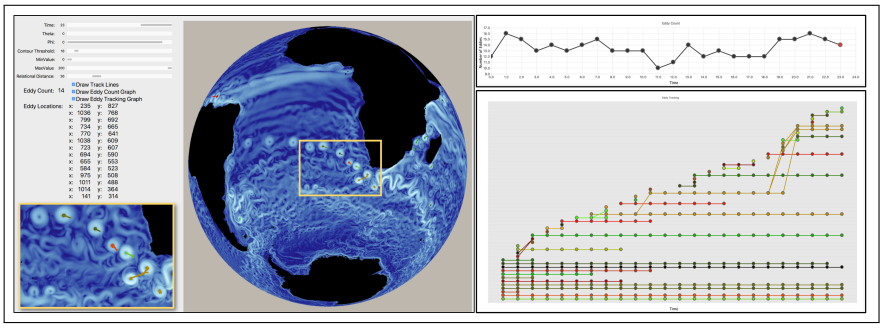

Fig. 11 Visualization of eddy feature tracking in the Agulhas Retroflection Region of an MPAS-Ocean Simulation. The computer vision framework includes a control panel (upper left, see Figure 12 for more detail) with sliders to select images and set algorithm parameters; visualization panels for overview and zoomed display; and output information such as the eddy count (upper right) and tracking chart (lower right). (Image is adapted from our previous work [6].)

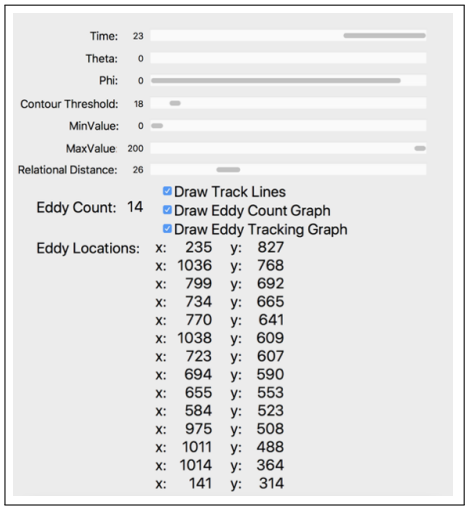

Fig. 12 Close-up of the computer vision framework control panel. At the top are sliders for the database parameters: time, theta, and phi. Underneath are algorithm parameters such as thresholds and ranges. User interface options allow the user to customize the views. Brief overview of the algorithm output is also included. (Image is adapted from our previous work [6].)

3.1.1 Case Study: MPAS-Ocean Eddy Tracking

A Cinema enabled computer vision workflow for eddy analysis is described in [6]. This is an example of a typical analysis workflow leveraging the Cinema ecosystem. This system allows the user to identify mesoscale ocean eddies, track their movement and visualize these results over time through count and tracking graphs. The input to the application is a Cinema database of MPAS-Ocean floating point images, to

which the feature analysis algorithms are applied. The Cinema database images have the relevant physical variables, in this example, kinetic energy, embedded within the images allowing the user access to the simulation data at the resolution of the saved images.

An interface such as the one shown allows certain areas of domain expertise to be inherently captured. What is considered an "eddy" is intuitively understood by the scientist even though a rigorous mathematical definition of an eddy eludes ocean scientists. The eddy tracking application, shown in Figure 11, allows the user to select contour detection as the computer vision technique and flexibly set thresholds to identify a set of eddies spatially. The application then tracks and creates a timeline of eddy progression. To track big eddies or small eddies, to include weaker eddies or only allow stronger eddies, and to determine over which region the eddy analysis will occur are all actions enabled by the interface. The user can choose to output a running count of the number of eddies found and a tracking graph that indicates the death/birth/merge events in eddy formation (upper and lower right of Figure 11, respectively). This combination of low-cost Cinema database images and optimized computer vision algorithms enables an interface for scientists where exploration of the data is possible in real-time. The real-time Cinema based approach can be compared to mesh-based analysis methods such as geometric eddy detection algorithms. Running those types of algorithms on large data sets requires far more time, limiting the exploratory capabilities for the user.

3.2 Statistical Methods

Statistical tools in the Cinema ecosystem are another data analysis approach within a Cinema workflow. Statistical tools are a ubiquitous choice for analysis of numerical data. The data may be derived from a Cinema image database, obtained during in situ processing, extracted from simulation data or added to the Cinema database from external sources. `cinema` [12] is a command line python-based tool that allows users to extract typical image properties such as mean, standard deviation, Shannon entropy and joint entropy into numerical quantities added to the Cinema database. Maack et al.[18] leveraged the Cinema framework to apply statistical metrics to find features of interest.

Another approach enabled by Cinema is statistical change point detection (see for example, [23] for a discussion of the concept of change point detection). Within a physical system, *change* often denotes an interesting time step or event. Visually scanning thousands of images may be too time-intensive for a domain scientist to find events of interest. Change point detection can be applied to properties of simulation images to identify time steps or parametric values of interest. Both [7] and [5] are examples of the application of change point detection within the body of Cinema literature. The change point detection algorithm available as part of the Cinema ecosystem, [13], allows the user to identify locations in a sequence of numbers where *change* has occurred [19], as defined through a linear regression model.

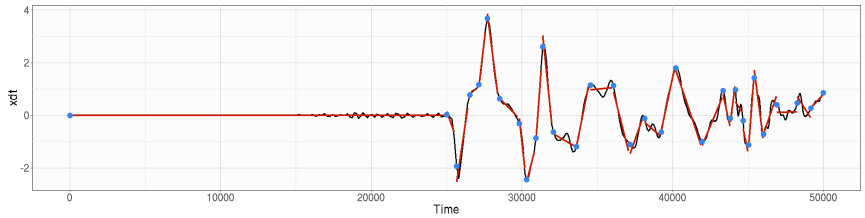

Fig. 13 Statistical change point detection for the 'xdt' parameter of the Deep Water Impact Ensemble Data Set [22]. Change detection parametric values are: $B = 4$, $\alpha = 0.8$, $\delta^2 = 1$. The change points are noted in blue while the red lines drawn show the best fit line to the data between change points.

In Figure 13, the Cinema statistical change point detection algorithm is applied to a simulation of the Deep Water Impact Ensemble Data Set [22], the same simulation shown in Figure 3. Instead of using images stored in a Cinema database, the parameters in the database can be used for the statistical change point detection analysis. In this analysis, a representative slice is taken in the z-plane through the center of the data and the data values on that slice are averaged. This process is repeated for each of the 476 saved time steps spanning about 50,000 steps in simulation time. Change detection is then applied to a parameter of interest from the simulation parameters. The selected parameter, *xdt*, is the x component of the velocity in each cell, in centimeters per second. Change point detection is applied to this parameter over time to identify points of interest.

During the first 25000 time units of the simulation, as the asteroid is descending towards earth, there is very little activity and therefore, no change points detected. Intense and frequent change points are seen as the asteroid strikes the ocean surface. The algorithm identifies points in time where moderate to large amounts of change occur. Even in the latter part of the simulation, regions of time with smaller amount of change are still grouped together. The algorithm enables the flexibility to adjust the change detection parameters so as to find smaller or larger amounts of change, as desired by the user for their particular analysis and data set.

4 Task-Based Workflow Examples

To streamline in situ Cinema database production, a task-based workflow has been developed that works in a distributed parallel environment. In this workflow, the user first specifies the parameters that will be used to generate the Cinema database and the range of values for each parameter. The user also can select if the parameter ranges will be divided regularly or randomly while creating specific parameter combinations. For example, the user can specify the ranges of the viewing angle parameters (e.g., $240 \leq \phi \leq 360$ *and* $50 \leq \theta \leq 100$) and how many visualization samples are needed from this range. Then the framework will sample the parameter

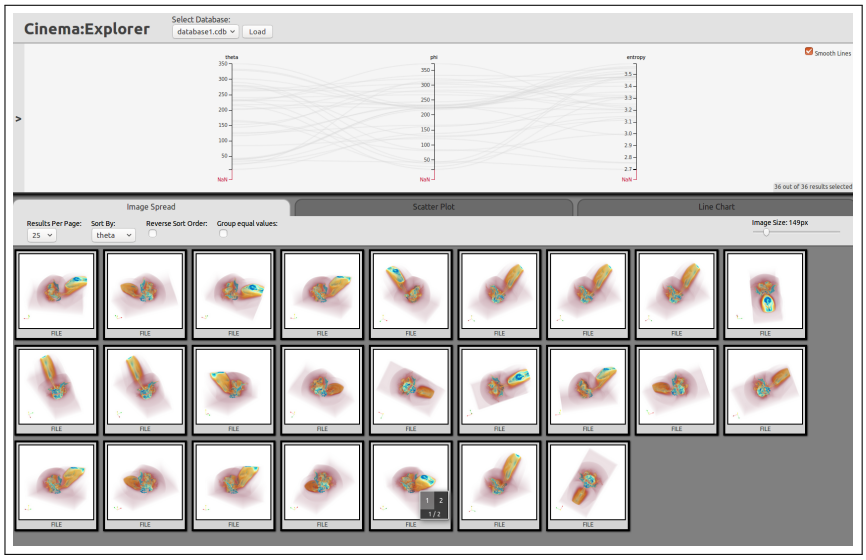

Fig. 14 Visualization of a Cinema database produced by our task-based workflow using Asteroid impact data set [22]. The database produced visualization of Temperature field using volume visualization technique.

ranges either regularly or randomly as specified by the user to produce combinations of (ϕ, θ) values. For each such (ϕ, θ) combination, a visualization extract will be added to the Cinema database. The user can also specify the type of visualization algorithm such as surface rendering or volume rendering that will be used to produce the visualization data extracts.

In this workflow, a *unit task* is characterized as the production of a visualization extract for a specific parameter combination (phi, $theta$, etc). Since the visualization and rendering in parameter space is high-dimensional in nature, and can be quite large, a through exploration of such space while producing a Cinema database may need a very large number of such unit tasks. Therefore, the task-based workflow first generates the list of tasks that will be needed in order to produce the complete Cinema database as specified by the user. Those tasks are then distributed among the different compute nodes in a high performance cluster. Finally, the tasks are executed in parallel. At the end of the task-based workflow, a Cinema database is created with all the visualization extracts. Finally, the workflow installs a Cinema database viewer customized for the resultant Cinema database so that the results can be readily explored interactively. Figure 14 shows an example Cinema database generated using this task-based workflow and viewed using the Cinema:Explorer viewer, Section 2.4.

The current version of the task-based workflow is implemented in Python using the `mpi4py` library for distributed processing. The visualizations are produced using ParaView [1] in off-screen rendering mode. User control is through a `JSON` file specifying the range of parameters and types of visualizations needed. The task-

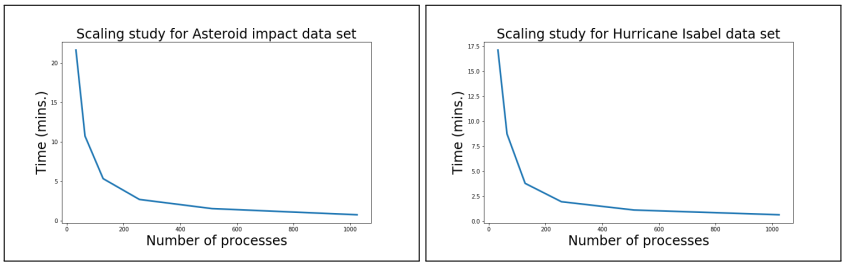

Fig. 15 Strong scaling study of Cinema's task-based workflow for (left) the asteroid impact data set and (right) the Hurricane Isabel data set.

based workflow will then generate the list of tasks and execute them in parallel on the HPC machine.

Performance testing of the task-based workflow was done on the high performance cluster (HPC) Snow, an Institutional Computing (IC) Commodity Technology System Phase I cluster (CTS-1) located at Los Alamos National Laboratory. Snow has two integrated Scalable Units (SU) and each unit forms a building-block to assemble the CTS-1 cluster. Each SU has 184 compute nodes plus other nodes for services, I/O, etc. Each node in the SU has 36 Processor cores: 2 x (E5 2695v4 2.1GHz, 18 cores, 45MB cache), 128GB Memory, and Intel OmniPath OP HFI, Single-port, PCIe-gen3 x16 Network Interconnect.

In Figure 15, the result of a strong scaling study of the task-based workflow is shown for two sets of data: the asteroid impact data [22] and Hurricane Isabel data [30]. The asteroid data used volume rendering for visualization and the Hurricane Isabel data used an isocontour visualization technique. Each test case consisted of 10000 tasks and the number of processing cores were varied from 32 to 1024. Since the tasks were distributed among different processing nodes and the tasks are independent, it can be observed that with increased number of processing cores, the computation time goes down and the workflow scales as expected for both the data sets. This study demonstrates the practicality of a task-based workflow for in situ generation of image-based Cinema databases.

5 Conclusion

Cinema provides a powerful ecosystem for extracting, storing and interacting with scientific and experimental data at any scale – from small tables of data to extreme-scale simulations running on the largest supercomputers. As an open data definition, a Cinema database is a data set that any application can read and write, making it simple to start using Cinema, and adding capabilities as needed. In contrast to monolithic applications, Cinema provides an opportunity to participate by re-using small components from open source libraries. Cinema enabled data analysis and visualization workflows have been used across a wide range of data. Already included

in many standard applications and frameworks for large scale science, Cinema is a good option for experimentation or production systems for complex scientific data analysis.

Acknowledgements

Many collaborators have contributed to the development of Cinema. We thank everyone who contributed and would like to acknowledge our LANL Data Science at Scale team members: A. Biswas, C. Biwer, R. Bujack, L-T. Lo, D. Orban, J. Patchett, C. Tauxe, J.Q. Wofford; our science collaborators: G. Gisler, M. Petersen, J. Schoonover, Z. Lukić, J-L. Vay; our industry partners at Kitware and Intelligent Light and other collaborators: J. Lukasczyk, C. Harrison, M. Larsen, J. Woodring, G. Aldrich, G. Streletz. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. The Hurricane Isabel data was provided by Wei Wang, Cindy Bruyere, Bill Kuo, and others at NCAR with Tim Scheitlin at NCAR converting the data into the Brick-of-Float format. This research was released under LA-UR-20-20649.

References

1. Ahrens, J., Geveci, B., Law, C.: Paraview: An end-user tool for large data visualization. The Visualization Handbook **717** (2005)
2. Ahrens, J., Jourdain, S., O'Leary, P., Patchett, J., Rogers, D.H., Petersen, M.: An image-based approach to extreme scale in situ visualization and analysis. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 424–434. IEEE Press (2014)
3. Almgren, A.S., Bell, J.B., Lijewski, M.J., Lukić, Z., Andel, E.V.: Nyx: A massively parallel AMR code for computational cosmology. The Astrophysical Journal **765**(1), 39 (2013). DOI 10.1088/0004-637x/765/1/39
4. Ayachit, U., Bauer, A., Geveci, B., O'Leary, P., Moreland, K., Fabian, N., Mauldin, J.: Paraview catalyst: Enabling in situ data analysis and visualization. In: Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, pp. 25–29. ACM (2015)

5. Banesh, D., Petersen, M., Wendelberger, J., Ahrens, J., Hamann, B.: Comparison of piecewise linear change point detection with traditional analytical methods for ocean and climate data. Environmental Earth Sciences **78**(21), 623 (2019)
6. Banesh, D., Schoonover, J.A., Ahrens, J.P., Hamann, B.: Extracting, Visualizing and Tracking Mesoscale Ocean Eddies in Two-dimensional Image Sequences Using Contours and Moments. In: K. Rink, A. Middel, D. Zeckzer, R. Bujack (eds.) Workshop on Visualisation in Environmental Sciences (EnvirVis). The Eurographics Association (2017). DOI 10.2312/envirvis.20171103
7. Banesh, D., Wendelberger, J., Petersen, M., Ahrens, J., Hamann, B.: Change point detection for ocean eddy analysis. In: Workshop on Visualisation in Environmental Sciences (EnvirVis). The Eurographics Association (2018). DOI 10.2312/envirvis.20181134
8. Bujack, R., Rogers, D., Ahrens, J.: Reducing occlusion in cinema databases through feature-centric visualizations. In: Leipzig Symposium on Visualization In Applications (LEVIA) (2018). URL https://datascience.dsscale.org/wp-content/uploads/2019/01/ReducingOcclusioninCinemaDatabasesthroughFeature-CentricVisualizations.pdf
9. Childs, H., et al.: VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In: High Performance Visualization—Enabling Extreme-Scale Scientific Insight, pp. 357–372. CRC Press/Francis–Taylor Group (2012)
10. Cinema: Cinema specification. https://github.com/cinemascience/cinema/blob/master/specs/dietrich/01/cinema_specD_v012.pdf (2018). (Accessed January 2020.)
11. Cinema-Developers: Cinema publications. https://cinemascience.github.io/publications.html (2018). (List of Cinema publications, available on CinemaScience website.)
12. Cinema-Developers: Cinema science. http://cinemascience.org/ (2018). (Accessed on 12/11/2019)
13. Cinema-Developers: Cinema change point detection. https://github.com/cinemascience/cinema_change_detection (2019). (Accessed on 1/22/2020)
14. ECP: Exascale Computing Project. https://www.exascaleproject.org/ (2017). (Accessed January 2020.)
15. Fabian, N., Moreland, K., Thompson, D., Bauer, A.C., Marion, P., Gevecik, B., Rasquin, M., Jansen, K.E.: The ParaView coprocessing library: A scalable, general purpose in situ visualization library. In: Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on, pp. 89–96. IEEE (2011)
16. Grosset, P., Biwer, C.M., Pulido, J., Mohan, A.T., Biswas, A., Patchett, J., Turton, T.L., Rogers, D.H., Livescu, D., Ahrens, J.: Foresight: Analysis that matters for data reduction (2020). To appear in SC '20: International Conference on High Performance Computing, Data, and Analytics., November 2020
17. Larsen, M., Ahrens, J., Ayachit, U., Brugger, E., Childs, H., Geveci, B., Harrison, C.: The alpine in situ infrastructure: Ascending from the ashes of strawman. In: Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization, ISAV'17, pp. 42–46. ACM, New York, NY, USA (2017). DOI 10.1145/3144769.3144778
18. Maack, R., Rogers, D., Gillmann, C.: Exploring cinema databases using multi-dimensional image measures. In: Leipzig Symposium on Visualization In Applications (LEVIA) (2019)
19. Myers, K., Lawrence, E., Fugate, M., Bowen, C.M., Ticknor, L., Woodring, J., Wendelberger, J., Ahrens, J.: Partitioning a large simulation as it runs. Technometrics **58**(3), 329–340 (2016)
20. O'Leary, P., Ahrens, J., Jourdain, S., Wittenburg, S., Rogers, D.H., Petersen, M.: Cinema image-based in situ analysis and visualization of mpas-ocean simulations. Parallel Comput. **55**(C), 43–48 (2016). DOI 10.1016/j.parco.2015.10.005. URL https://doi.org/10.1016/j.parco.2015.10.005
21. Orban, D., Banesh, D., Banesh, C., Biwer, C., Biswas, A., Saavedra, R., Sweeney, C., Sandberg, R., Bolme, C.A., Ahrens, J., Rogers, D.: Cinema:bandit: a visualization application for beamline science demonstrated on xfel shock physics experiments. Journal of Synchrotron Radiation **27**(1) (2020). DOI 10.1107/S1600577519014322

22. Patchett, J.M., Gisler, G.R.: Deep Water Impact Ensemble Data Set. Tech. rep., Los Alamos National Laboratory (2017). LA-UR-17-21595
23. Ray, B.K., Tsay, R.S.: Bayesian methods for change-point detection in long-range dependent processes. Journal of Time Series Analysis **23**(6), 687–705 (2002)
24. Ringler, T., Petersen, M., Higdon, R.L., Jacobsen, D., Jones, P.W., Maltrud, M.: A multi-resolution approach to global ocean modeling. Ocean Modelling **69**, 211 – 232 (2013). DOI http://dx.doi.org/10.1016/j.ocemod.2013.04.010
25. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit (4th ed.). Kitware, Clifton, New York (2006)
26. Syamlal, M., Musser, J., Almgren, A., Bell, J., Hrenya, C., Hauser, T., Liu, P.: MFIX-Exa: A CFD-DEM code for exascale computer architectures. Abstract and Presentation in Computational Modeling and Validation for Fluidization Processes at AIChE Annual Meeting (2018)
27. Turton, T.L., Banesh, D., Overmyer, T., Sims, B.H., Rogers, D.H.: Enabling domain expertise in scientific visualization with cinemascience. IEEE Computer Graphics and Applications **40**(1), 90–98 (2020). DOI 10.1109/MCG.2019.2954171
28. Vay, J.L., et al.: Warp-X: a new exascale computing platform for beam-plasma simulations. ArXiv e-prints (2018). ArXiv:1801.02568 [physics.acc-ph]
29. Vogel, S.C., Biwer, C.M., Rogers, D.H., Ahrens, J.P., Hackenberg, R.E., Onken, D., Zhang, J.: Interactive visualization of multi-dataset rietveld analyses using cinema:debye-scherrer. J. Appl. Crystallogr. **51** (2018). DOI 10.1107/S1600576718003989
30. Wang, W., Bruyere, C., Kuo, B., Scheitlin, T.: IEEE visualization 2004 contest data set. http://sciviscontest.ieeevis.org/2004/data.html (2004). NCAR
31. Woodring, J., Ahrens, J.P., Patchett, J., Tauxe, C., Rogers, D.H.: High-dimensional scientific data exploration via Cinema. In: 2017 IEEE Workshop on Data Systems for Interactive Analysis (DSIA), pp. 1–5 (2017). DOI 10.1109/DSIA.2017.8339086