Ascent: Flyweight In Situ Visualization and Analysis for HPC Simulations

ATPESC 2020

Matt Larsen (LLNL) + Cyrus Harrison (LLNL)

Monday August 3rd, 2020

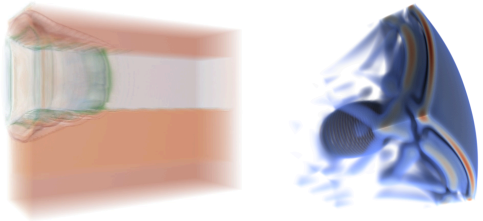
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Ascent is an easy to use flyweight in situ visualization and analysis library for HPC simulations

- **Easy to use in-memory visualization and analysis**
  - Use cases: ***Making Pictures, Transforming Data,*** and ***Capturing Data***
  - Young effort, yet already supports most common visualization operations
  - Provides a simple infrastructure to integrate custom analysis
  - Provides C++, C, Python, and Fortran APIs
- **Uses a flyweight design targeted at next-generation HPC platforms**
  - Efficient distributed-memory (MPI) and many-core (CUDA or OpenMP) execution
    - Demonstrated scaling: In situ filtering and ray tracing across ***16,384 GPUs*** on LLNL's Sierra Cluster
  - Has lower memory requirements than current tools
  - Requires less dependencies than current tools (ex: no OpenGL)
    - Builds with Spack https://spack.io/

Ascent



Visualizations created using Ascent

python™ HDF CINEMA

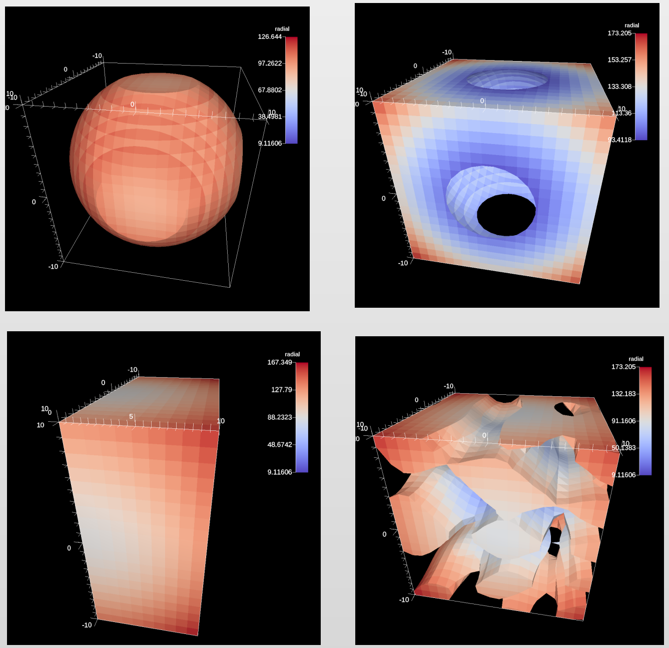
Extracts supported by Ascent

http://ascent-dav.org

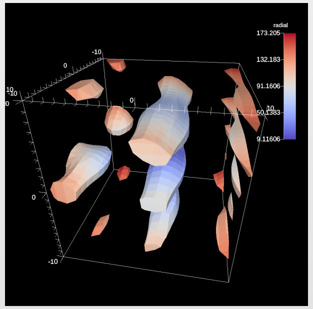
https://github.com/Alpine-DAV/ascent

Website and GitHub Repo

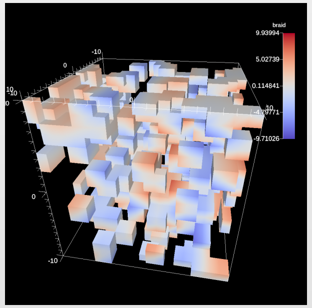
Ascent is ready for common visualization use cases



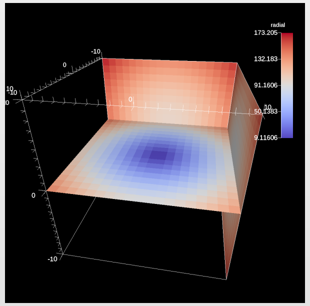
Clips



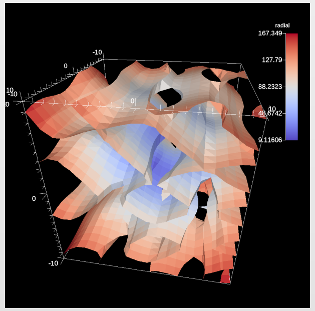
Iso-Volume



Threshold

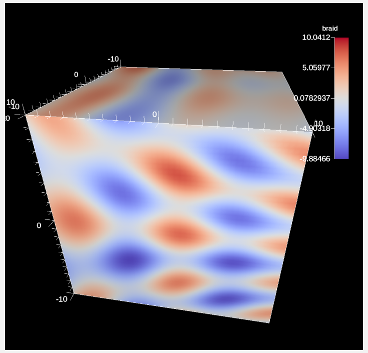


Slice

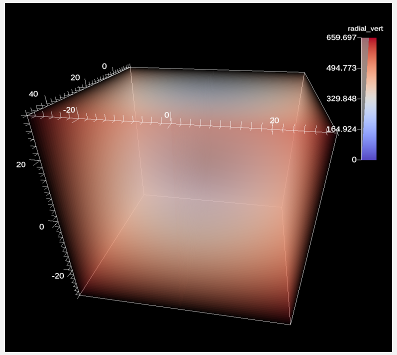


Contour

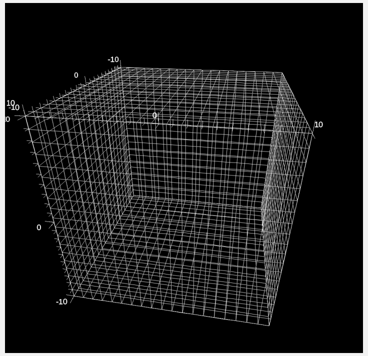
Rendering



Pseudocolor

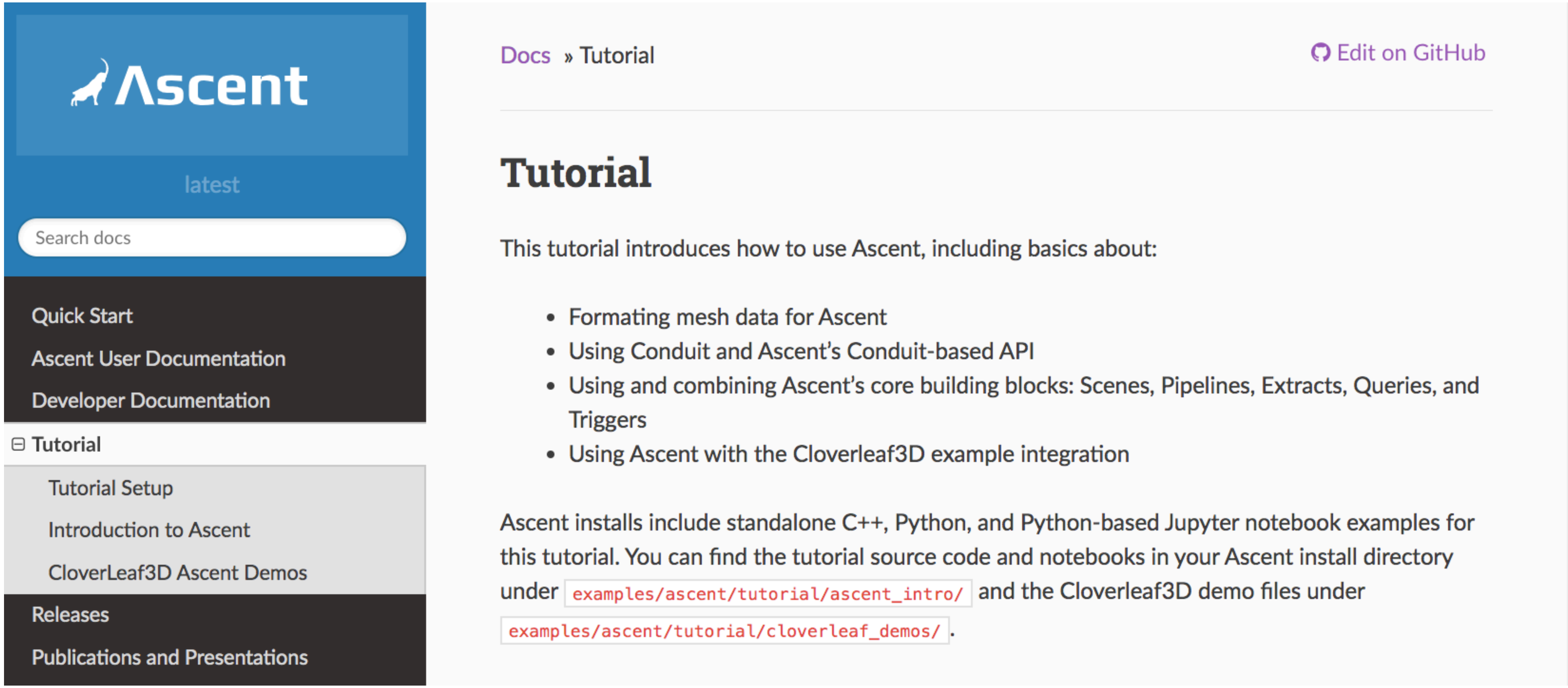


Volume



Mesh

Ascent tutorial examples are outlined in our documentation and included ready to run in Ascent installs



Tutorial

This tutorial introduces how to use Ascent, including basics about:

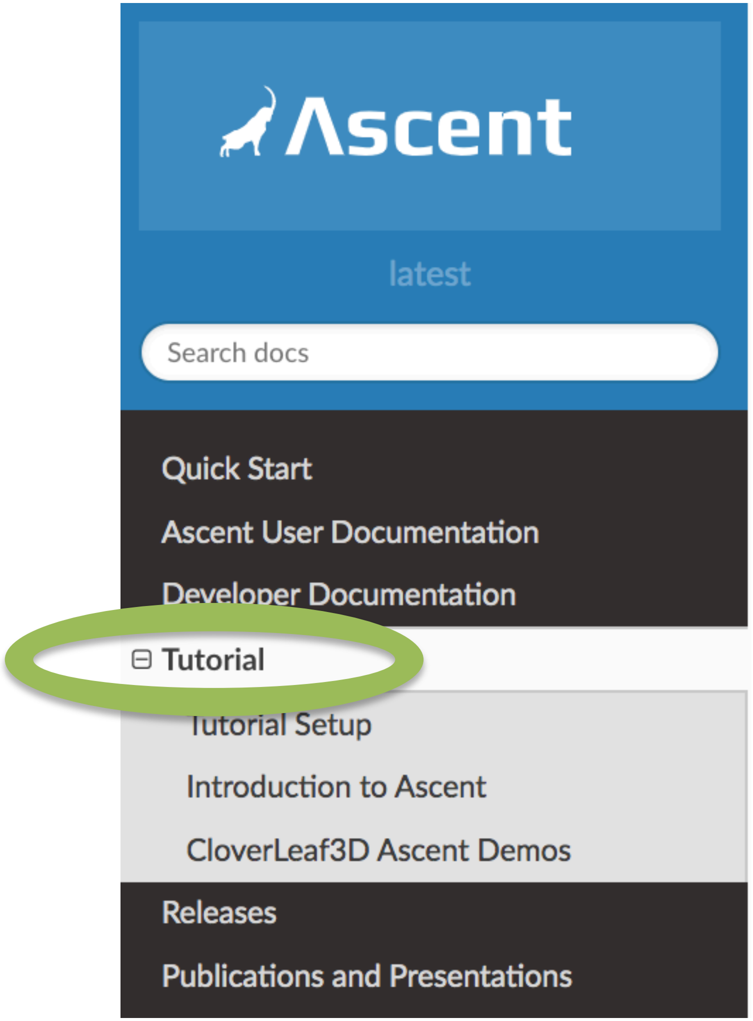
- Formating mesh data for Ascent
- Using Conduit and Ascent's Conduit-based API
- Using and combining Ascent's core building blocks: Scenes, Pipelines, Extracts, Queries, and Triggers
- Using Ascent with the Cloverleaf3D example integration

Ascent installs include standalone C++, Python, and Python-based Jupyter notebook examples for this tutorial. You can find the tutorial source code and notebooks in your Ascent install directory under `examples/ascent/tutorial/ascent_intro/` and the Cloverleaf3D demo files under `examples/ascent/tutorial/cloverleaf_demos/`.

http://ascent-dav.org

Ascent tutorial examples are outlined in our documentation and included ready to run in Ascent installs

- http://ascent-dav.org
- Click on “Tutorial”



Ascent

latest

Search docs

Quick Start

Ascent User Documentation

Developer Documentation

⊟ Tutorial

Tutorial Setup

Introduction to Ascent

CloverLeaf3D Ascent Demos

Releases

Publications and Presentations

Ascent’s interface provides five composable building blocks

Scenes
(Render Pictures)

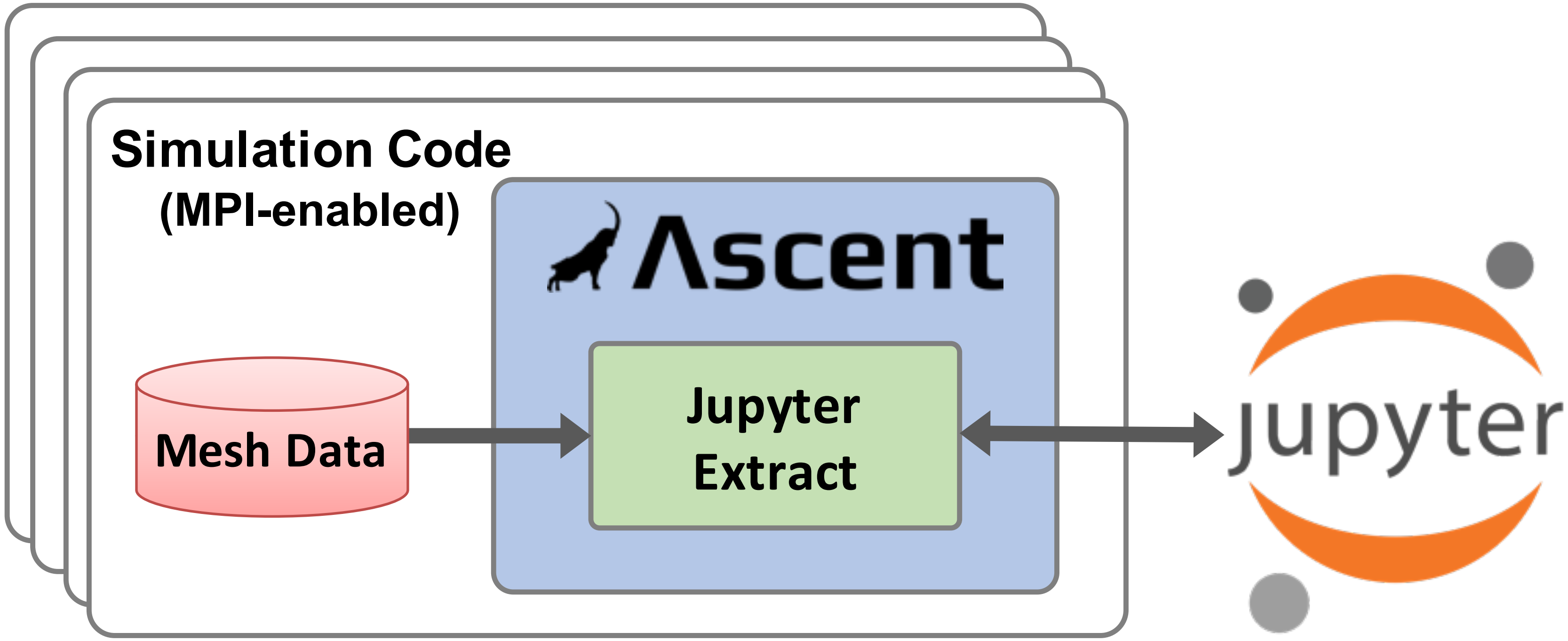
Pipelines
(Transform Data)

Extracts
(Capture Data)

Queries
(Ask Questions)

Triggers
(Adapt Actions)

Ascent’s Jupyter Extract provides a path to connect your simulation to a Jupyter Notebook



With the *Jupyter Extract*, users of any simulation code with Ascent integrated can run Jupyter Notebooks and use Python to interact with in-memory data

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.
Lawrence Livermore National Security, LLC

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.

Lawrence Livermore
National Laboratory

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.