Fountains and garden-hoses: Visualizing the intricacies of primary jet atomization

Greg Foss
Texas Advanced Computing Center
University of Texas at Austin
Austin, Texas, USA
foss@tacc.utexas.edu

Baskar Ganapathysubramanian
Department of Mechanical Engineering
Iowa State University
Ames, Iowa, USA
baskarg@iastate.edu

Masado Ishii
School of Computing
The University of Utah
Salt Lake City, Utah, USA
masado@cs.utah.edu

Makrand Ajay Khanwale
Center for Turbulence Research
Stanford University
Stanford, California, USA
khanwale@stanford.edu

Kumar Saurabh
Department of Mechanical Engineering
Iowa State University
Ames, Iowa, USA
maksbh@iastate.edu

Dave Semeraro
Texas Advanced Computing Center
University of Texas at Austin
Austin, Texas, USA
semeraro@tacc.utexas.edu

Hari Sundar
School of Computing
The University of Utah
Salt Lake City, Utah, USA
hari@cs.utah.edu

Sean Cunningham
Texas Advanced Computing Center
University of Texas at Austin
Austin, Texas, USA
seanc@tacc.utexas.edu

Link to the video: Visualizing the intricacies of primary jet atomization

***Abstract*—A jet of fluid – when we open a garden hose, for instance – exhibits a rich tapestry of flow physics. This includes the rupture of fluid films, the breakup of filaments in droplets, and a cascade of droplet breakup and coalescence. In addition to its breathtaking beauty – as evidenced by our fondness for fountains and waterfalls – this jet atomization is a critical component for a broad spectrum of applications in the energy, healthcare, entertainment, and coatings industries. Simulating and visualization jet atomization is an ideal way to understand and control this phenomenon. However, the multiscale nature of jet atomization makes this a very challenging problem. Here, we visualize one of this phenomenon's highest resolution simulation datasets. The dataset consists of over 120000 time steps of an adaptively resolved spatial mesh with length scales spanning three orders of magnitude totaling over 15 TB of data. The simulation required over 200000 node hours on TACC Frontera. We describe the parallel workflow and associated challenges while visualizing the time evolution of the fluid-air interface isosurface from this dataset. We show how this visualization produces a deep qualitative understanding of fluid dynamics and provides quantitative metrics of the instabilities. Such visualization-enabled analysis provides the best way to understand the outputs of these massive simulations.**

I. Introduction

A jet of a heavier density fluid (like water) injected into a lighter density fluid (like air) exhibits a rich tapestry of flow physics phenomena. This process, called (primary) jet atomization, is harnessed in a dizzying array of engineered applications. These include fuel injectors in combustion engines, various types of printing (think additive manufacturing), and most coating and spraying operations (think industrial painting and agriculture).

Understanding the jet atomization process – particularly the instabilities that result in the rupture of fluid films, followed by the instabilities that cause the breakup of filaments – can transform how we design and control a wide array of engineered systems. However, jet atomization has proved notoriously challenging to understand, primarily due to the presence of a wide range of spatial scales [1, 2] and the long time horizons over which these instabilities are triggered and exhibited.

Computationally understanding jet atomization is complicated by a cascade of challenges. First, novel computational techniques must be deployed at scale to accurately model the atomization process. The multiple spatial and temporal scales call for spatially adaptive and temporally higher-order methods to capture the instabilities one is interested in. Second, these simulations produce a huge amount of spatio-temporal data that makes manual exploration non-trivial. In particular, the data is produced on underlying meshes that are non-uniform due to the spatially varying resolution needed for resolving multiple scales involved. Finally, exploring these huge datasets via visual analytics becomes exceedingly difficult but is crucial for qualitative understanding. Thus, the design of efficient and revealing visualization workflows becomes critical. This serves as the motivation for our ***exploration*** through visualization effort.

Here, we present an extremely effective visualization workflow that allows us to explore and understand the underlying mechanisms in jet atomization. The visualization workflow uses a state-of-the-art massively parallel simulation that provides the highest known resolution to capture the rich multi-scale physics involved in jets. In particular, we show that these visualization workflows qualitatively reveal the physical mechanisms of how gas-fluid films and filaments break up into smaller filaments and drops. In addition, the workflows are able to extract quantitative metrics of the location of instabilities and droplet distribution that can serve as benchmarks (for other simulations) as well as datasets for potential AI/ML workflows.

II. Simulation details

To understand jets, the dataset is generated using a highly resolved simulation of a representative (canonical) problem. This canonical problem is of a high-velocity pulsating jet of fluid entering a chamber containing stagnant air. The two-phase flow is modeled using thermodynamically consistent Cahn-Hilliard Navier-Stokes equations. The underlying numerical method and equations are detailed in [3]. In this context, the interface between the liquid jet and air is diffused such that it can be captured using the underlying mesh. As the jet enters and pulsates at high velocity, it disintegrates into films that rupture to form filaments and subsequently break into small droplets forming the primary phase of jet atomization.

The dominant mechanisms of the initial breakup of these jets are not very well understood [1]. In this two-phase flow regime, the inertial hydrodynamic scales are much stronger than surface tension forces. However, surface tension is not zero. The non-zero surface tension influences the stability and breakup of films and filaments generated in the evolution of jets. Particularly, if the correct numerical resolution is not achieved, non-physical film rupture is observed, whereas physical film rupture is very difficult to resolve as the thickness of these rupturing films are orders of magnitude smaller than the large length scales in jets. In the current simulation, appropriate mesh adaption is provided to always ensure appropriate mesh resolution to capture physically meaningful film rupture. See fig. 1 for an example visualization of realistic film rupture.

Film rupture generally has a characteristic rounded smoothness due to local surface tension as the film ruptures and recedes to form filaments called Taylor-Culick rims [4, 5, 6]. The simulation that generates the dataset affords one of the highest resolutions ever used to resolve these rich physics. This is done by adaptively providing the targeted mesh resolution needed to resolve these breakup phenomena. In addition to film rupture, filaments can be formed due to local instabilities either following film rupture or independently. These filaments subsequently break up into smaller droplets which are also resolved using adaptive mesh resolution. See fig. 2 for visualization examples of filament breakup.

The aforementioned microscale mechanisms can now be understood in depth through detailed visualizations from the dataset. However, with adaptive mesh resolutions spanning three orders of magnitude to capture all these fine-scale physics, the dataset has highly non-uniform mesh, and the underlying dataset is massive, occupying about 15TB. Effective visualization here provides never-before unprecedented access to fine-scale details to understand the rich tapestry of breakup phenomena that will push our understanding further than ever before. All of this is capable due to efficient deployment of simulation and then visualization workflow in a large-scale HPC platform of TACC Frontera. The simulation achieves an equivalent [1] mesh resolution of 35 Trillion mesh points. In comparison, the previous best-resolved simulation of this canonical example used an equivalent mesh of 549 billion [7] mesh points.

In the accompanying video, we evolve the jet. The video is then stopped to show a slice visualization of the adaptive mesh resolution of the simulation. We then let the jet evolve, and as it becomes unstable and ruptures, we stop the video and zoom in to show an example of a physical film rupture. We then let the jet continue to evolve and form more filaments. The video is stopped again, and we zoom in to show the droplet pinch off from these filaments. In addition to probing the breakup, the massive dataset also contains velocities and pressures, which the visualization can explore by going back and forth in time snapshots to explore the hydrodynamic conditions underlying the breakup mechanisms whose understanding is limited in the literature, truly pushing the boundaries of our understanding of the complex physics involved.

III. Visualization

This section describes the visualization of the jet dataset. The process followed the familiar pattern of raw data reduction and cleaning, followed by the application of visualization tools. We first describe the raw data and how it was prepared for visualization.

A. Raw Data

The simulation produced several time steps. The data was arranged on a large parallel file system in directories corresponding to the time step. As is often the case, as the simulation progressed, the fields became more complex, the mesh was adapted and refined, and the time step checkpoints became more voluminous. The initial time state was 16 GB in size. The final time state solution used in the production of this video was 473 GB in size. There were 222 timestep checkpoint states saved as of this writing.

The data was stored in `VTK` parallel unstructured grid format. The data was divided into categories with different variables written to different `.vtu` files. For example fluid velocity data was written to `vel_<timestep>.pvtu` files. There were 14560 `.vtu` files or parts associated with the

[1] If the finest resolution was used as a uniform resolution in the whole domain

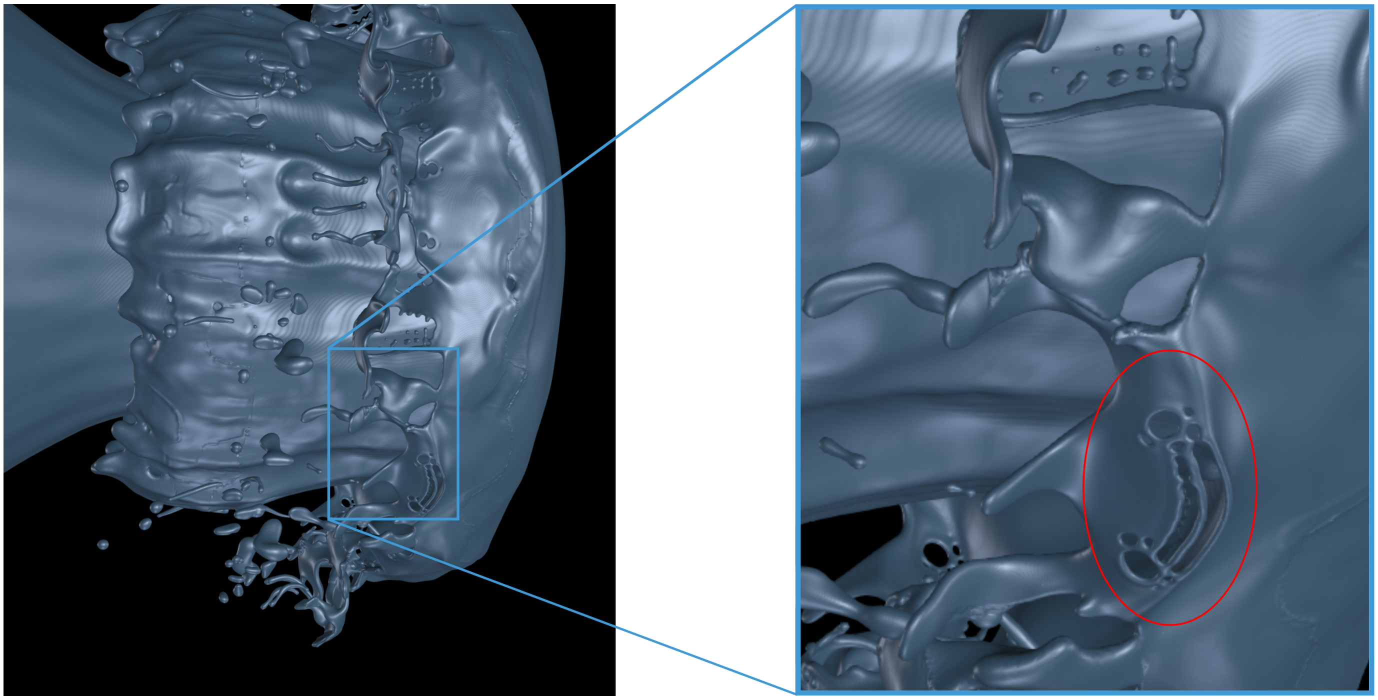

Fig. 1. A snapshot of the pulsating jet. The image on the left shows a visualization of the liquid-air interface (isosurface of $\phi = 0$). Notice the leading surface of the jet has bloomed out and formed a film. This film is undergoing rupture to form fluid filaments. The figure on the right zooms into one such rupture event, shown within the red ellipse.

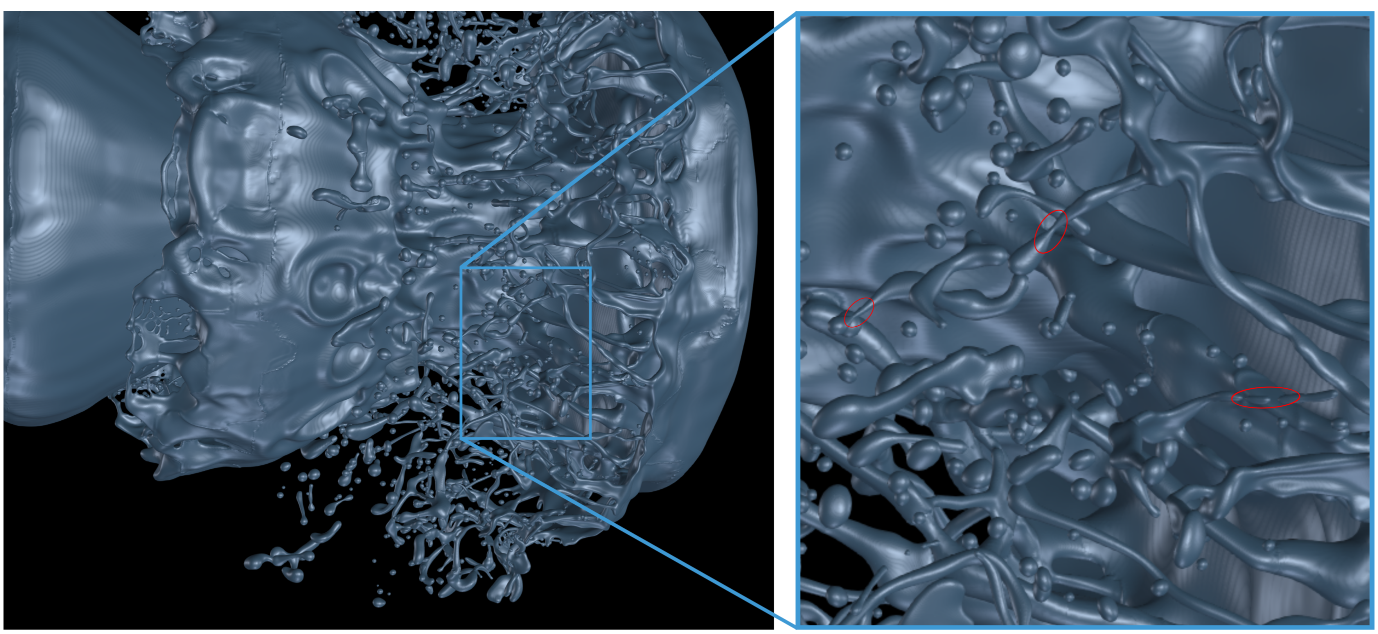

Fig. 2. A snapshot of the jet at a later time. Notice the dramatically more complex structures formed due to a cascade of film ruptures in the pulsating jet. We see a number of fluid filaments connected to the leading film and free-standing in the surrounding air. These filaments undergo droplet pinch-off to produce a wide distribution of droplets. The figure on the right zooms in and identifies three such droplet formation events seen within the red ellipses. Such visualization tools allow us to track the fate of each of these droplets to construct qualitative measures of atomization.

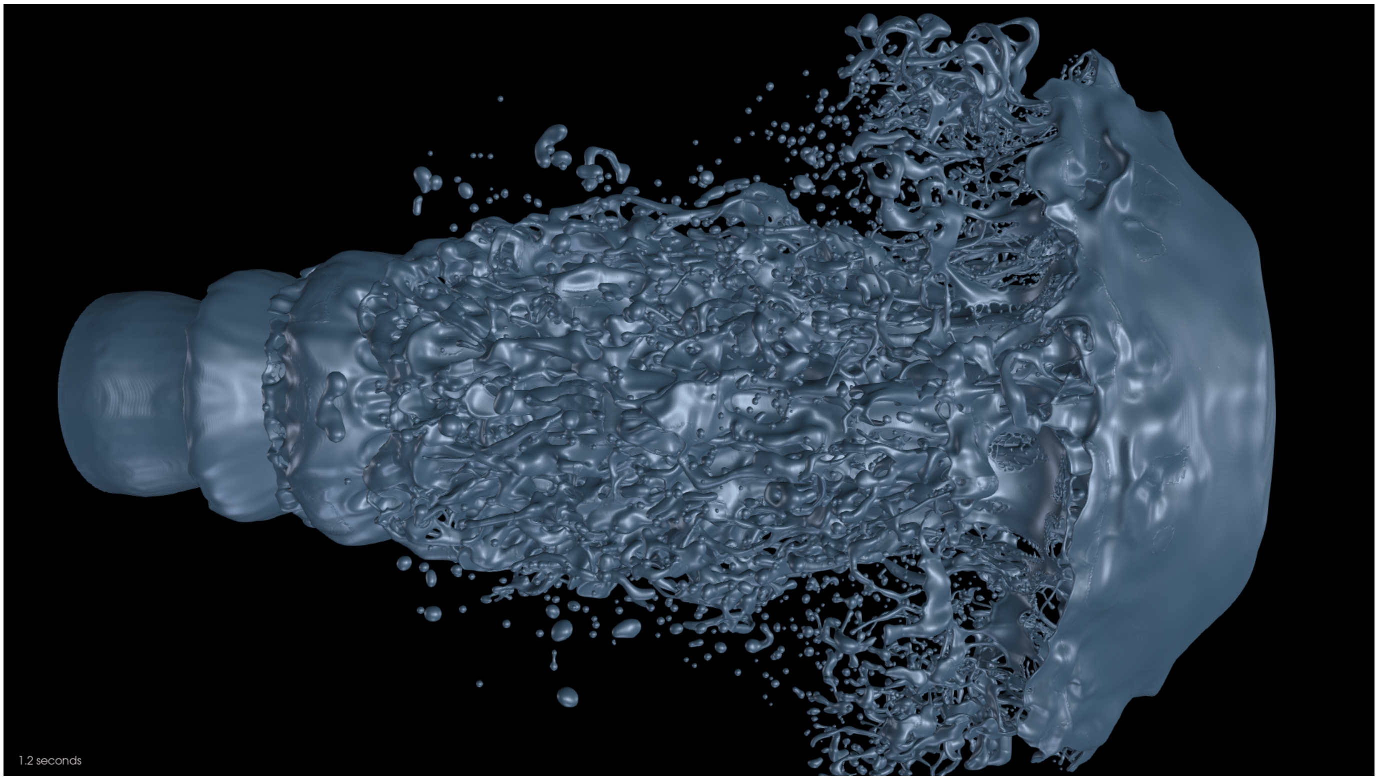

Fig. 3. Visualization of the interface (isosurface of $\phi = 0$): A snapshot of the jet at the latest time in the simulation. Notice the complexities and the fine-scale features captured in the visualization.

data categories. Each part corresponded to a particular parallel process.

Our analysis focused on the phi (ϕ) variable. A value of zero for this variable corresponded to the interface between the liquid and air. Plotting an isosurface of $\phi = 0$ gave the jet's shape and the droplets' position and distribution. The ϕ variable was located in the ch_<timestep>.pvtu data.

B. Data Cleaning

The raw data format was amenable to input into ParaView. However, some modifications to the .pvtu files were necessary for ParaView to be able to read the data. The .pvtu file is an xml formatted file that contains information about the variables and the individual .vtu files that make up the parallel parts of the data.

The information about the scalar variables is coded into the XML tag form as follows:

```
<PPointData>
<PDataArray type="Float64" Name="phi" format="binary"/>
<PDataArray type="Float64" Name="mu" format="binary"/>
</PPointData>
```

The name specification in this file differs from the specification in the vtu parts file. Each .pvtu file required modification to include spaces before and after the variable name to allow ParaView to assemble the data from the parts files. The SED and AWK utilities were used to parse each .pvtu file and correct these variable names. The .pvtu files owned by the science team were first copied to an area of the file system owned by the visualization team. The changes were applied to these files rather than altering the original science team data.

Moving the .pvtu files made further modifications necessary. The .pvtu files contain XML that specifies file names of the constituent parts of the data set. These .vtu file parts were originally in the same directory as the .pvtu XML files, and the file specification therein was relative to the .pvtu file location. It was impractical to copy the large sets of vtu part data files to the same location as the altered .pvtu files for the relative path to remain accurate. The remedy was to replace the relative path of the .vtu file with an absolute path thereby allowing the .pvtu and .vtu files to reside anywhere on the filesystem and still effectively load the dataset. This operation was performed using a combination of SED and AWK. The data for each time step consisted of thousands of individual part files. It was, therefore, impractical to change the XML by hand. An example of the part file XML found in a typical .pvtu file after modification appears below.

```
<Piece Source="/scratch/ch_120000_1_14560.vtu"/>
```

C. Visualization Workflow

The size of the data associated with each time step produced by the simulation grew over time as the fluid surface became more complex and required grid refinement for resolution. The time required to extract and render the gas/fluid interface increased proportionally. The time required to read the `phi` field of the latter time steps was on the order of several minutes.

Parallelism was successfully employed to reduce the time to read the data and extract the isosurface. However, some of the rendering features we hoped to employ either did not work in parallel or left artifacts in the images associated with domain boundaries when parallelism was employed. This made it necessary to render the geometry in serial to obtain the image quality we desired.

As a result, our analysis workflow consisted of parallel extraction of iso-surface geometry followed by storing the geometry on the filesystem. The rendering pass then reads the geometry files. The geometry creation pre-processing phase was done once per time step, while the rendering pass was performed many times during the course of analysis. The geometry files took up much less space than the full fields. `I/O` time in the rendering pass was greatly reduced as a result. Rendering of the reduced geometry in serial posed no problems. Parallelism could still be employed across image frames. Rendering of a single frame for the larger surface geometry took on the order of one minute, including the `I/O` of data.

In the end, once the data raw data was cleaned, we employed a parallel iso-surface extraction and storage pre-processing pass to the entire time series. The resulting geometry data was then visualized interactively with ParaView's gui to set rendering parameters and views for the production of the animation frames.

Several animation tests were designed to allow the science team to examine their data and choose favorite interesting features to highlight. Since this field becomes the geometry's surface, we didn't need volume rendering; instead used a 3D contour (isosurface). A series of animations revealed useful viewpoints and tested surface modeling, color, and lighting in order to most effectively present significant results from the simulation. ParaView's Physically Based Rendering was enabled with diffuse, roughness and metallic parameters interactively adjusted to model soft broad highlights producing surface contrast for picking out the tiny details in the data. Ray tracing was not used because its surface was not as crisp, and shadows tended to add confusion with redundant or skewed patterns. The rendering process could be performed in serial owing to the greatly reduced size of the geometry data as compared to the size of the raw field data. Even 4K animation frames rendered in no more than two minutes.

Adobe After Effects was used to add text, label, and edit the animation sequences into the video.

Acknowledgment

We acknowledge NSF LEAP-HI award number 2053760 and NSF CPS Frontiers award number 1954556, the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute: for Resilient Agriculture, Award No. 2021-67021-35329. We are grateful to TACC for computing resources on Frontera through award number ASC21024. We are also grateful to XSEDE for computing resources on Stampede2, Bridges2, and Expanse through award number CTS110007.

References

[1] M. Gorokhovski and M. Herrmann, "Modeling primary atomization," *Annu. Rev. Fluid Mech.*, vol. 40, pp. 343–366, 2008.

[2] E. Villermaux, "Fragmentation," *Annu. Rev. Fluid Mech.*, vol. 39, pp. 419–446, 2007.

[3] M. A. Khanwale, K. Saurabh, M. Ishii, H. Sundar, J. A. Rossmanith *et al.*, "A projection-based, semi-implicit time-stepping approach for the cahn-hilliard navier-stokes equations on adaptive octree meshes," *arXiv preprint arXiv:2107.05123*, 2021.

[4] G. I. Taylor, "The dynamics of thin sheets of fluid. iii. disintegration of fluid sheets," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 253, no. 1274, pp. 313–321, 1959.

[5] F. E. Culick, "Comments on a ruptured soap film," *Journal of applied physics*, vol. 31, no. 6, pp. 1128–1129, 1960.

[6] F. P. Contò, J. F. Marín, A. Antkowiak, J. R. Castrejón-Pita, and L. Gordillo, "Shape of a recoiling liquid filament," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[7] C. I. Pairetti, S. M. Damián, N. M. Nigro, S. Popinet, and S. Zaleski, "Mesh resolution effects on primary atomization simulations," *Atomization and Sprays*, vol. 30, no. 12, pp. 913–935, 2020.