

# The Cinema Project

Terry Turton

ECP Community BOFs



Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

# The Cinema Team

David Rogers (PI), James Ahrens, Soumya Dutta, Divya Banesh, Ethan Stam, Roxana Bujack, Ollie Lo, Terry Turton

Data Science at Scale Team, Information Sciences, Los Alamos National Laboratory

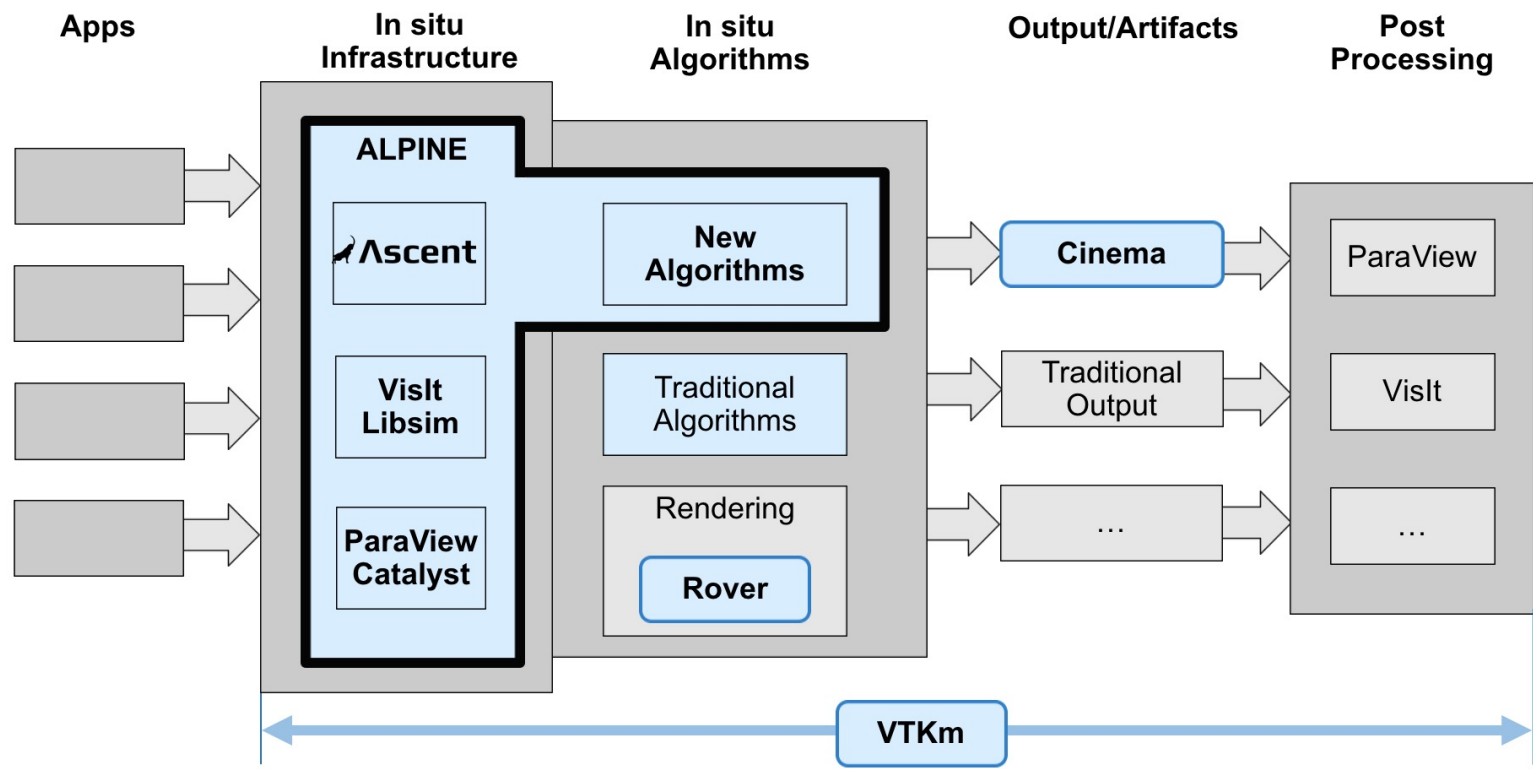
*And with thanks to the many users, students, and collaborators who have expanded Cinema functionality and capabilities over the years.*

## Acknowledgement

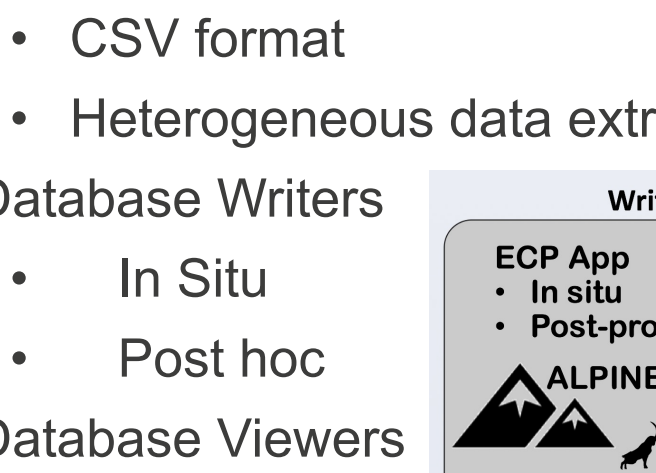
This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

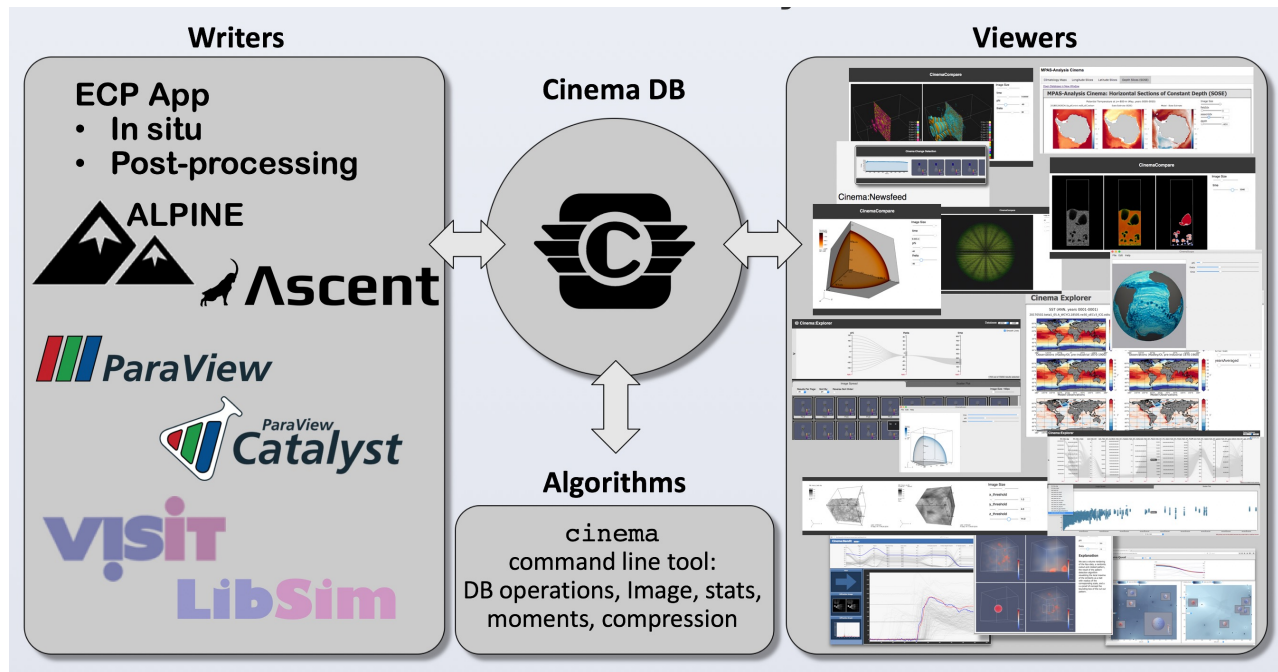


# Cinema within the ECP In Situ Workflow



# The Cinema Ecosystem

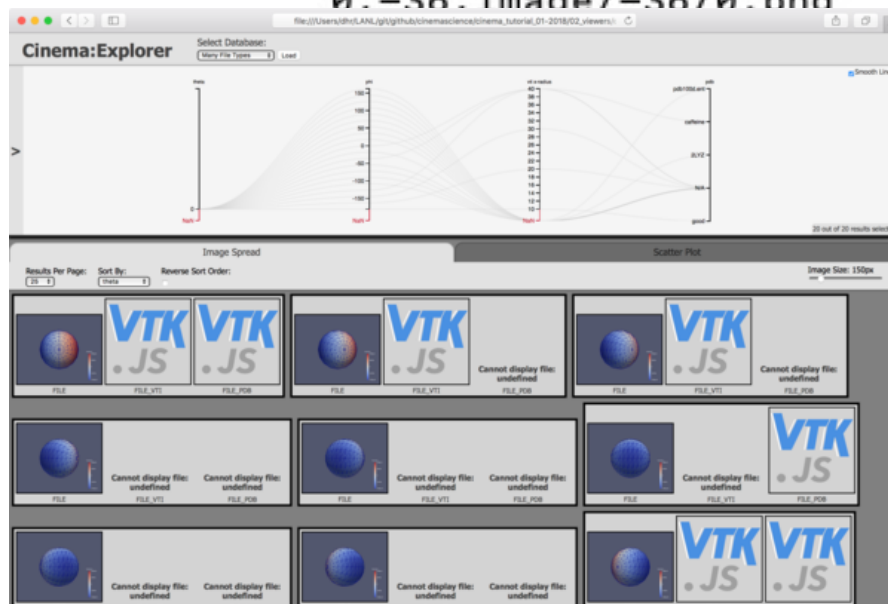
- Database Specifications
    - CSV format
    - Heterogeneous data extracts
  - Database Writers
    - In Situ
    - Post hoc
  - Database Viewers
  - Algorithms
- 
- The diagram illustrates a data processing pipeline. It starts with a box labeled 'Database Specifications' which contains a list of items: 'CSV format', 'Heterogeneous data extracts', 'Database Writers', 'Database Viewers', and 'Algorithms'. An arrow points from 'Database Specifications' to 'Database Writers'. The 'Database Writers' box contains a list: 'In Situ', 'Post hoc', and 'ECP App'. The 'ECP App' item has a sub-list: 'In situ' and 'Post-processing'. An arrow points from 'Database Writers' to 'Database Viewers'. The 'Database Viewers' box contains a list: 'ALPINE', 'ASPECT', and 'ParaView'. An arrow points from 'Database Viewers' to 'Algorithms'. The 'Algorithms' box contains a list: 'Fuzzy logic', 'Machine learning', 'Bayesian networks', and 'Neural networks'.



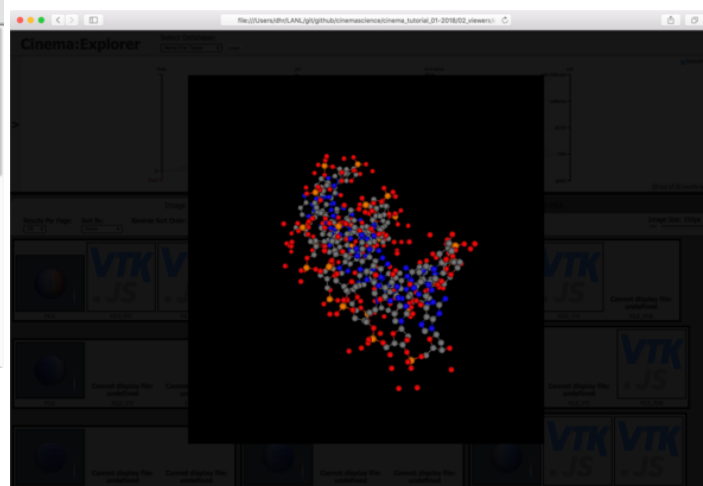
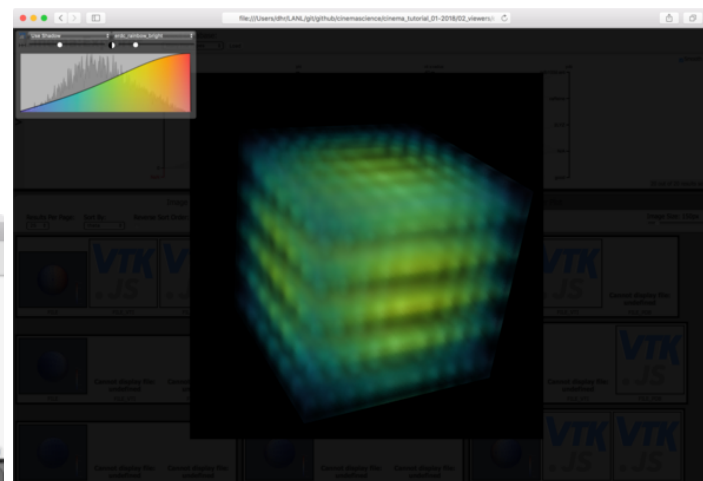
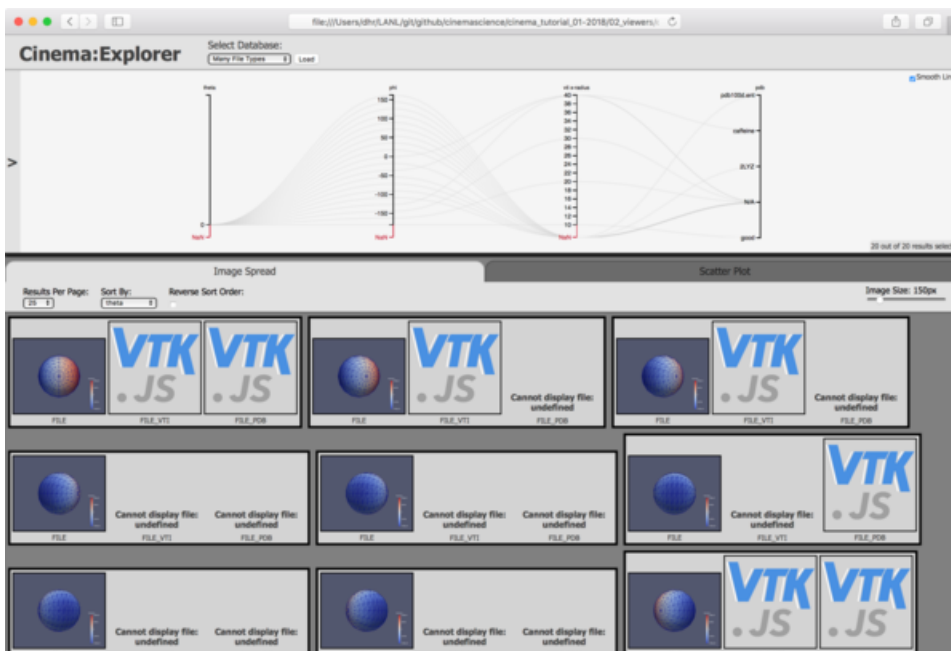
```
theta,phi,FILE
0,-180,image/-180/0.png
0,-162,image/-162/0.png
0,-144,image/-144/0.png
0,-126,image/-126/0.png
0,-108,image/-108/0.png
0,-90,image/-90/0.png
0,-72,image/-72/0.png
0,-54,image/-54/0.png
0,-36,image/-36/0.png
```

# The Cinema database specification

- CSV format
- Row  $\Leftrightarrow$  observation
- Column  $\Leftrightarrow$  metadata parameter, data variable, data extract, etc.
- Heterogeneous data extracts
  - Images and visualizations
  - Small simulation grids (.vti, .vtk, .pbd, etc.)
  - Output graphs and plots
  - Other relevant data files (.csv, .txt, .xlsx, etc.)

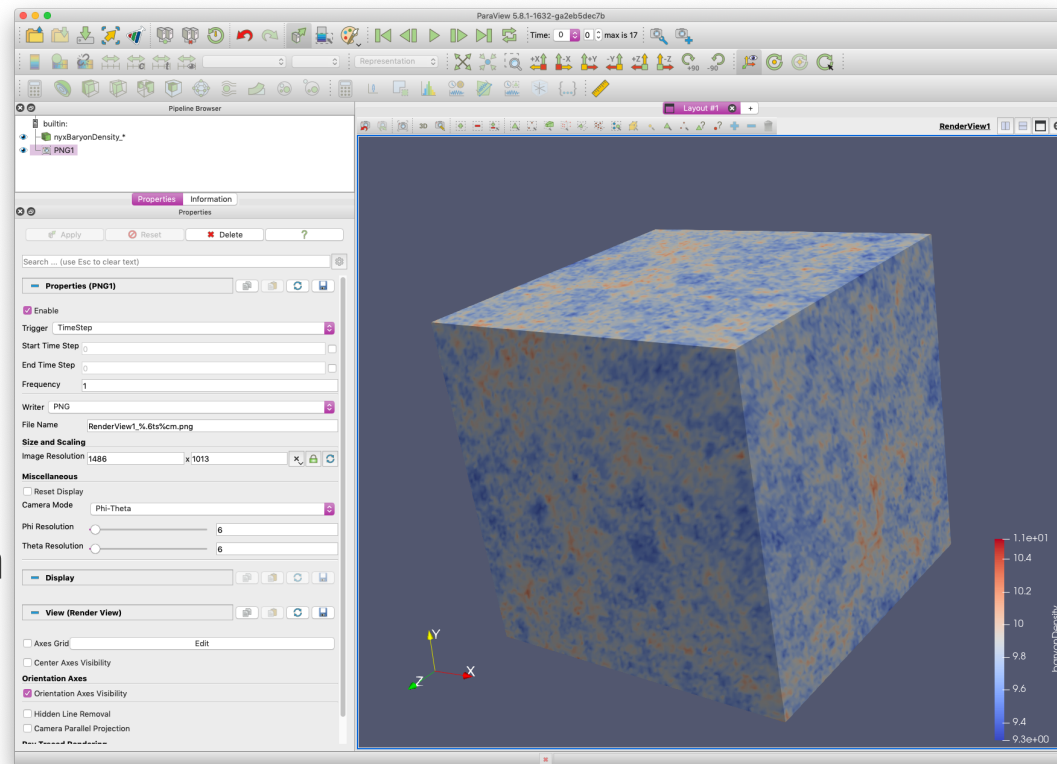


# Multiple artifact example



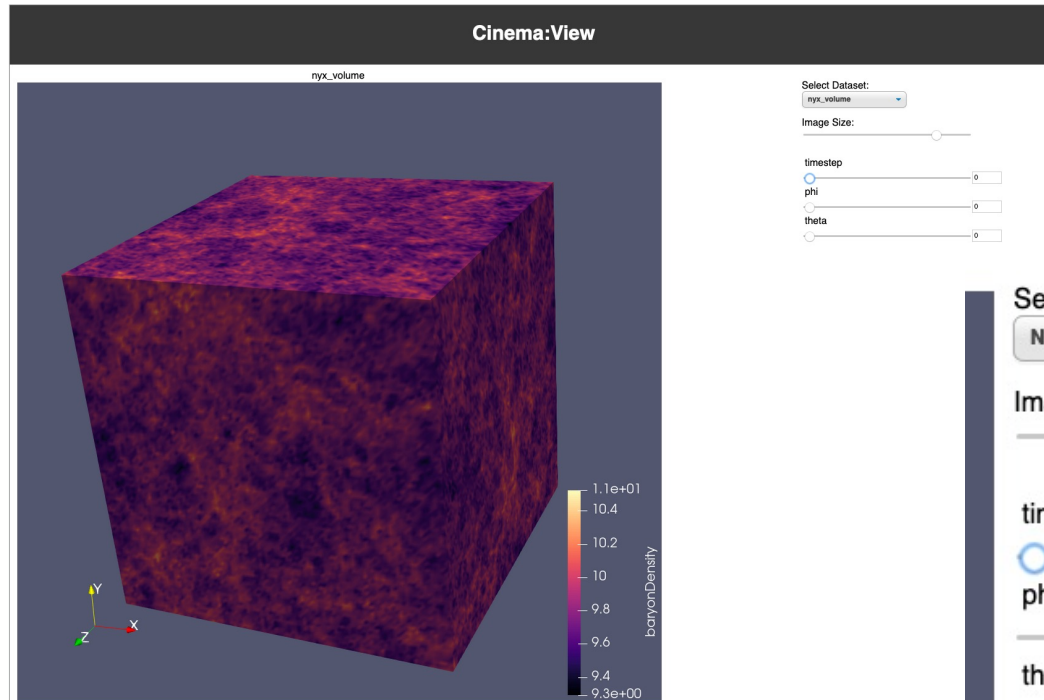
# Exporting a Cinema database

- ParaView
- VisIt
- ALPINE Ascent
- Direct from user code per specification
- Post hoc via shell script or python script

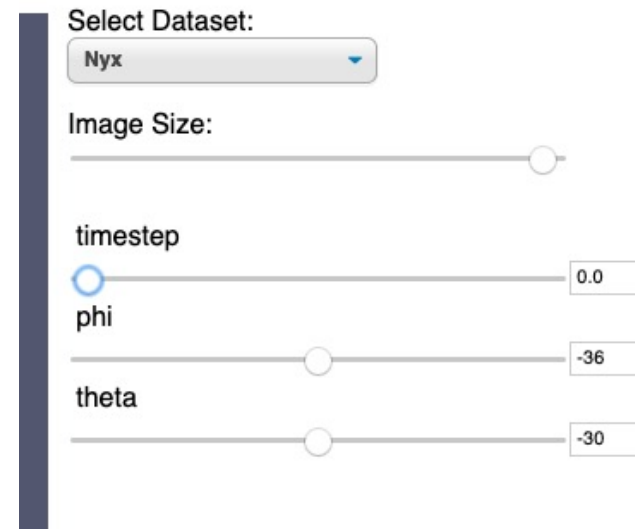


Nyx cosmology simulation: A. S. Almgren, J. B. Bell, M.J. Lijewski, Z. Lukic, E. Van Andel, "Nyx: A Massively Parallel AMR Code for Computational Cosmology" Astrophysical Journal, 765, 39, 2013. <https://amrex-astro.github.io/Nyx/index.html>

# Cinema Viewers



**Cinema:View** – Single or multiple databases can be viewed at same time. Typical usage: phi/theta/time

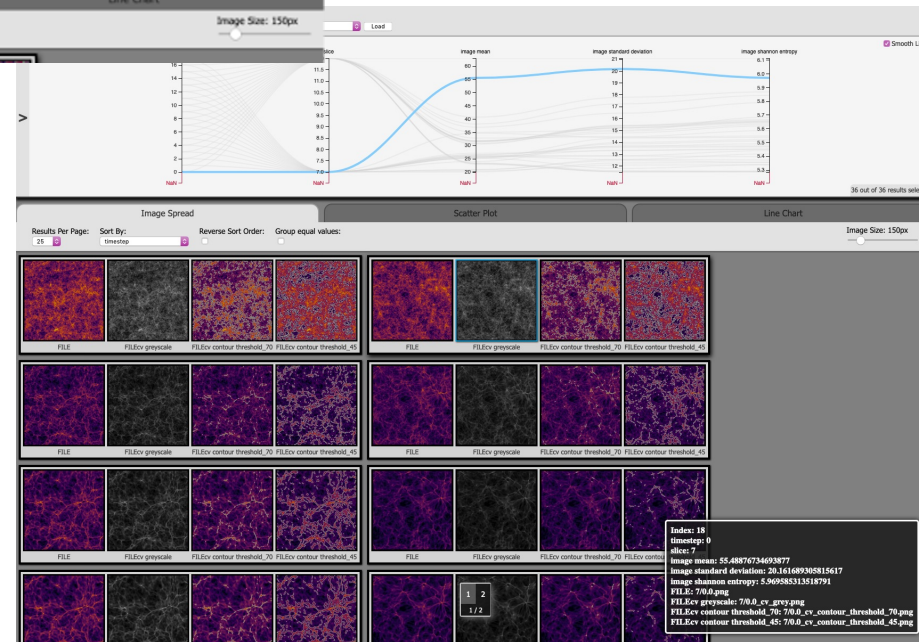
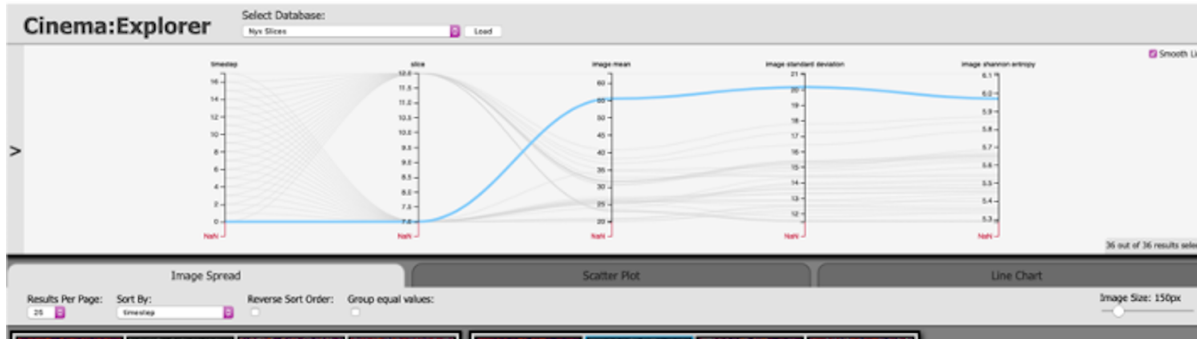




Nyx cosmology simulation: A. S. Almgren, J. B. Bell, M.J. Lijewski, Z. Lukic, E. Van Andel, "Nyx: A Massively Parallel AMR Code for Computational Cosmology" Astrophysical Journal, 765, 39, 2013. <https://amrex-astro.github.io/Nyx/index.html>

# Cinema Viewers

**Cinema:Explorer** – Parallel Coordinate view allows user to select, threshold, and interactively explore.



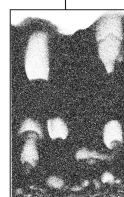
# MFiX-Exa Cinema Workflow

With thanks to J. Musser, A. Almgren, and the MFiX-Exa team

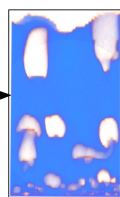
CFD-DEM simulation of a chemical looping reactor (CLR) -  $5 \times 10^9$  particles for 5 minutes of simulation time

Post hoc bubble dynamics analysis using in situ extracts generated from ALPINE statistical feature detection algorithm

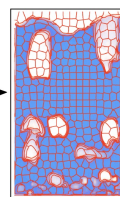
ALPINE in situ statistical feature algorithm identifies bubbles and extracts feature descriptors



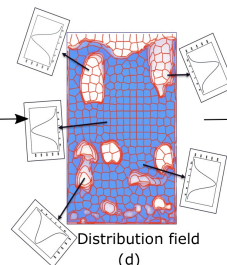
Raw Particle data  
(a)



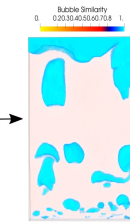
Particle Density field  
(b)



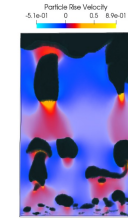
SLIC based partitioning  
(c)



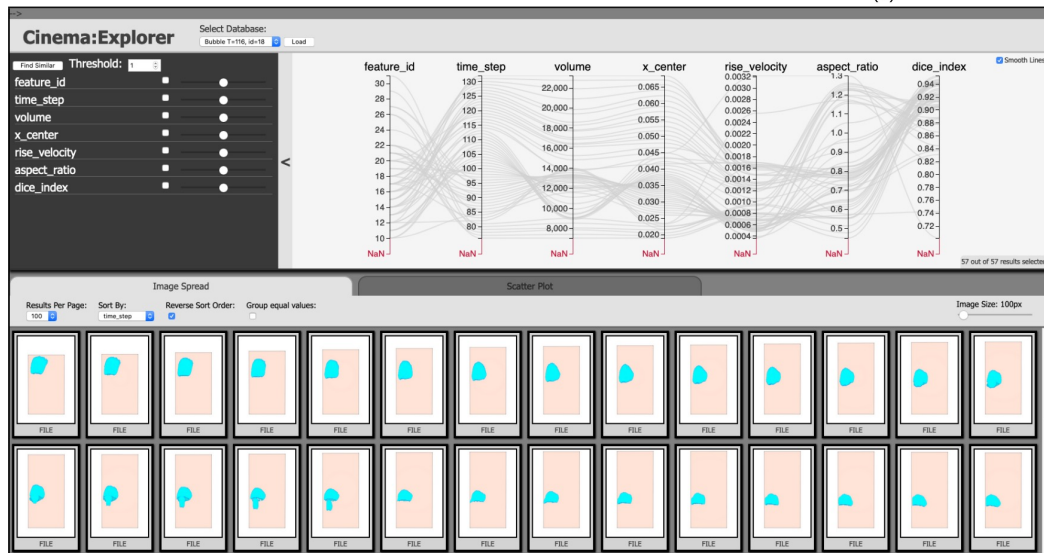
Distribution field  
(d)



Bubble Similarity field  
(e)



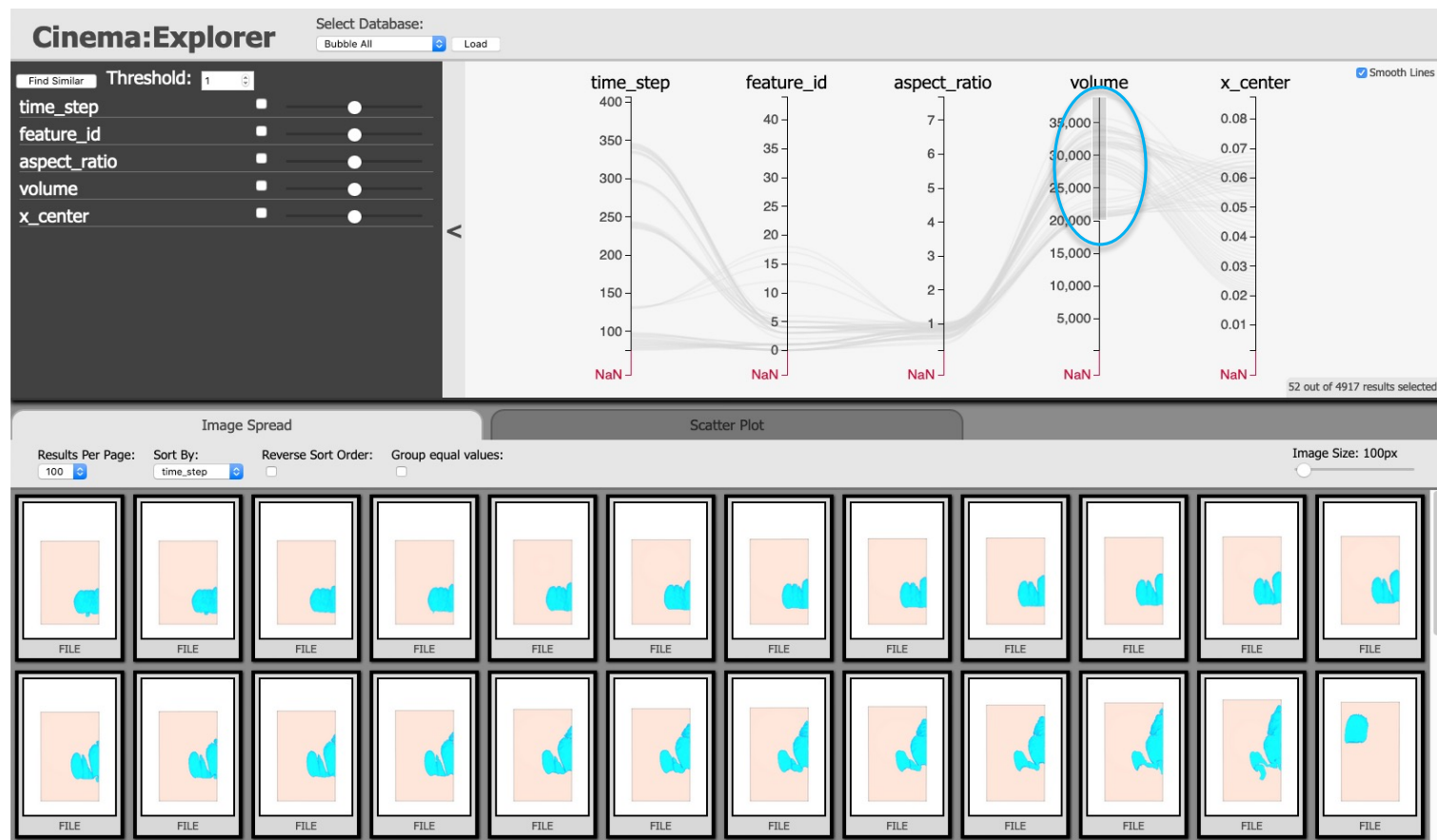
Particle Rise Velocity field  
(f)



Post hoc analysis extracts bubbles and bubble properties into a Cinema Database; use Cinema:Explorer to threshold, identify, & interactively explore bubble dynamics



# MFIX-Exa Cinema Workflow



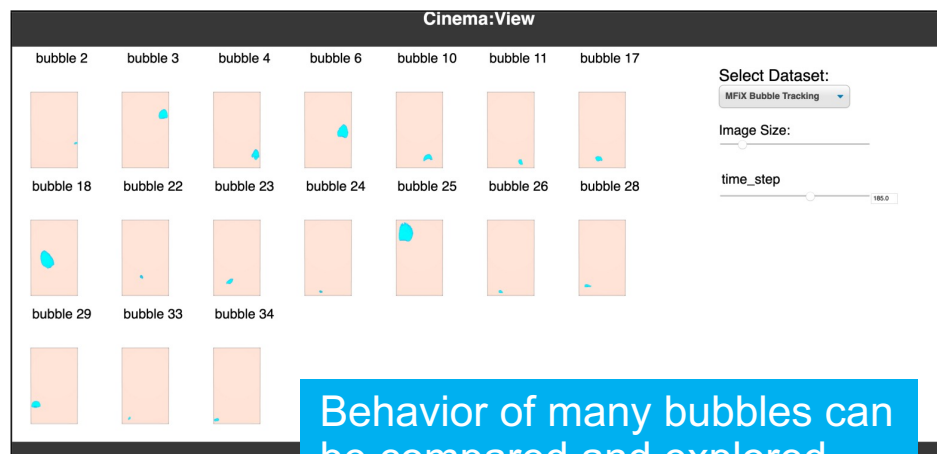
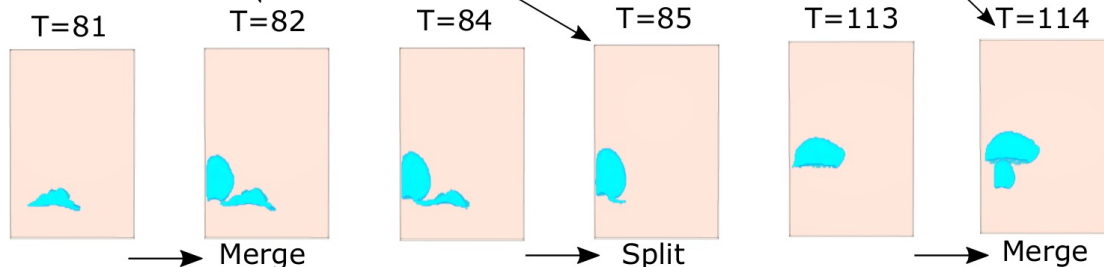
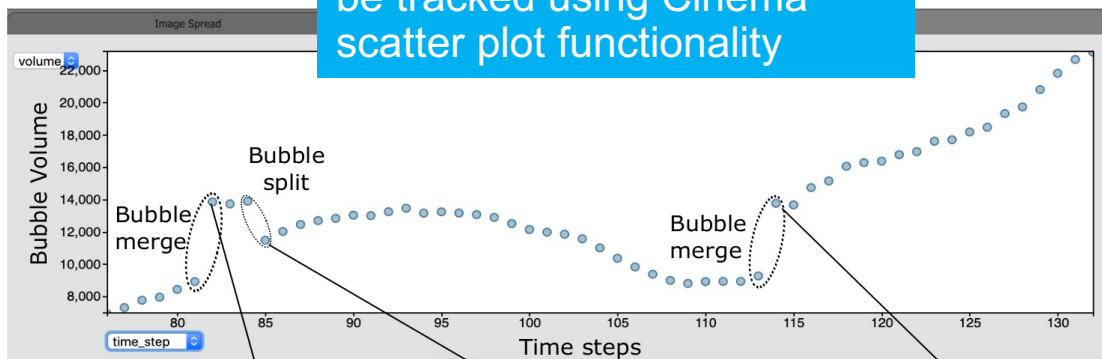
Use parallel coordinates to select on bubble volume and display specific bubble properties



# MFIX-Exa Cinema Workflow

Post hoc bubble dynamics analysis using in situ extracts generated from ALPINE statistical feature detection algorithm

Single bubble evolution can be tracked using Cinema scatter plot functionality



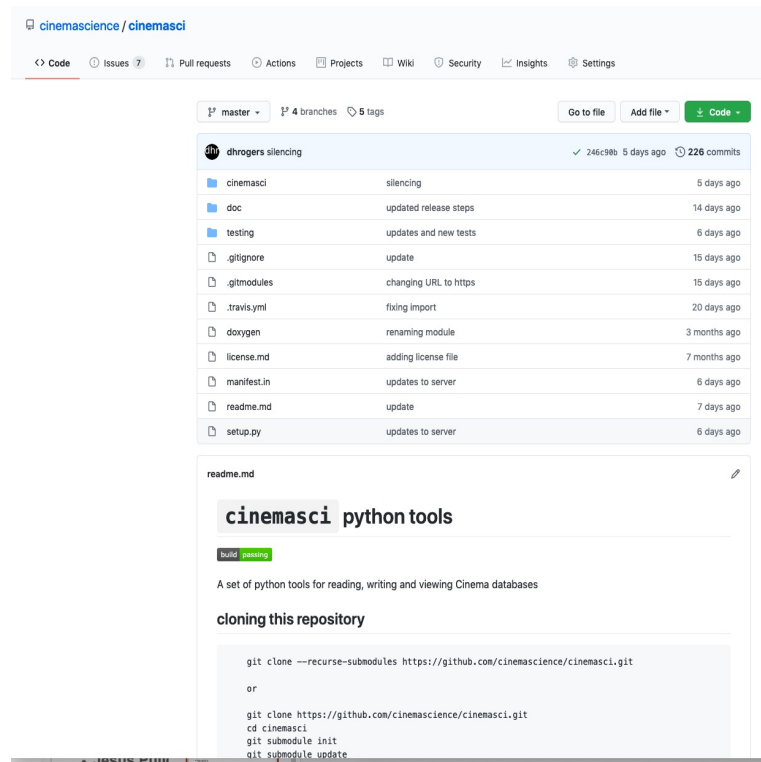
Behavior of many bubbles can be compared and explored over time with Cinema:View



# New Python cinemasci module

=> notebook workflow:

<https://github.com/cinemascience/cinemasci>



cinemascience / cinemasci

<> Code Issues 7 Pull requests Actions Projects Wiki Security Insights Settings

master 4 branches 5 tags Go to file Add file Code

dhrogers silencing 246c98b 5 days ago 226 commits

cinemasci	silencing	5 days ago
doc	updated release steps	14 days ago
testing	updates and new tests	6 days ago
.gitignore	update	15 days ago
.gitmodules	changing URL to https	15 days ago
.travis.yml	fixing import	20 days ago
doxygen	renaming module	3 months ago
license.md	adding license file	7 months ago
manifest.in	updates to server	6 days ago
readme.md	update	7 days ago
setup.py	updates to server	6 days ago

readme.md

## cinemasci python tools

build passing

A set of python tools for reading, writing and viewing Cinema databases

### cloning this repository

```
git clone --recurse-submodules https://github.com/cinemascience/cinemasci.git
or
git clone https://github.com/cinemascience/cinemasci.git
cd cinemasci
git submodule init
git submodule update
```

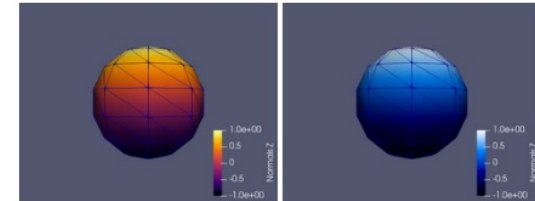


# Python notebook workflow

```
In [1]: import cinemasci

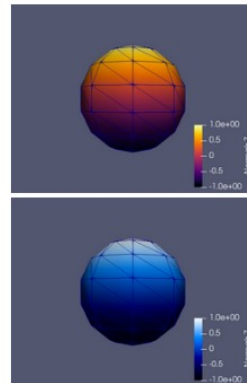
# create a viewer object
viewer = cinemasci.pynb.CinemaViewer()
# optionally set the layout of the viewer
viewer.setLayoutToHorizontal()
# optionally set the height of the viewer
viewer.setHeight(250)
# load one or more cinema databases
viewer.load("data/sphere_red.cdb data/sphere_blue.cdb")
```

timestep  0.0  
 producer  cview\_0  
 phi  -180.0  
 theta  -90.0  
 image size  213



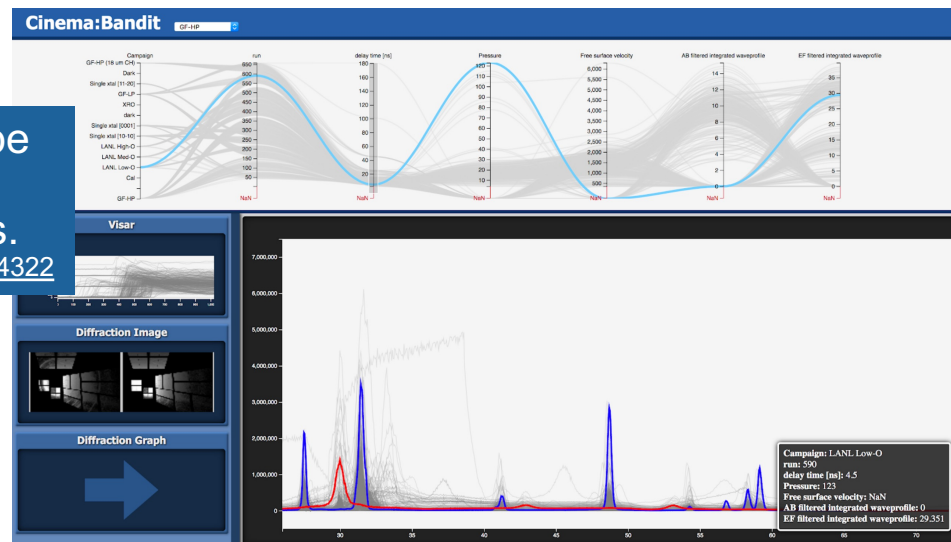
```
In [2]: # example with defaults
viewer2 = cinemasci.pynb.CinemaViewer()
viewer2.load("data/sphere_red.cdb data/sphere_blue.cdb")
```

timestep  0.0  
 producer  cview\_0  
 phi  -180.0  
 theta  -90.0  
 image size  216



# Cinema Viewers

Modular components can be combined to design application-specific viewers.  
Orban et al. DOI: 10.1107/S1600577519014322



Cosmology Simulation  
Compression study using  
Foresight

Grosset et al. DOI:10.1109/SC41405.2020.00087

NERSC Science Portal  
radiography data set





# Additional Resources

- Cinema github
  - <https://github.com/cinemascience>
- Cinema website
  - <https://www.cinemascience.org>

