LA-UR-21-22952

The Cinema Project

Terry Turton

ECP Community BOFs

Los Alamos NATIONAL LABORATORY

NNSA National Nuclear Security Administration

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

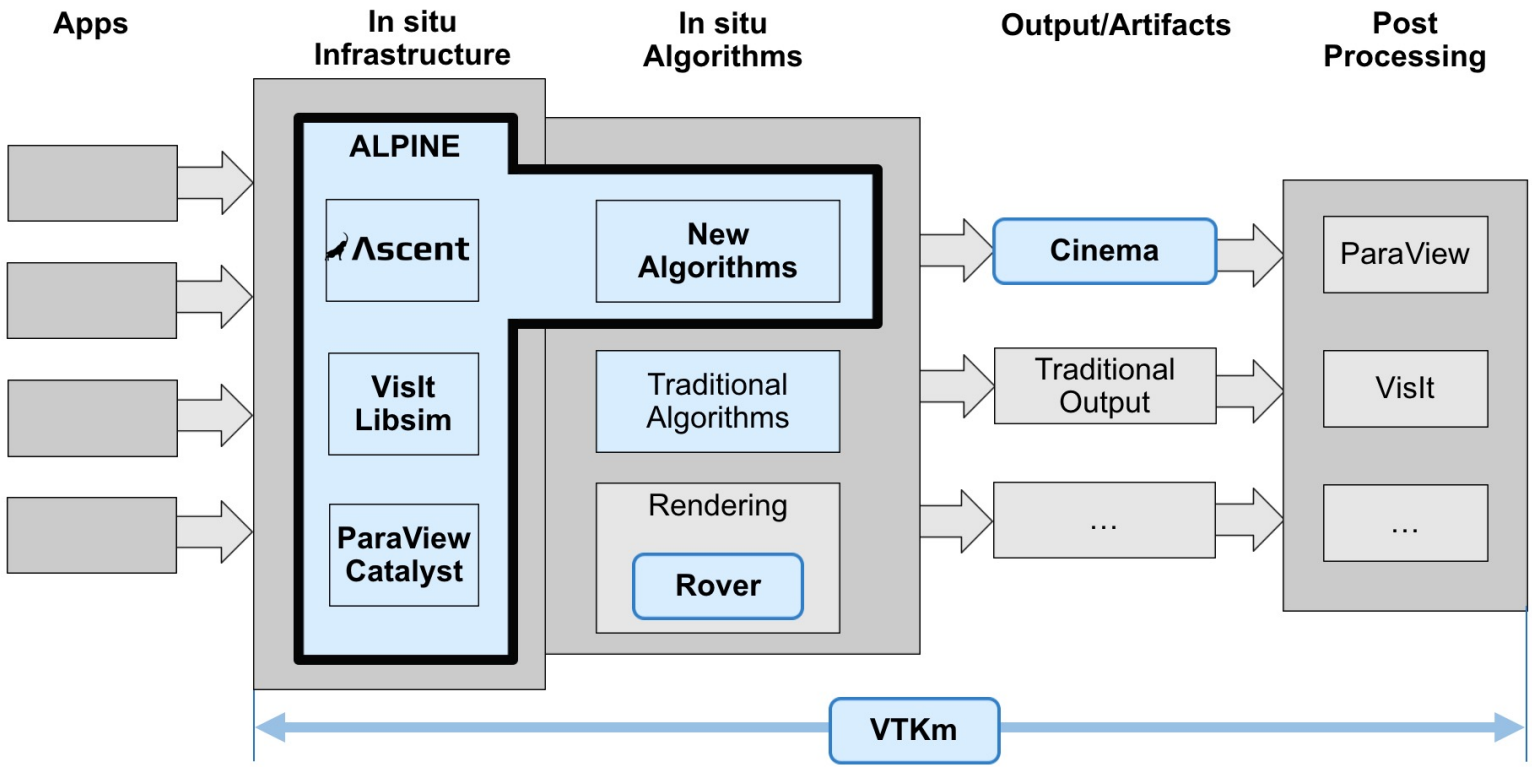
Data Science at Scale Team, Information Sciences, Los Alamos National Laboratory

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

Acknowledgement

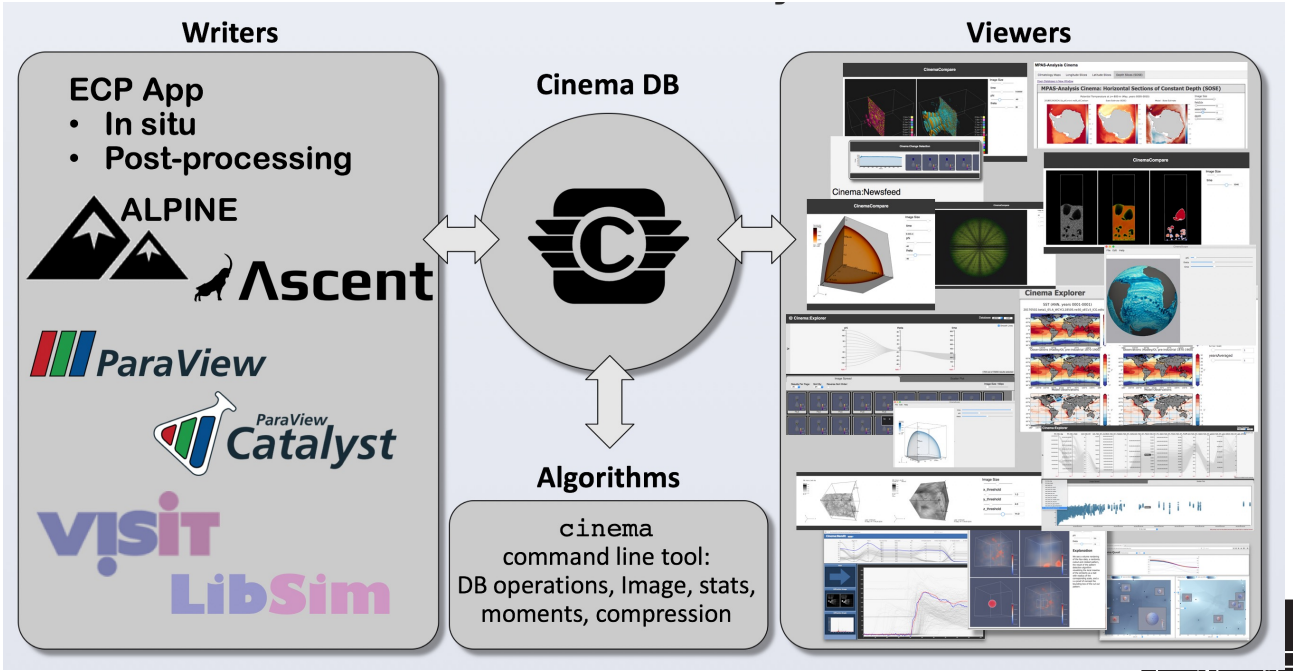
This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology,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

Cinema within the ECP In Situ Workflow

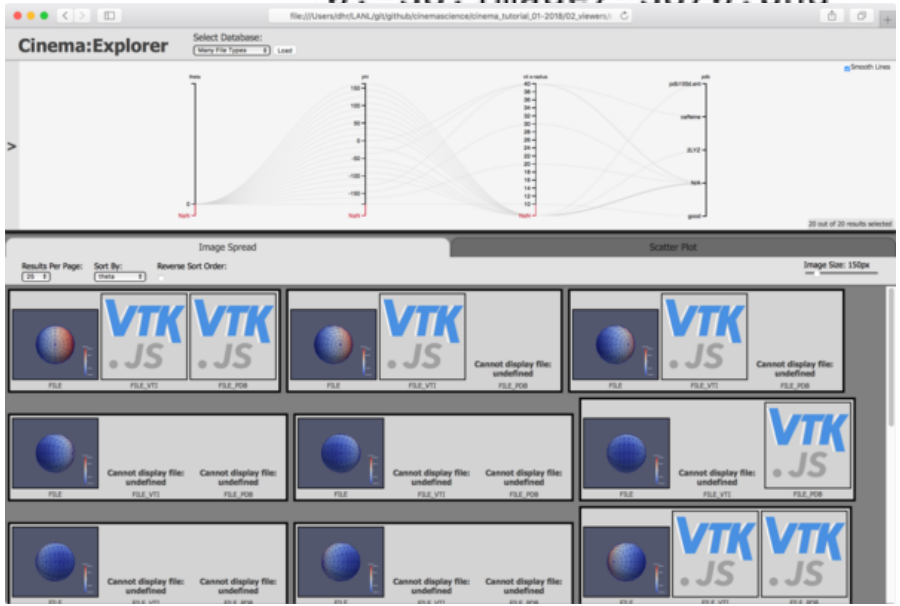


The Cinema Ecosystem

- Database Specifications
  - CSV format
  - Heterogeneous data extracts
- Database Writers
  - In Situ
  - Post hoc
- Database Viewers
- Algorithms



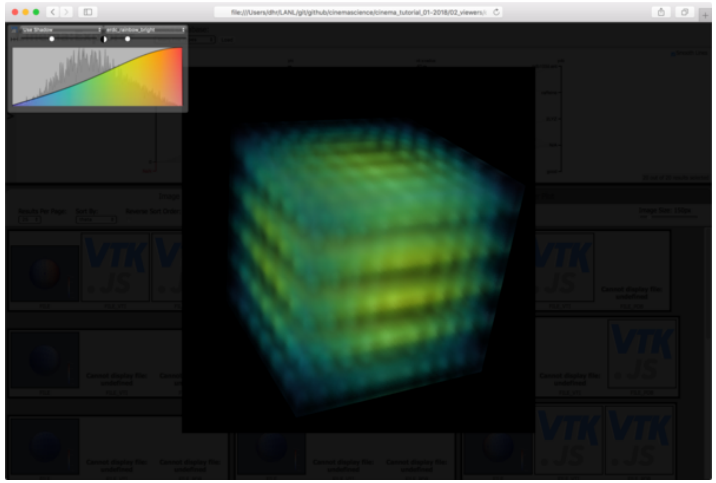
```
theta,phi,FILE
0,-180,image/-180/0.png
0,-162,image/-162/0.png
0,-144,image/-144/0.png
0,-126,image/-126/0.png
0,-108,image/-108/0.png
0,-90,image/-90/0.png
0,-72,image/-72/0.png
0,-54,image/-54/0.png
0,-36,image/-36/0.png
```

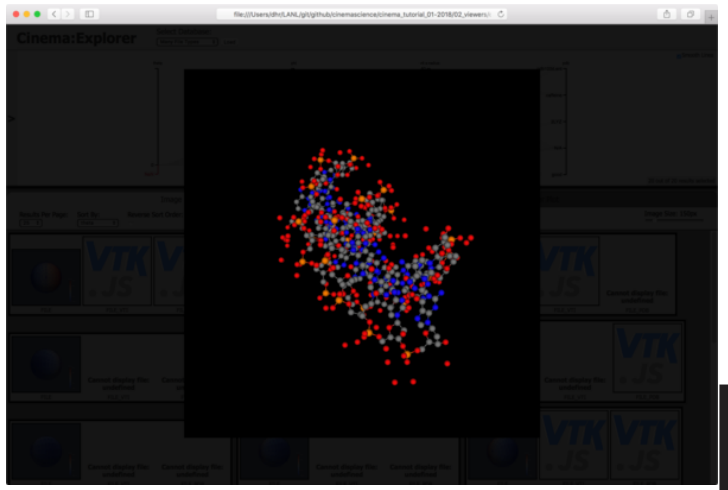


The Cinema database specification

- CSV format
- Row <=> observation
- Column <=> 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.)

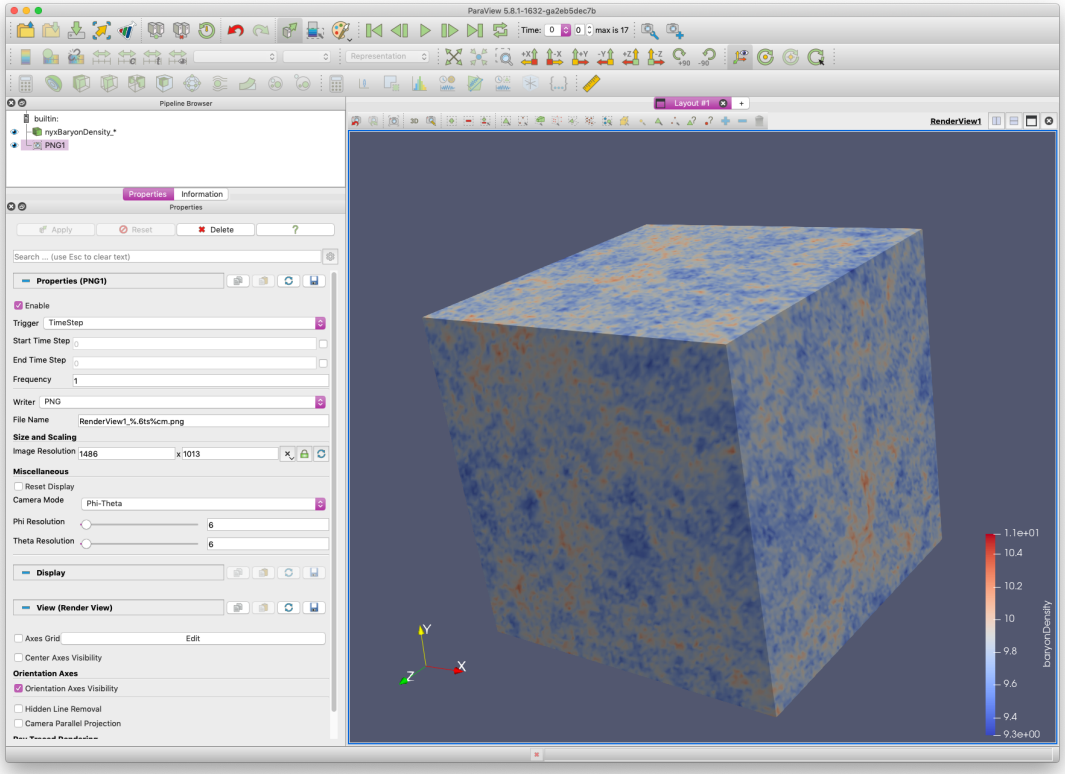
Multiple artifact example





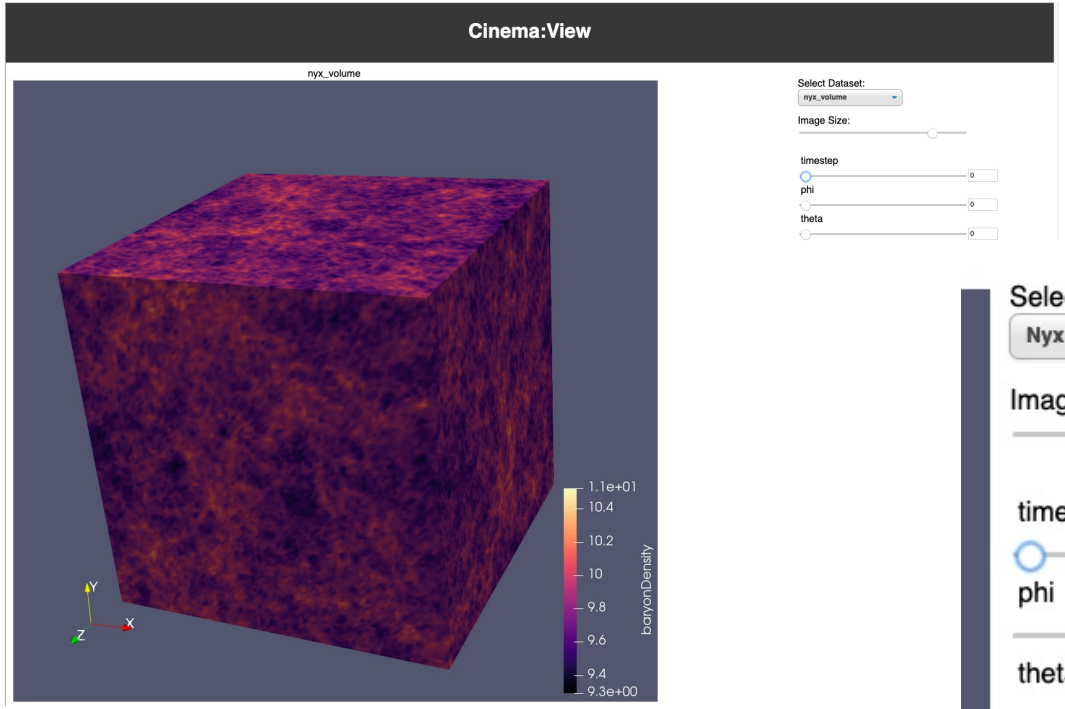
Exporting a Cinema database

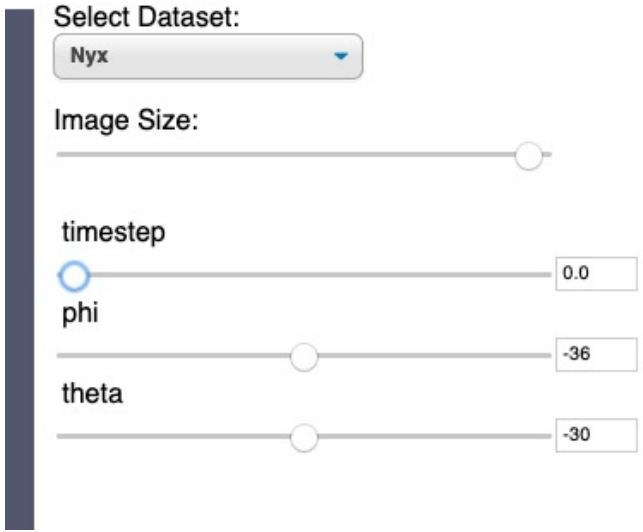
- ParaView
- VisIt
- ALPINE Ascent
- Direct from user code per specification
- Post hoc via shell script or python script



Cinema Viewers

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

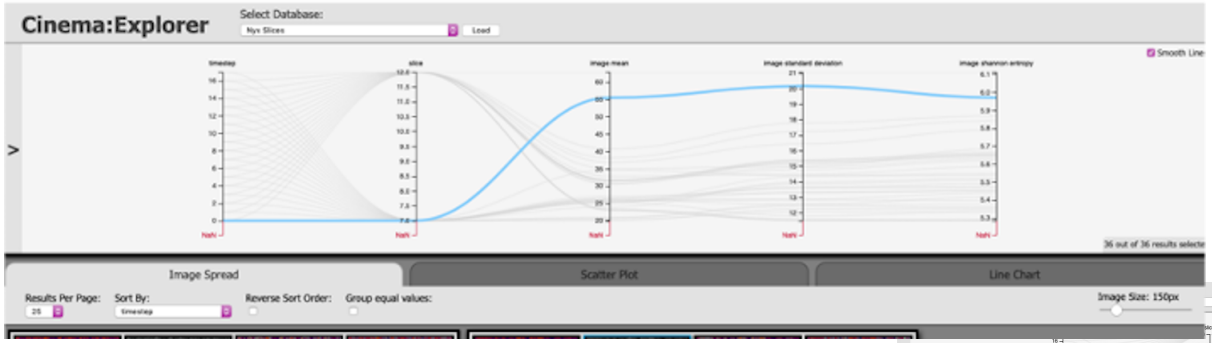




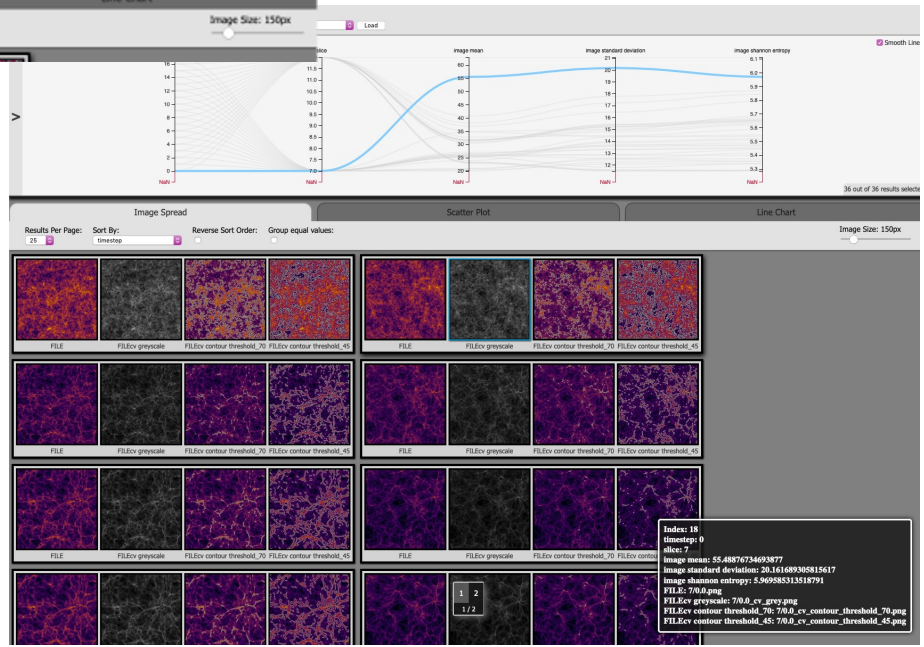
Cinema:View – Single or multiple databases can be viewed at same time. Typical usage: phi/theta/time

Cinema Viewers

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



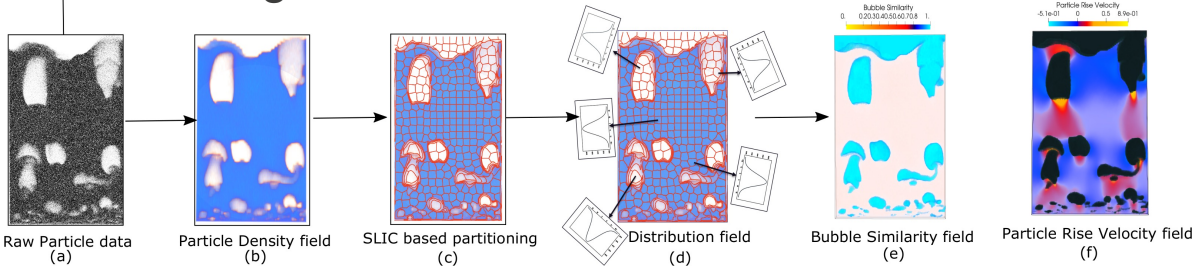
MFiX-Exa Cinema Workflow

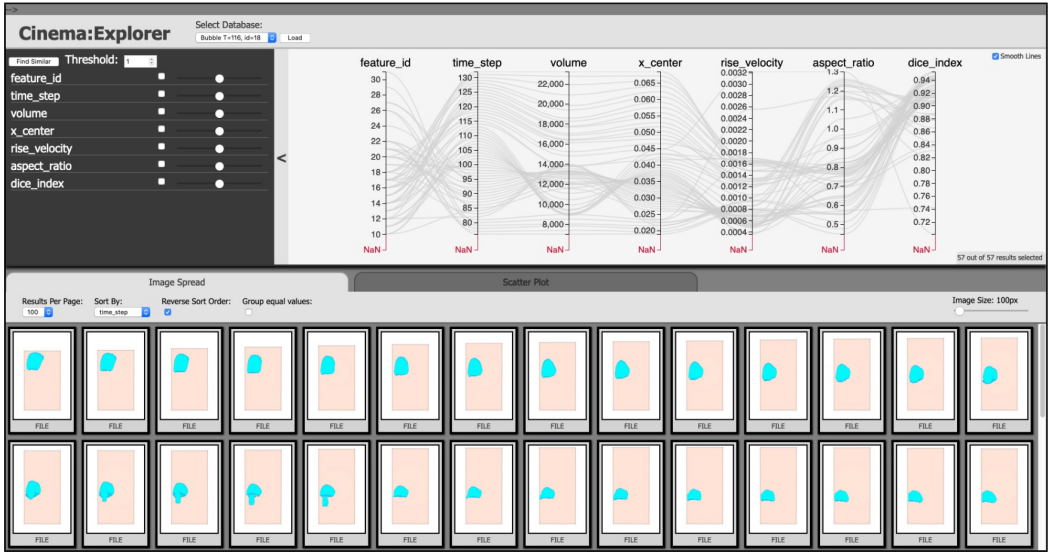
With thanks to J. Musser, A. Almgren, and the MFiX-Exa team

CFD-DEM simulation of a chemical looping reactor (CLR) - $5x10^9$ particles for 5 minutes of simulation time

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

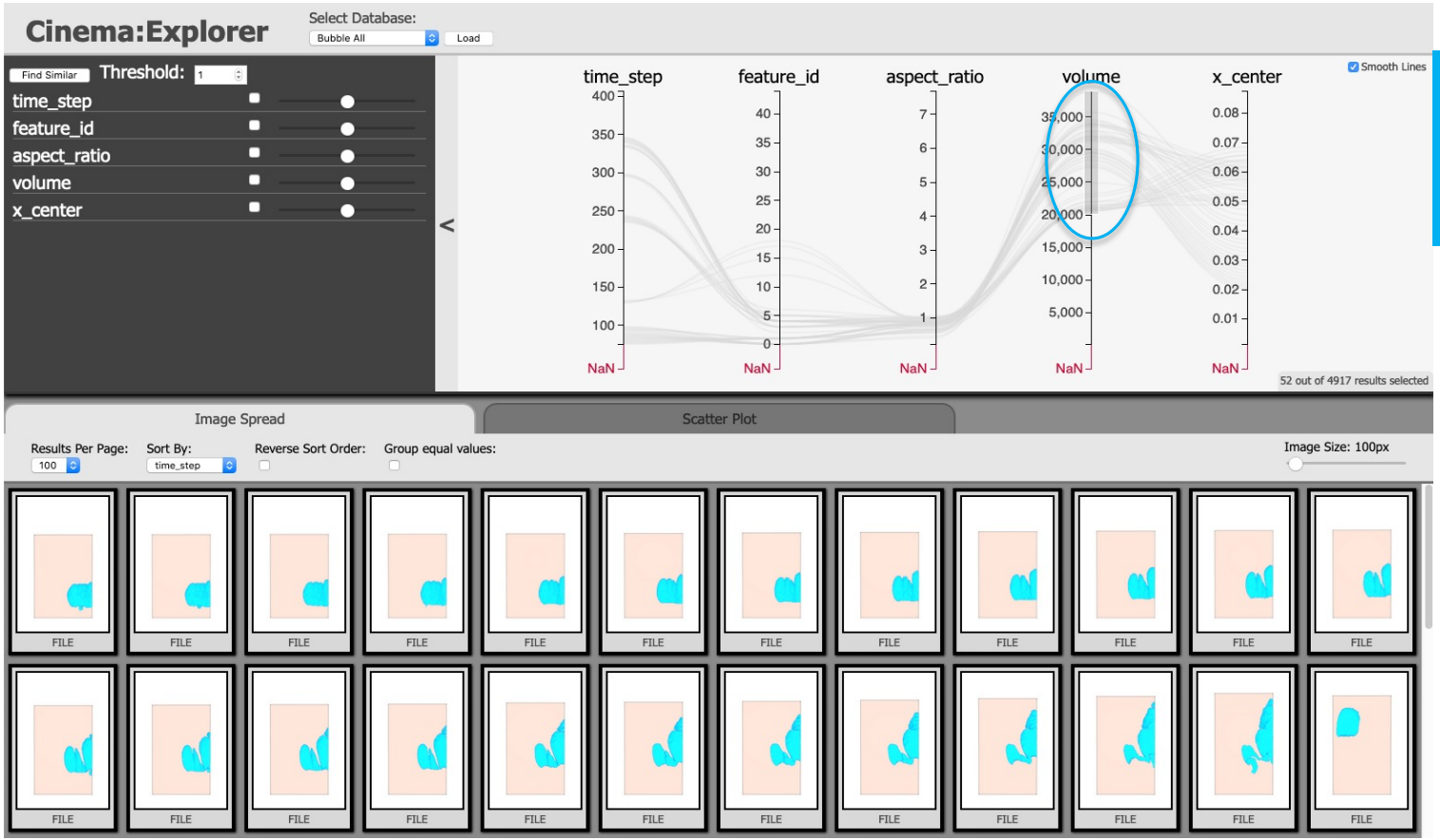
ALPINE in situ statistical feature algorithm identifies bubbles and extracts feature descriptors





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

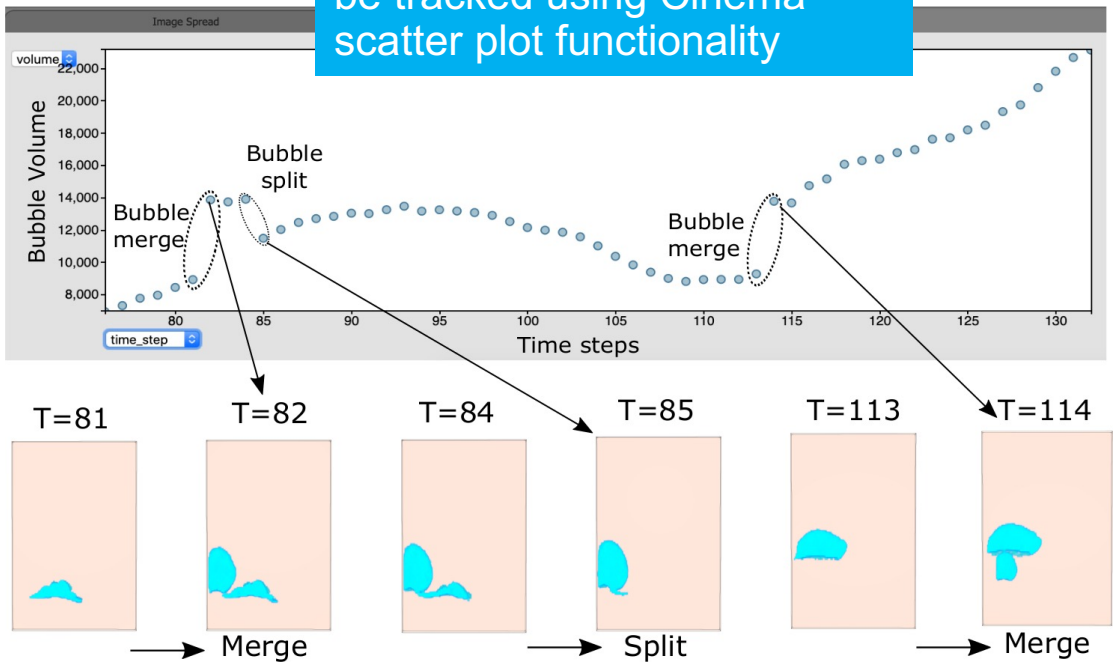


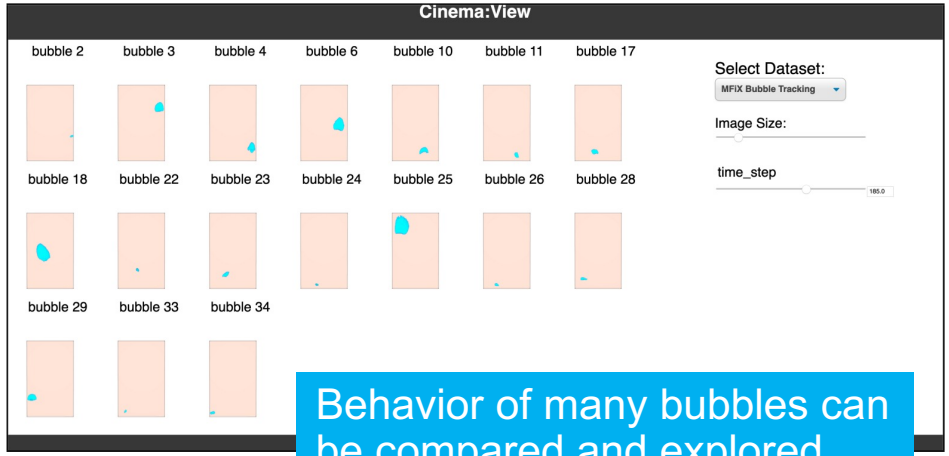
Use parallel coordinates to select on bubble volume and display specific bubble properties

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



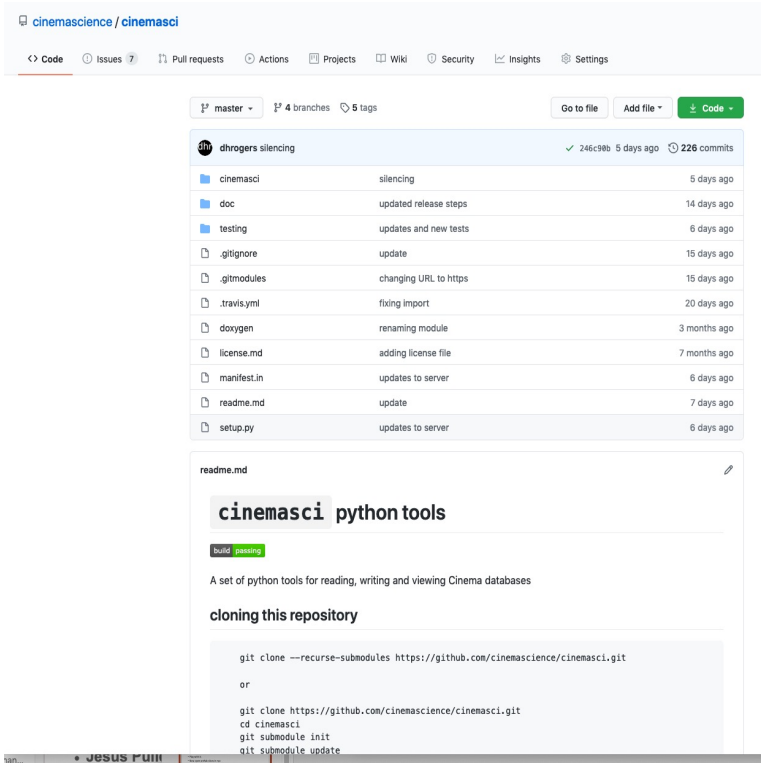


Behavior of many bubbles can be compared and explored over time with Cinema:View

New Python `cinemasci` module

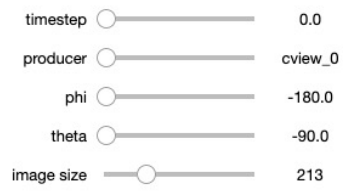
=> notebook workflow:

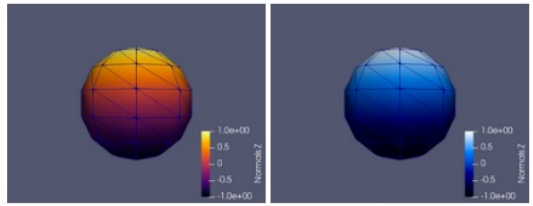
`https://github.com/cinemascience/cinemasci`



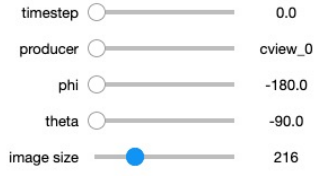
Python notebook workflow

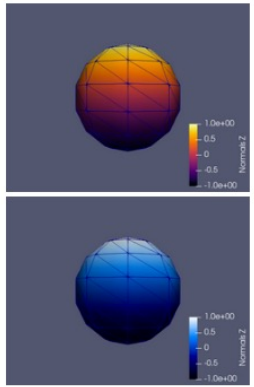
```
In [1]: import cinemasci

        # 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")
```





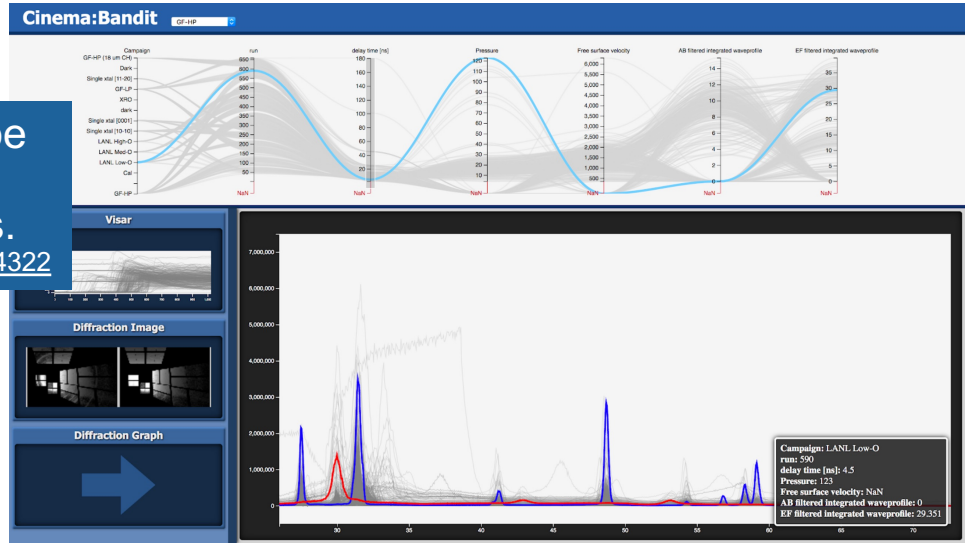
```
In [2]: # example with defaults
        viewer2 = cinemasci.pynb.CinemaViewer()
        viewer2.load("data/sphere_red.cdb data/sphere_blue.cdb")
```

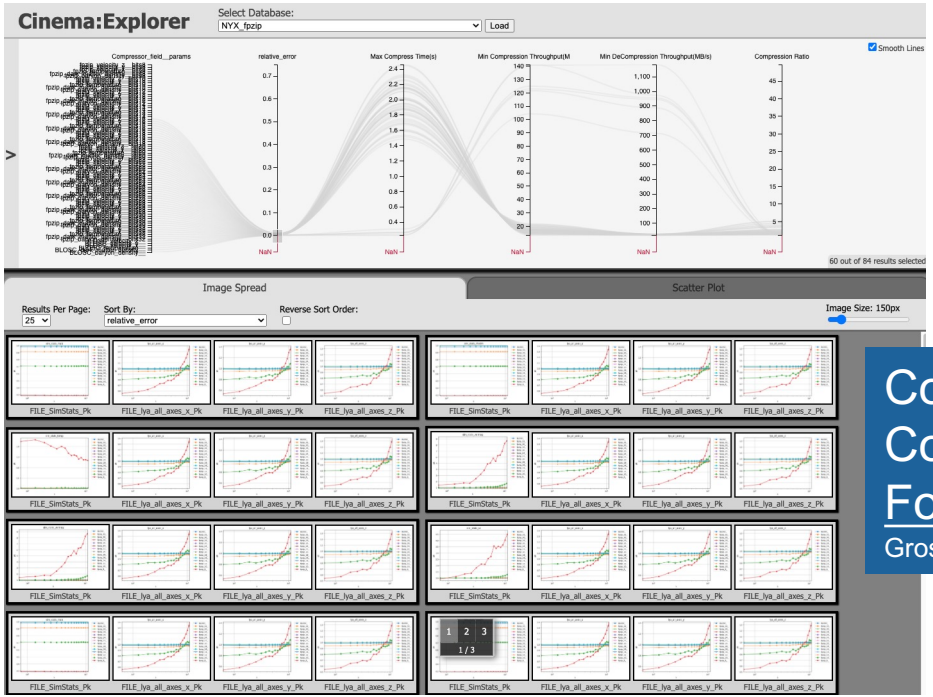




Cinema Viewers

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