Contents lists available at ScienceDirect



Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs



# Demonstrating the viability of Lagrangian in situ reduction on supercomputers

# Sudhanshu Sane<sup>a,\*</sup>, Chris R. Johnson<sup>a</sup>, Hank Childs<sup>b</sup>

<sup>a</sup> SCI Institute at University of Utah, United States of America
 <sup>b</sup> University of Oregon, United States of America

# ARTICLE INFO

Keywords: Lagrangian analysis Data reduction Computational fluid dynamics

# ABSTRACT

Performing exploratory analysis and visualization of large-scale time-varying computational science applications is challenging due to inaccuracies that arise from under-resolved data. In recent years, Lagrangian representations of the vector field computed using in situ processing are being increasingly researched and have emerged as a potential solution to enable exploration. However, prior works have offered limited estimates of the encumbrance on the simulation code as they consider "theoretical" in situ environments. Further, the effectiveness of this approach varies based on the nature of the vector field, benefitting from an indepth investigation for each application area. With this study, an extended version of Sane et al. (2021), we contribute an evaluation of Lagrangian analysis viability and efficacy for simulation codes executing at scale on a supercomputer. We investigated previously unexplored cosmology and seismology applications as well as conducted a performance benchmarking study by using a hydrodynamics mini-application targeting exascale computing. To inform encumbrance, we integrated in situ infrastructure with simulation codes, and evaluated Lagrangian in situ reduction in representative homogeneous and heterogeneous HPC environments. To inform post hoc accuracy, we conducted a statistical analysis across a range of spatiotemporal configurations as well as a qualitative evaluation. Additionally, our study contributes cost estimates for distributed-memory post hoc reconstruction. In all, we demonstrate viability for each application — data reduction to less than 1% of the total data via Lagrangian representations, while maintaining accurate reconstruction and requiring under 10% of total execution time in over 90% of our experiments.

# 1. Introduction

High-performance computing resources play a critical role in advancing computational science by enabling modeling of scientific phenomena at high spatiotemporal resolutions. A challenge with regard to studying the output of a simulation is the prohibitively large size of the total data generated. Compromise in the form of storing a subset of the data can impact the extent and accuracy of subsequent post hoc exploratory analysis and visualization. In particular, for accurate time-varying vector field analysis and visualization, access to the full spatiotemporal resolution is required. Since storing the entire simulation output is expensive, scientists resort to temporal subsampling or lossy compression, and often limit analysis to individual time slices. An emerging paradigm to address large data challenges is the use of in situ processing to perform runtime analysis/visualization or data reduction to support exploratory post hoc analysis.

Lagrangian analysis is a powerful tool to study time-varying vector fields and is widely employed for ocean modeling applications [1].

The notion of calculating a Lagrangian representation or *flow map*, i.e., sets of particle trajectories, "online" (in situ) for "offline" (post hoc) exploration was first proposed by Vries et al. [2] for an ocean modeling simulation. Fig. 1 illustrates the approach. More recently, multiple works have advanced research on the topic of data reduction via Lagrangian analysis along axes such as strategies for in situ extraction of reduced Lagrangian representations [3–5], post hoc reconstruction [6,7], feature extraction [8–12], and theoretical error analysis [13–15].

Although enabling accurate time-varying vector field exploration is challenging, prior evaluations of Lagrangian techniques on analytical, SPH, climate and ocean modeling data have provided encouraging results [3,4,6,7,12–18]. Here, the quality of post hoc reconstruction depends on the vector field itself, as well as, configuration specifics such as sampling strategy and frequency of storage. Thus, to gauge viability in practice and to investigate the potential benefits for a broader range of applications, we leverage runtime in situ infrastructure that

https://doi.org/10.1016/j.jocs.2022.101615

Received 15 October 2021; Received in revised form 23 January 2022; Accepted 14 February 2022 Available online 24 February 2022 1877-7503/© 2022 Published by Elsevier B.V.

<sup>\*</sup> Corresponding author. E-mail address: ssane@sci.utah.edu (S. Sane).



**Fig. 1.** Notional space-time visualization of Lagrangian representations for a time-varying 1D flow. The black trajectories are computed in situ and encode the behavior of the vector field between start time  $t_s$  and end time  $t_e$ . In a post hoc setting, a Lagrangian-based advection scheme *L*, i.e., a technique to interpolate the extracted data, is used to calculate the trajectory of a new particle  $p_1$ . The red trajectory is the trajectory reconstructed post hoc and the blue trajectory is the ground truth. The end location of the red trajectory deviates by a margin of error from the ground truth.

enables the straightforward extraction via APIs to study Lagrangian representations for cosmology and seismology simulations as well as a hydrodynamics mini-application targeting exascale computing.

With this work, our unique contribution is an investigation of Lagrangian representations to encode self-gravitating gas dynamics of a cosmology simulation and seismic wave propagation of a seismology simulation. To extend our work in [19], we contribute a performance benchmarking study using a hydrodynamics mini-application to serve as a baseline for future research efforts. Importantly, our study is the first to evaluate the encumbrance placed on a simulation code during in situ reduction. All previous studies ran in "theoretical" in situ environments, meaning data sets were loaded from disk, rather than truly integrated with a simulation code. We inform the in situ encumbrance when considering homogeneous and heterogeneous HPC resource usage as well as provide cost estimates for distributed-memory post hoc reconstruction. For each application, our experiments show that Lagrangian representations offer high data reduction, in many cases requiring less than 1% storage of the complete time-varying vector fields, for a small loss of accuracy. Further, our study shows Lagrangian representations are viable to compute in representative HPC environments, requiring under 10% of total execution time for in situ processing in the majority of configurations tested.

#### 2. Background and related work

#### 2.1. Frames of reference

In fluid dynamics, there are two frames of reference to observe fluid motion: Eulerian and Lagrangian. With the Eulerian frame of reference, the observer is in a fixed position. With the Lagrangian frame of reference, the observer is attached to a fluid parcel and is moving through space and time.

Storage of a flow field in an Eulerian representation is typically done by means of its velocity field. A velocity field v is a time-varying vector field that maps each point  $x \in \mathbb{R}^d$  in space to the velocity of the flow field for a given time  $t \in \mathbb{R}$ 

$$v: \mathbb{R}^{d} \times \mathbb{R} \to \mathbb{R}^{d}, \ x, t \mapsto v(x, t) \tag{1}$$

Storage of a flow field in a Lagrangian representation is done by means of its flow map  $F_{t_0}^t$ . The flow map is comprised of the starting positions of massless particles  $x_0$  at time  $t_0$  and their respective trajectories that are interpolated using the time-varying vector field. Mathematically, a flow map is defined as the mapping

$$F_{t_0}^t(x_0) : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d, \ t \times t_0 \times x_0 \mapsto F_{t_0}^t(x_0) = x(t)$$
(2)

of initial values  $x_0$  to the solutions of the ordinary differential equation

$$\frac{d}{dt}x(t) = v(x(t), t) \tag{3}$$

#### 2.2. Lagrangian analysis

Within the vector field analysis and visualization community, Lagrangian methods have been increasingly researched in the past decade. In this paper, we focus on the use of Lagrangian methods to store timevarying vector fields in situ and enable subsequent post hoc analysis. In sparse temporal settings, Lagrangian representations are expected to perform better than their Eulerian counterparts. The key intuition behind this expectation is that Lagrangian representations capture the behavior of the flow field over an interval of time, as opposed to the state at a single time slice. However, in addition to the frequency of temporal sampling, the nature of the vector field and spatial sampling resolution impacts the quality of reconstruction.

Agranovsky et al. [3] conducted the seminal work to evaluate the efficacy of reduced Lagrangian representations. To maintain domain coverage, the study proposed the use of uniform spatial sampling to extract sets of temporally non-overlapping basis trajectories. Sane et al. [16] studied performance across a range of spatiotemporal configurations when operating using a fixed storage budget. The experiments in these works were conducted in a theoretical in situ setting, i.e., files were loaded from disk. Most recently, Jakob et al. [12] trained a DNN to upsample FTLE visualizations derived from reduced Lagrangian representations. To generate training data, they first computed Lagrangian representations of a 2D flow field using a tightly-coupled integration with an open-source CFD solver on HPC resources and reported computation costs. However, the grid size of  $4 \times 4$  per rank used in the study is not representative of real-world applications. Thus, the current literature lacks in situ encumbrance measurements in representative settings.

Lagrangian representations of a time-varying vector field can be extracted by adopting various strategies. Sane et al. [7] explored computing trajectories of variable duration and placement. Rapp et al. [4] applied their void-and-cluster sampling technique to identify a representative set of scattered samples. Although these strategies improved accuracy, they increased computation costs and are presently limited to single-node settings. To address the scalability challenges of extracting a Lagrangian representation in distributed memory, Sane et al. [5] explored an accuracy-performance tradeoff and demonstrated the use of a communication-free model that stored only trajectories that remain within the rank domain during the interval of computation.

Prior works have presented research pertaining to post hoc reconstruction using Lagrangian-based interpolation schemes. Hlawatsch et al. [20] proposed a hierarchical reconstruction scheme that can improve accuracy, but relies on access to data across multiple time intervals. Chandler et al. [6] proposed a modified k-d tree as a search structure for Lagrangian data extracted from an SPH simulation. Further, Chandler et al. [14] identified correlations between Lagrangianbased interpolation error and divergence in the flow field. Bujack et al. [13] evaluated the use of parameter curves to fit interpolated pathline points to improve the aesthetic of trajectories calculated using Lagrangian data. Hummel et al. [15] provided theoretical upper bounds



Fig. 2. Schematic of the Lagrangian in situ reduction and post hoc exploration workflow.

for error propagation that can occur when calculating trajectories using Lagrangian representations. Lastly, a recent study by Han et al. [21] investigated predicting new particle trajectories during post hoc reconstruction of a 2D flow field by employing a machine learning model trained using basis trajectories as input data.

#### 2.3. Time-varying vector field reduction

Although Eulerian representations have been shown to be susceptible to temporal sparsity [3,16,22–24], temporal subsampling remains the widely used solution to limit data storage. Our study adds to this body of work by using temporal subsampling for comparison. Multiple works have proposed single time step vector field reduction strategies while maintaining an Eulerian representation [25–28]. Although these techniques could be used to reduce and store data more frequently, they do not inherently address the challenge of increasing temporal sparsity.

In a recent large-scale tornadic supercell thunderstorm study [29], Leigh Orf modified the I/O code to use a hierarchical data format and lossy floating-point compression via ZFP [30]. ZFP provides dynamic accuracy control by allowing the user to specify a maximum amount of deviation. Orf stated that although ZFP is effective for scalar fields that do not require differentiation during post hoc analysis, only a very small value of deviation can be chosen for each component of velocity to maintain accurate time-varying vector field reconstruction. Thus, ZFP allowed a limited amount of compression to vector field data without introducing significant uncertainty to post hoc analysis. The technique provided an average reduction of 30% of total uncompressed vector field data, with regions of high gradient resulting in less compression. Overall, Orf referred to the use of lossy compression as unfortunate but necessary.

### 3. Lagrangian in situ reduction and post hoc exploration

This section focuses on the methodology we consider for extraction of a Lagrangian representation and its use during post hoc analysis. Fig. 2 shows a high-level description of the Lagrangian in situ reduction post hoc exploration (L-ISR-PHE) workflow. For our study, we focused on the current best practices in this space accounting for the scale of experiments. To describe the instantiation we consider, the remainder of this section is divided based on the two phases: in situ reduction and post hoc exploration.

# 3.1. In situ reduction

**Computation of a Lagrangian Representation** When considering a Lagrangian frame of reference to store time-varying vector field data, the flow map (as defined in (2)) is represented as sets of particle trajectories calculated in the time interval  $[t_0, t] \subset \mathbb{R}$ . The stored information, encoded in the form of known particle trajectories (i.e., a Lagrangian representation), encodes the behavior of the time-varying vector field over an interval of time.

In contrast, when considering the traditional Eulerian frame of reference in a practical setting, a flow field at a specific time/cycle is defined as vector data on a fixed, discrete mesh. Time-varying flow is represented as a collection of such data over a series of times/cycles. Considering the vector data is stored without any reduction or transformation, there is no additional execution time required by this approach.

As previously stated in Section 2.2, a Lagrangian representation of the simulation time-varying vector field can be computed by adopting various spatial and temporal flow map sampling strategies. The encumbrance placed on the simulation code from in situ processing can be quantified by measuring the total execution time and runtime memory usage of the in situ reduction routine. These metrics vary based on the sampling strategy implemented as well as the particle advection workload and underlying hardware. With our study, we were interested in capturing time-varying vector field behavior across the entire domain for each application. Thus, for our in situ data reduction strategy, we prioritized domain coverage. Similarly to Agranovsky et al. [3], we used uniform spatial sampling and a predetermined temporal interval to store/reset particles. Thus, we computed sets of temporally nonoverlapping basis trajectories over the duration of the simulation. Each set of basis trajectories stored flow field information for a specific interval of time. Our particle termination followed the local Lagrangian flow map model from Sane et al. [5], where particles are terminated once they reach the end of the interval or exit the block. Our implementation had two main knobs that control the total data storage and quality of reconstruction: number of basis trajectories N, i.e., spatial sampling resolution, and frequency of storing information to disk I, i.e., storage interval. The effect of these settings on reconstruction quality varies depending on the underlying vector field.

The total execution time E of computing a Lagrangian representation is a product of the average execution time **Step** for advancing a set of basis trajectory particles one step and the total number of cycles C of the simulation. The execution time for each **Step** depends on several factors including the number of particles N, choice of ODE solver, vector field interpolation and cell location methods, grid size, grid type, and underlying hardware used for parallelism. This cost can be measured as follows

$$Step = \frac{L}{C}$$
(4)

In our study, we report **Step** as the average measured execution time required for advancing various particle advection workloads forward a step in an experiment setting.

With respect to memory usage, the total runtime memory **InSituMem** required by the in situ reduction routine can be calculated as

$$InSituMem = P * N$$
(5)

where P is the storage cost per basis trajectory, and N is the total number of particles. The total data storage **DS** can be approximated as

$$\mathbf{DS} = (\mathbf{InSituMem} + c) \times \frac{C}{I}$$
(6)

F

where c is an overhead storage cost (coordinate system and file format), C is the total number of simulation cycles, and I is the storage interval. Section 4.5 contains further details regarding the metrics we used for evaluation of in situ encumbrance.

**Implementation** Based on the in situ system classification in [31], the system we implement for Lagrangian in situ reduction is classified as one with a dedicated API integration, on-node proximity, direct access, a time division of execution, automatic operation, and a derived output type.

Each application we considered used MPI and partitioned space amongst ranks, with each rank owning one portion of the vector field. Our in situ routines followed this pattern, with an instance of our Lagrangian analysis routine executing for each rank, accessing its portion of the vector field. On each individual rank, we leveraged the available shared-memory parallelism for particle advection.

We used the Ascent [32] in situ infrastructure and VTK-m [33] library to implement L-ISR. The Ascent API can be used to perform tightly-coupled integration with an application code and access various in situ analytics capabilities. The VTK-m Lagrangian filter instance executing on each rank operated independently and maintained its own list of particles. We used the existing particle advection infrastructure available in VTK-m [34]. RK4 particle advection is implemented using VTK-m worklets (kernels) that offer performance portability by utilizing the underlying hardware accelerators. In our implementation, each Lagrangian filter stored the displacement of each particle (three double), as well as its validity (one Boolean), i.e., whether the particle remained within the domain during the interval of calculation. Overall, computing a Lagrangian representation increased the runtime memory cost on the simulation by approximately by four one-dimensional simulation "fields". Simulations often have tens to hundreds of fields defined on the simulation grid, and thus, this cost would likely be acceptable for most simulations.

The simulation invoked Ascent after every cycle it advanced to compute the Lagrangian representation using the full temporal resolution. Ascent accessed the simulation vector field data and consequently invoked the Lagrangian filter. The Lagrangian filter used the vector field to advance particles, and triggered the storage of trajectories at the end of an interval. Execution control is returned to the simulation and the process repeats for every cycle of the simulation.

#### 3.2. Post hoc exploration

For post hoc analysis, new particle trajectories are computed to explore the time-varying vector field. To construct new particle trajectories, we first identified which basis trajectories to follow and then performed interpolation. Based on the study of accuracy of various Lagrangian-based advection schemes in [35], our study employed a Delaunay triangulation to identify the neighborhood of valid basis trajectories and second-order barycentric coordinates for interpolation. We used the CGAL [36] library to construct and search the Delaunay triangulation. After constructing new pathlines or deriving new scalar fields from the basis trajectories, we used VisIt [37] to generate visualizations.

#### 4. Study overview

This section provides an overview of our study. It is organized as follows: runtime environment (4.1), shared-memory parallel particle advection (4.2), simulation codes (4.3), experiments (4.4), and evaluation metrics (4.5).

#### 4.1. Runtime environment

Our study used the Summit supercomputer at ORNL. A Summit compute node has two IBM Power9 CPUs, each with 21 cores running at 3.8 GHz and 512 GBytes of DDR4 memory. Nodes on Summit also have enhanced on-chip acceleration with each CPU connected via NVLink to 3 GPUs, for a total of 6 GPUs per node. Each GPU is an NVIDIA Tesla V100 with 5120 CUDA cores, 6.1 TeraFLOPS of double precision performance, and 16 GBytes of HBM2 memory. Lastly, it has a Mellanox EDR 100G InfiniBand, non-blocking fat tree as its interconnect topology.

#### 4.2. Shared-memory parallel particle advection

In our experiments, we leverage the available shared-memory parallelism for particle advection on each rank to compute the Lagrangian representation. VTK-m as a platform portable library provides particle advection capabilities that can execute in parallel on the GPU via CUDA and CPU via OpenMP. In our setup, each rank operates over a specific region of the simulation domain, maintains a set of particles, and has an independent instance of a VTK-m Lagrangian filter executing. VTKm handles launching of the particle advection kernels in parallel and performs other particle management operations within the Lagrangian filter.

#### 4.3. Simulation codes

**Nyx:** In this cosmological application [38], baryonic matter is evolved by solving the equations of self-gravitating gas dynamics. We derived the velocity field using the fields of momentum and density of the baryonic gas. The simulation involves particles gravitating toward high-density regions to form multiple clusters across the domain. The distribution of high-density clusters and their formation are of interest to scientists. To study the distribution, scientists currently perform statistical analysis of gas particle density at a single time slice. We investigated the potential of reduced Lagrangian representation to accurately visualize the particle evolution and the distribution of high-density clusters using pathlines. The Nyx simulation we built executed as a single rank using two CPUs on a single Summit compute node. The Lagrangian analysis routines execute on the CPUs as well.

**SW4:** In this seismology application [39], seismic wave propagation is studied using a fourth-order method. The application simulates waves radiating from the epicenter through viscoelastic media. We used the 3D time-varying displacement vector defined at each grid point as input. We investigated how accurately we can derive and visualize the field encoding displacement over time in two regions: at the epicenter and away from the epicenter. The SW4 simulation we built executed using six ranks per compute node on Summit. Each rank was allocated a GPU for execution of the simulation code as well as the Lagrangian analysis routines.

**Cloverleaf3D:** In this hydrodynamics mini-application [40], compressible Euler equations are solved on a Cartesian grid using an explicit second-order method. Cloverleaf3D has been developed for studies to evaluate emerging architectures [41,42] as well as analysis workflows targeting exascale computing [43–47]. The significantly simplified physics of the simulation for computer science experimentation serves as a useful benchmark application. As hydrodynamics is a common base to physics models of interest, if a methodology performs poorly for Cloverleaf3D, it will be considerably harder for other physics models. The Cloverleaf3D mini-application we built executed using six ranks on two CPUs on a single Summit compute node. Here, each rank was allocated a GPU to execute the Lagrangian analysis routines.

### 4.4. Experiments

For each application in this study, we organized our experiments to inform in situ encumbrance and post hoc accuracy. We considered four evaluation criteria (EC). To inform in situ encumbrance, we measured the execution time (EC1) and runtime memory usage (EC2) for in situ processing. To inform post hoc accuracy, we measured the size of data artifacts (EC3) and the reconstruction quality of time-varying vector field data (EC4). Lastly, using Cloverleaf3D for a cost estimate, we measured the execution time (EC5) of distributed-memory post hoc pathline reconstruction. Next, we identified four factors that when varied produce the workloads we want to evaluate for our study:

- Number of particles: the spatial sampling resolution denoted using 1:X, where X is the reduction factor. For example, a 1:8 configuration states that one basis particle is used for every 8 grid points (≈12.5% of the original data size).
- Storage interval: the number of cycles between saves and denoted by I.
- · Grid size: the spatial resolution of the mesh.
- **Concurrency:** the scale of the execution and underlying parallelization hardware.

Rather than consider a complete cross-product of options for every workload factor, we sampled the space of possible options. Our goal was to provide coverage and allow us to see the impact of certain workload factors, all while staying within our compute budget. For Nyx, we ran 18 experiments, with 6 informing in situ encumbrance (varying 1:X, grid size) and 12 informing post hoc accuracy (varying 1:X, I). For SW4, we ran 11 experiments, with 7 informing in situ encumbrance (varying 1:X, grid size, concurrency) and 4 informing post hoc accuracy (varying 1:X). For Cloverleaf3D, we ran 18 experiments, with 9 informing in situ encumbrance (varying 1:X, I). The specific options selected are presented along with the results in Section 5.

## 4.5. Evaluation metrics

We selected our evaluation metrics based on the evaluation criteria listed in Section 4.4.

For EC1, we measured the average cost of invoking the Lagrangian VTK-m filter through Ascent every cycle, **Step**, in seconds. Additionally, we presented the percentage of simulation time spent on data analysis and visualization, or **DAV%**. We used  $Sim_{cycle}$  to denote the average time required for a single simulation cycle in seconds.

For EC2, we measured **InSituMem**, the runtime memory cost incurred by every compute node to maintain the state (current position) of particles at runtime in Bytes.

For EC3, we measured the total data storage (**DS**) required on the file system and report it in Bytes stored. In addition to I/O being infrequently performed, we observed that for the scale of study we conducted, Summit provided fast write times. In comparison to performing in situ processing every cycle, we found the I/O write cost to be negligible.

For EC4, we considered both a statistical and qualitative analysis. For Nyx and Cloverleaf3D, we derived pathlines from the basis trajectories and measured the reconstruction error by calculating the average Euclidean distance of interpolated points from the ground truth (precomputed using the complete simulation data) for each trajectory. We visualized the distribution of pathline reconstruction error for every configuration using violin plots, and for a subset of Nyx configurations, the pathline clustering directly. For SW4, we derived a field encoding magnitude of displacement over time from the basis trajectories. In this case, we visualized and compared the derived field to the ground truth time-varying displacement field using violin plots and isosurfaces.

For EC5, we measured the execution time for the triangulation, interpolation, and communication required during distributed-memory post hoc reconstruction.

Table 1

In situ encumbrance evaluation and experiment configurations for the Nyx simulation executing on CPUs.

Nodes	Ranks	Dimensions	<b>Sim</b> <sub>cycle</sub>	Particles	InSituMem	Step	DAV%
1	1	65 <sup>3</sup>	10.9 s	9k 32k 274k	0.2 MB 0.8 MB 6.8 MB	0.0025 s 0.0033 s 0.0122 s	0.023% 0.030% 0.112%
		129 <sup>3</sup>	88.3 s	78k 262k 2.1M	1.9 MB 6.5 MB 53.6 MB	0.0044 s 0.0101 s 0.0596 s	0.005% 0.011% 0.067%

### 5. Results

Our results are organized as follows: Sections 5.1, 5.2, and 5.3 present findings from our study investigating the viability and efficacy of reduced Lagrangian representations for the Nyx cosmology, SW4 seismology, and Cloverleaf3D hydrodynamics applications, respectively. Tables 1, 2, and 3 provide information pertaining to in situ encumbrance experiments, such as concurrency information, spatial dimensions,  $Sim_{cycle}$ , number of particles per compute node, InSituMem per compute node, Step, and DAV%, for each application. Fig. 3 shows the execution time per cycle for all the in situ encumbrance experiments from each application. Figs. 4, 5, 6, 7, and 8 show the results of our post hoc accuracy evaluation, and Table 4 provides cost estimates of reconstruction. For each application, the figures are annotated with configuration specifics such as the DS, 1:X, and I. Further, Lagrangian and Eulerian tests are distinguished explicitly in the captions or are labeled LT and ET, respectively, where T is the test number.

# 5.1. Nyx cosmology simulation

In Situ Encumbrance Using all the cores of two CPUs on a single compute node, we used OpenMP to parallelize the Nyx simulation and Lagrangian VTK-m filter. We tested two options for grid size -  $69^3$  and  $129^3$  - on a single rank, and three particle advection workloads (1:1, 1:8, 1:27) each. In a single compute node hour, the simulation performed approximately 300 and 38 cycles when using  $69^3$  and  $129^3$  grid sizes, respectively. An 8X increase in grid size resulted in a proportional increase in Sim<sub>cycle</sub> but only a small increase in particle advection costs for the same number of particles. In practice, we would expect a single rank to operate on between  $32^3$  to  $256^3$  grid points, and thus our workloads provided a representative estimate of DAV%.

An encouraging finding was the low in situ encumbrance when performing L-ISR on the CPUs. Depending on the setup of various simulations and the form of integration for in situ processing, future work can consider offloading L-ISR computation to CPUs. Overall, considering the longer  $\mathbf{Sim}_{cycle}$  times for the Nyx simulation, and parallel computation coupled with low memory latency when using CPUs, the highest in situ encumbrance to extract a Lagrangian representation was 0.1% of the simulation time or under 0.06s to compute 2.1M basis trajectories per cycle.

**Post Hoc Accuracy** To evaluate the usefulness of Lagrangian representations to encode time-varying self-gravitating gas dynamics, we considered a  $69^3$  grid over 400 cycles, three options for data reduction (1:1, 1:8, 1:27) and four options for I (25, 50, 100, 200). We constructed pathlines for 50,000 randomly placed particles during post hoc analysis. We visualize the distribution of reconstruction error for all tests in Fig. 4.

The self-gravitating gas dynamics of this simulation produce a vector field that captures the transport of randomly distributed particles to multiple high-density clusters. Particles travel with increasing velocity as clusters increase in density. For this data, we found that Eulerian temporal subsampling performs better for small values of **I**. This result can be expected given reconstruction using an Eulerian



Fig. 3. Lagrangian in situ reduction cost per cycle for all in situ encumbrance experiments. The SW4 application executes with six ranks (each allocated one GPU) sharing memory on every node. The Nyx application executes on a single rank using all the cores of two CPUs on a single node. The Cloverleaf3D application itself executes with six ranks on the CPU, while the corresponding in situ processing executes on the GPU allocated to each rank. The legend includes concurrency and number of simulation grid points in parenthesis and both axes use logarithmic scales.



**Fig. 4.** Accuracy results for the Nyx experiments. Each violin plot shows the distribution of the particle reconstruction error for a specific configuration, and the horizontal blue dashed line in the chart represents an error equivalent to a single grid cell side. The error axis uses a logarithmic scale. Lagrangian and Eulerian tests are labeled LT and ET, respectively, where T is the test number. Whereas Eulerian configurations contain greater uncertainty as the value of storage interval I increases, the Lagrangian representations offer the opportunity for improvements in accuracy. Additionally, we find high reconstruction accuracy relies on a high spatial sampling resolution as well.



Fig. 5. Pathline visualization of baryonic particles evolution in self-gravitating gas dynamics of Nyx simulation. Using 10,000 randomly seeded particles, we visualize pathlines over 300 cycles. To focus on regions where particles cluster to form dense regions, we set opacity of the pathline to be directly proportional to time. Thus, we are able to focus on clustering as well as provide context of transport toward these regions. Lagrangian representations are able to reconstruct the ground truth trajectories and capture clustering accurately when high spatial sampling is used (1:1, 1:8). However, when using a 1:27 data reduction factor, some clusters are visualized less clearly.

#### Table 2

In situ encumbrance evaluation and experiment configurations for the SW4 simulation executing on GPUs. Particles and **InSituMem** are per compute node.

Nodes	Ranks	Dimensions	$\mathbf{Sim}_{cycle}$	Particles	InSituMem	Step	DAV%
		$251\times251\times70$	0.35 s	555k	13.89 MB	0.0412 s	11.67%
1	6	$335 \times 335 \times 93$	2.02 s	1.3M	33.16 MB	0.2125 s	10.48%
		$501\times501\times139$	7.58 s	4.4M	111.13 MB	0.3309 s	4.365%
64	384	1001 × 1001 × 276	1.6 s	66k	1.6 MB	0.0194 s	1.201%
			1.5 s	146k	3.6 MB	0.0295 s	1.944%
			1.3 s	540k	13.5 MB	0.0798 s	6.175%
		1335 × 1335 × 368	2.9 s	1.2M	31.9 MB	0.2095 s	7.074%

representation and fourth-order Runge Kutta interpolation remain more accurate than second-order barycentric coordinates interpolation employed to interpolate Lagrangian representations [13,15]. However, as the value of I increases, the distribution of error for the Lagrangian tests indicates that a larger percentage of samples are reconstructed more accurately. In particular, this is true when a high spatial sampling resolution is used. Thus, particle evolution in this cosmology simulation can be tracked more accurately when a dense set of basis trajectories integrated for a long duration is interpolated. In contrast, Eulerian representations become less effective at reconstructing the vector field due to increased numerical approximation.

We used pathlines with manually set transfer functions to visualize the evolution and clustering of particles in regions of high density. The total size of the simulation vector field data used to compute the ground truth is 5.3 GB. We visualized a random subset of 10,000 pathlines in Fig. 5 for configurations with I set to 25. The Lagrangian representations demonstrate the ability to closely reconstruct regions where dense clusters are formed while requiring a fraction of the total simulation data size. For example, the 1:8 Lagrangian configuration enables the visualization of transport to dense clusters while requiring only 27MB, i.e., a 200X data reduction of the uncompressed vector field.

#### 5.2. SW4 seismology simulation

**In Situ Encumbrance** For the SW4 simulation, we considered five grid sizes at varying concurrencies. In each case, we used all six GPUs available on a compute node to execute the simulation and L-ISR. For all L-ISR workloads tested, the execution time required per cycle remained under 0.5 s on average, and the maximum in situ memory required by a node was 112 MB to compute the trajectories for 4.4M particles. The cost for performing L-ISR was most dependent on the number of particles and only slightly impacted by increasing grid sizes. Referencing Fig. 3, although the SW4 experiments used six GPUs, we found execution time to be slower than for the Nyx experiments due to the use of shared memory by multiple ranks (each has its own data block) and the high cost of launching kernels on the GPU for limited amounts of computation (each basis particle advances by only a single step/cycle each invocation).

**Post Hoc Accuracy** We studied the reconstruction of the time-varying displacement vector field encoding wave propagation by considering four options for data reduction (1:1, 1:8, 1:27, 1:64) and one option for I (250). The ground truth was computed using data defined on a regular mesh containing 4.5M grid points over 2000 simulation cycles and required 245 GB. The displacement was highest near the epicenter and diminished as waves propagated further away. For each simulation run, we measured the displacement of 200,000 samples reconstructed near the epicenter (Fig. 6a) and 90,000 samples reconstructed in six regions away from the epicenter (Fig. 6b). Here, we directly compared against the distribution of ground truth (GT) displacement. In both cases, Lagrangian representations offered significant data reduction while maintaining high accuracy. We found that as the number of

Table 3

In situ encumbrance evaluation and experiment configurations for the Cloverleaf3D simulation executing on CPUs and Lagrangian analysis routines performed on GPUs.

Nodes	Ranks	Dims	Interval	Sim <sub>cycle</sub>	Particles	InSituMem	Step	DAV%
16	96	586 <sup>3</sup>	20	4.73 s	1.5M	40.2 MB	0.4475 s	9.408
			40	4.08 s			0.3221 s	7.894
			60	4.39 s			0.3838 s	8.742
			20	4.50 s	474k	12 MB	0.1882 s	4.182
			40	4.14 s			0.1628 s	3.932
			60	4.33 s			0.1498 s	3.459
			20	4.19 s	186k	4.2 MB	0.0925 s	2.207
			40	4.11 s			0.1043 s	2.537
			60	3.87 s			0.0830 s	2.144

basis trajectories extracted decreases, the displacement for some samples near the epicenter can be underestimated. In contrast, using a temporally subsampled Eulerian representation (E01) results in significant overestimation of displacement. This result can be expected since temporal subsampling fails to capture the transient nature of wave propagation, whereas Lagrangian representations encoding behavior over an interval of time remain accurate. Compared to Fig. 6a, the ground truth in Fig. 6b has smaller displacement and a multimodal distribution, which is the result of samples collected from six regions of the domain away from the epicenter.

Fig. 7 visualizes field encoding displacement over time near the epicenter using multiple semiopaque isosurfaces. Although regions of highest displacement can be underestimated as the data reduction factor increases, the overall structure is well preserved using highly compressed Lagrangian representations. In all cases, Lagrangian representations required less than 1% of the storage of the complete vector data.

#### 5.3. Cloverleaf3D hydrodynamics mini-application

In Situ Encumbrance For Cloverleaf3D, we considered a single grid size and concurrency. 600 cycles of the Cloverleaf3D simulation execute across six ranks per node using the CPUs, whereas the Lagrangian analysis routines execute by launching kernels every cycle on the GPUs. For our specific grid size and domain decomposition, each MPI rank operated over 2M grid points, and the  $Sim_{cycle}$  was usually between 4–5 s. We considered three particle advection workloads (1:8, 1:27, 1:64) and three intervals for each option of number of particles. Referencing Fig. 3, using GPUs for L-ISR is most expensive when data movement is required for the simulation data. Further, by executing the same workload multiple times, we capture the variation in the costs within a workload. The variation in the **Step** cost is greater when the workload is larger due to increased memory allocation and memory transfer costs each step. Going forward, we expect the performance for heterogeneous compute resources can be optimized by leveraging unified memory.

Overall, in our experiments, increasing the number of particles by 8X results in the **Step** increasing by 3X–4X. The cost of a single **Step** to calculate the Lagrangian representation for Cloverleaf3D was as low as 0.08 s and in all cases, below half a second. With  $Sim_{cycle}$  relatively stable, the **DAV%** increases from 2% to 10% as the particle advection workload is increased.

**Post Hoc Accuracy and Reconstruction** To evaluate the efficacy of Lagrangian representations to encode the time-varying hydrodynamics of the mini-application, we considered a  $586^3$  grid over 600 cycles, three options for data reduction (1:8, 1:27, 1:64) and three options for I (20, 40, 60). A The total size of the simulation vector field data used to compute the ground truth is 2TB. We constructed pathlines for 100,000 randomly placed particles during post hoc analysis and visualize the distribution of reconstruction error for all tests in Fig. 8. Referencing Fig. 8, Lagrangian tests result in accurate reconstruction of the pathlines with particles remaining within the same cell as the ground truth



(a) High displacement near the epicenter.

(b) Low displacement away from the epicenter.

**Fig. 6.** Violin plots of the distribution of particle displacement for the ground truth (GT), one Eulerian configuration and four Lagrangian configurations. Lagrangian and Eulerian tests are labeled LT and ET, respectively, where T is the test number. The Eulerian configuration, with access to a limited number of time slices, overestimates the displacement. The Lagrangian representation captures displacement in both settings, in regions near and away from the epicenter, accurately.



Fig. 7. Visualization of the displacement field derived from reduced Lagrangian representations near the epicenter using multiple isosurfaces. The ground truth is computed using 2000 cycles of the seismic wave propagation simulation. Although at higher data reduction factors regions of high displacement are underestimated, Lagrangian representations are capable of accurately reconstructing the overall feature structure.



**Fig. 8.** Accuracy results for the Cloverleaf3D experiments. Each violin plot shows the distribution of the particle reconstruction error for a specific configuration, and the horizontal blue dashed line in the chart represents an error equivalent to a single grid cell side. The error axis uses a logarithmic scale. Lagrangian and Eulerian tests are labeled LT and ET, respectively, where T is the test number. Whereas the Eulerian configurations contain greater uncertainty as the value of storage interval I increases, the reduced Lagrangian representations continue to offer strong accuracy-storage propositions.

in most cases. Although this simple hydrodynamics model demonstrates the shortcomings of the Eulerian technique as the interval increases, we observed the Lagrangian technique benefits minimally from an increase in the number of particles sampling the domain. We believe this is due to Cloverleaf3D being a mini-application, where increasing the spatial resolution does not increase the complexity of the physics, i.e., no new features are introduced as they would be in a real-world simulation. That being said, even if the Eulerian techniques used multiresolution to achieve reduced storage, it would be less accurate that Lagrangian analysis given the full spatial resolution is less accurate.

Finally, we provide an estimate of the cost to construct new trajectories in a distributed-memory setting (16 nodes, 96 MPI ranks) in Table 4. For every interval, each rank constructs a Delaunay triangulation of its own and adjacent process samples. We measured the time to perform a Delaunay triangulation on each rank using CGAL in serial

#### Table 4

Distributed-memory post hoc interpolation cost for 100,000 particles across a single interval of the Cloverleaf3D extracted data using 16 compute nodes and 96 MPI ranks on Summit. Values averaged over all reconstruction runs.

	v			
Samples /Rank	Delaunay Serial	Delaunay Parallel	Interpolation	Communication
7.2M	178 s	24.6 s	0.00246 s	
2.1M	53 s	7.05 s	0.00141 s	0.00125 s
887k	21 s	2.89 s	0.00093 s	

and in parallel via Intel TBB (8 threads) [48]. Even accounting for the speedup from using parallelism, construction of the search structure to identify particle neighborhoods for interpolation is the dominant cost during post hoc analysis. The total cost to reconstruct a set of

pathlines depends on the number of intervals and total number of cycles as well. Considering the unstructured forms Lagrangian representations might take, future efforts could study the use ray tracing hardware [49] as well as machine learning approaches to accelerate exploration of large-scale three-dimensional Lagrangian flow maps.

#### 6. Conclusion

Accurate exploratory analysis and visualization of time-varying vector fields produced on supercomputers is challenging. On the one hand, it can be performed accurately if the entire spatiotemporal resolution is available. On the other hand, storing all the data to disk for exploratory post hoc analysis is expensive. In this context, Lagrangian representations computed using the full spatiotemporal resolution via in situ processing demonstrate the potential to enable accurate exploratory time-varying vector field analysis for reduced data storage costs.

For wider adoption and consideration of Lagrangian representations, an important step is characterization of efficacy across a broad range of real-world applications and evaluation of viability at scale on a supercomputer. In this paper, we investigated in situ reduction via Lagrangian representations for vector fields from Nyx cosmology, SW4 seismology, and Cloverleaf3D hydrodynamics applications. For the Nyx cosmology simulation, we found that Lagrangian representations are sensitive to both the spatial and temporal sampling rate, notably providing higher reconstruction accuracy when basis trajectories are computed using a high spatial and low temporal resolution. For the SW4 seismology simulation, we found Lagrangian representations are well suited to capture the transient seismic waves and offer high data reduction options for a small loss of accuracy. Similarly, for Cloverleaf3D, accuracy was maintained under high data reduction, and we provided cost estimates to perform distributed-memory post hoc reconstruction on unstructured Lagrangian representations. For each application, irrespective of homogeneous and heterogeneous resource usage, the percentage of execution time spent on computing the Lagrangian representation in situ, i.e., the encumbrance, was under 10% in most cases. Overall, we believe the findings of this study demonstrates the two computational science applications considered, benefit from Lagrangian representations for time-varying vector field exploration. This finding also provides confidence that more computational areas can benefit from and viably perform in situ vector field via Lagrangian representations on supercomputers, and we encourage future work in this direction.

#### CRediT authorship contribution statement

Sudhanshu Sane: Designing and conducting the experiments, Preparing the manuscript. Chris R. Johnson: Advising, Provided valuable input for analysis, Visualization of data sets. Hank Childs: Advising, Preparing the manuscript.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

# Acknowledgments

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 of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under grant number DE-AC05-000R22725. The authors acknowledge current research support provided in part by the Intel Graphics and Visualization Institutes of XeLLENCE, the National Institutes of Health under grant number P41 GM103545 and R24 GM136986 and the Department of Energy under grant number DE-FE0031880.

# References

- E. van Sebille, et al., Lagrangian ocean analysis: Fundamentals and practices, Ocean Model. 121 (2018) 49–75.
- [2] P. Vries, K. Döös, Calculating lagrangian trajectories using time-dependent velocity fields, J. Atmos. Ocean. Technol. 18 (6) (2001) 1092–1101.
- [3] A. Agranovsky, D. Camp, C. Garth, E.W. Bethel, K.I. Joy, H. Childs, Improved post hoc flow analysis via lagrangian representations, in: 4th IEEE Symposium on Large Data Analysis and Visualization, LDAV, 2014, pp. 67–75.
- [4] T. Rapp, C. Peters, C. Dachsbacher, Void-and-cluster sampling of large scattered data and trajectories, IEEE Trans. Vis. Comput. Graphics 26 (1) (2019) 780–789.
- [5] S. Sane, A. Yenpure, R. Bujack, M. Larsen, K. Moreland, C. Garth, C.R. Johnson, H. Childs, Scalable in situ computation of Lagrangian representations via local flow maps, in: Eurographics Symposium on Parallel Graphics and Visualization, The Eurographics Association, 2021.
- [6] J. Chandler, H. Obermaier, K.I. Joy, Interpolation-based pathline tracing in particle-based flow visualization, IEEE Trans. Vis. Comput. Graphics 21 (1) (2015) 68–80.
- [7] S. Sane, H. Childs, R. Bujack, An interpolation scheme for VDVP lagrangian basis flows, in: Eurographics Symposium on Parallel Graphics and Visualization, The Eurographics Association, 2019.
- [8] G. Froyland, K. Padberg-Gehle, A rough-and-ready cluster-based approach for extracting finite-time coherent sets from sparse and incomplete trajectory data, Chaos.
- [9] K.L. Schlueter-Kuck, J.O. Dabiri, Coherent structure colouring: identification of coherent structures from sparse data using graph theory, J. Fluid Mech..
- [10] A. Hadjighasem, M. Farazmand, D. Blazevski, G. Froyland, G. Haller, A critical comparison of Lagrangian methods for coherent structure detection, Chaos.
- [11] G. Froyland, O. Junge, Robust FEM-based extraction of finite-time coherent sets using scattered, sparse, and incomplete trajectories, SIAM J. Appl. Dyn. Syst..
- [12] J. Jakob, M. Gross, T. Günther, A fluid flow data set for machine learning and its application to neural flow map interpolation, IEEE Trans. Vis. Comput. Graphics (Proc. IEEE Scientific Visualization).
- [13] R. Bujack, K.I. Joy, Lagrangian representations of flow fields with parameter curves, in: IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2015, pp. 41–48.
- [14] J. Chandler, R. Bujack, K.I. Joy, Analysis of error in interpolation-based pathline tracing, in: Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers, Eurographics Association, 2016, pp. 1–5.
- [15] M. Hummel, R. Bujack, K.I. Joy, C. Garth, Error estimates for lagrangian flow field representations, in: Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers, Eurographics Association, 2016, pp. 7–11.
- [16] S. Sane, R. Bujack, H. Childs, Revisiting the evaluation of in situ lagrangian analysis, in: Eurographics Symposium on Parallel Graphics and Visualization, The Eurographics Association, 2018.
- [17] P. Nardini, M. Böttinger, G. Scheuermann, M. Schmidt, Visual study of the benguela upwelling system using pathline predicates, in: Workshop on Visualisation in Environmental Sciences (EnvirVis), The Eurographics Association, 2017.
- [18] L. Siegfried, M. Schmidt, V. Mohrholz, H. Pogrzeba, P. Nardini, M. Böttinger, G. Scheuermann, The tropical-subtropical coupling in the southeast atlantic from the perspective of the northern benguela upwelling system, PLoS One 14 (1).
- [19] S. Sane, C.R. Johnson, H. Childs, Investigating in situ reduction via lagrangian representations for cosmology and seismology applications, in: Computational Science – ICCS 2021, Springer International Publishing, Cham, 2021, pp. 436–450.
- [20] M. Hlawatsch, F. Sadlo, D. Weiskopf, Hierarchical line integration, IEEE Trans. Vis. Comput. Graphics 17 (8) (2011) 1148–1163.
- [21] M. Han, S. Sane, C.R. Johnson, Exploratory lagrangian-based particle tracing using deep learning, arXiv preprint arXiv:2110.08338.
- [22] M. Valdivieso Da Costa, B. Blanke, Lagrangian methods for flow climatologies and trajectory error assessment, Ocean Model. 6 (3) (2004) 335–358.
- [23] X. Qin, E. van Sebille, A. Sen Gupta, Quantification of errors induced by temporal resolution on lagrangian particles in an eddy-resolving model, Ocean Model. 76 (2014) 20–30.
- [24] M.P. Rockwood, T. Loiselle, M.A. Green, Practical concerns of implementing a finite-time lyapunov exponent analysis with under-resolved data, Exp. Fluids 60 (4) (2019) 1–16.
- [25] S.K. Lodha, J.C. Renteria, K.M. Roskin, Topology preserving compression of 2d vector fields, in: Proceedings Visualization 2000. VIS 2000, IEEE, 2000, pp. 343–350.
- [26] S.K. Lodha, N.M. Faaland, J.C. Renteria, Topology preserving top-down compression of 2d vector fields using bintree and triangular quadtrees, IEEE Trans. Vis. Comput. Graphics 9 (4) (2003) 433–442.
- [27] H. Theisel, C. Rossl, H. Seidel, Combining topological simplification and topology preserving compression for 2d vector fields, in: 11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings, 2003, pp. 419–423.
- [28] X. Tong, T.-Y. Lee, H.-W. Shen, Salient time steps selection from large scale timevarying data sets with dynamic time warping, in: IEEE Symposium on Large Data Analysis and Visualization (LDAV), IEEE, 2012, pp. 49–56.

- [29] L. Orf, A violently tornadic supercell thunderstorm simulation spanning a quarter-trillion grid volumes: Computational challenges, i/o framework, and visualizations of tornadogenesis, Atmosphere 10 (10).
- [30] P. Lindstrom, M. Isenburg, Fast and efficient compression of floating-point data, IEEE Trans. Vis. Comput. Graphics 12 (5) (2006) 1245–1250.
- [31] H. Childs, et al., A terminology for in situ visualization and analysis systems, Int. J. High Perform. Comput. Appl. 34 (6) (2020) 676–691.
- [32] M. Larsen, J. Ahrens, U. Ayachit, E. Brugger, H. Childs, B. Geveci, C. Harrison, The alpine in situ infrastructure, in: Proceedings of the in Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization, ACM, 2017, pp. 42–46.
- [33] K. Moreland, C. Sewell, W. Usher, L.-t. Lo, J. Meredith, D. Pugmire, J. Kress, H. Schroots, K.-L. Ma, H. Childs, et al., Vtk-m: Accelerating the visualization toolkit for massively threaded architectures, IEEE Comput. Graph. Appl. 36 (3) (2016) 48–58.
- [34] D. Pugmire, A. Yenpure, M. Kim, J. Kress, R. Maynard, H. Childs, B. Hentschel, Performance-portable particle advection with VTK-m, in: Eurographics Symposium on Parallel Graphics and Visualization, The Eurographics Association, 2018.
- [35] A. Agranovsky, D. Camp, K.I. Joy, H. Childs, Subsampling-based compression and flow visualization, in: Visualization and Data Analysis 2015, 9397, International Society for Optics and Photonics, SPIE, 2015, pp. 207–220.
- [36] The CGAL Project, [CGAL] User and reference manual, 5.3 edition, CGAL editorial board, 2021, URL https://doc.cgal.org/5.3/Manual/packages.html.
- [37] H. Childs, Visit: An end-user tool for visualizing and analyzing very large data. [38] A.S. Almgren, J.B. Bell, M.J. Lijewski, Z. Lukić, E. Van Andel, Nyx: A massively
- parallel amr code for computational cosmology, Astrophys. J. 765 (1) (2013) 39. [39] N.A. Petersson, B. Sjögreen, Wave propagation in anisotropic elastic materials
- and curvilinear coordinates using a summation-by-parts finite difference method, J. Comput. Phys. 299 (2015) 820–841.
  [40] A. Mallinson, D.A. Beckingsale, W. Gaudin, J. Herdman, J. Levesque, S.A. Jarvis,
- Cloverleaf: Preparing Hydrodynamics Codes for Exascale, The Cray User Group, 2013.
- [41] S. McIntosh-Smith, J. Price, T. Deakin, A. Poenaru, A performance analysis of the first generation of hpc-optimized arm processors, Concurr. Comput.: Pract. Exper. 31 (16) (2019) e5110.
- [42] T. Odajima, Y. Kodama, M. Tsuji, M. Matsuda, Y. Maruyama, M. Sato, Preliminary performance evaluation of the fujitsu a64fx using hpc applications, in: 2020 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2020, pp. 523–530.
- [43] S. Li, S. Sane, L. Orf, P. Mininni, J. Clyne, H. Childs, Spatiotemporal wavelet compression for visualization of scientific simulation data, in: 2017 IEEE International Conference on Cluster Computing (Cluster), IEEE, 2017, pp. 216–227.
- [44] S. Labasan, M. Larsen, H. Childs, B. Rountree, Power and performance tradeoffs for visualization algorithms, in: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2019, pp. 325–334.
- [45] I. Yakushin, K. Mehta, J. Chen, M. Wolf, I. Foster, S. Klasky, T. Munson, Featurepreserving lossy compression for in situ data analysis, in: 49th International Conference on Parallel Processing-ICPP: Workshops, 2020, pp. 1–9.
- [46] J. Kress, M. Larsen, J. Choi, M. Kim, M. Wolf, N. Podhorszki, S. Klasky, H. Childs, D. Pugmire, Comparing time-to-solution for in situ visualization paradigms at scale, in: 2020 IEEE 10th Symposium on Large Data Analysis and Visualization (LDAV), IEEE, 2020, pp. 22–26.
- [47] M. Larsen, L. Livermore, C. Harrison, T.L. Turton, S. Sane, S. Brink, H. Childs, Trigger happy: Assessing the viability of trigger-based in situ analysis, in: IEEE Symposium on Large Data Analysis and Visualization (LDAV), 2021.
- [48] J. Reinders, Intel Threading Building Blocks: Outfitting C++ for Multi-Core Processor Parallelism, O'Reilly Media, Inc., 2007.
- [49] N. Morrical, I. Wald, W. Usher, V. Pascucci, Accelerating unstructured mesh point location with rt cores, IEEE Trans. Vis. Comput. Graphics.



Journal of Computational Science 61 (2022) 101615

Sudhanshu Sane completed a Ph.D. in Computer Science at the University of Oregon in June 2020. Upon graduation, Sudhanshu joined the Scientific Computing and Imaging Institute at the University of Utah as a postdoctoral research fellow. Sudhanshu is currently a software engineer at Luminary Cloud. Sudhanshu's research interests include computational fluid dynamics, uncertain and multivariate data visualization, and high performance computing.



Chris Johnson is a Distinguished Professor of Computer Science and founding director of the Scientific Computing & Imaging (SCI) Institute at the University of Utah. He also holds faculty appointments in the Departments of Physics and Bioengineering. His research interests are in the areas of scientific computing and scientific visualization. In 1992, with Professor Rob MacLeod, Professor Johnson founded the SCI research group, now the SCI Institute, which has grown to employ over 150 faculty, staff and students. Professor Johnson serves on a number of international journal editorial and advisory boards to national and international research centers. He is a Fellow of AIMBE (2004), AAAS (2005), SIAM (2009), and IEEE (2014) and was recently inducted into the IEEE Visualization Academy (2019). He has received a number of awards including the NSF Presidential Faculty Fellow (PFF) award from President Clinton, a DOE Computational Science Award, the Governor's Medal for Science and Technology, the Utah Cyber Pioneer Award, the IEEE Visualization Career Award, IEEE CS Charles Babbage Award, the IEEE Sidney Fernbach Award, Rosenblatt Prize and most recently, the 2020 Leonardo Award.



Hank Childs is a professor of computer and information science at the University of Oregon. He received a Ph.D. in computer science from the University of California at Davis in 2006. His research focuses on scientific visualization, high performance computing, and the intersection of the two.