Flow Visualization using Lagrangian Paradigm

CIS 410/510: Scientific Visualization

Sudhanshu Sane
Announcements

• Hank is out of town.
• Project 4G was due last night.
• Friday: No lecture. (Hank is doing a YouTube lecture video?)
• Project 5 due Saturday – 10th of February
Outline

• Review: Flow Visualization
• Flow Field Exploration
• Lagrangian-based Flow Visualization
Particle Advection

- Displace massless particle based on velocity field.
- Most fundamental operation for flow visualization.

Primary building block for calculating particle trajectories.
Vector Field Example – Double Gyre

GIF Credit: Shadden Lab. UC Berkeley
Particle Advection is the basis for many Flow Visualization techniques.

Pathlines for an F5 Tornado simulation with a mature vortex.
Particle Advection is the basis for many Flow Visualization techniques

• Takeaways:
  • Used for diverse analyses.
  • Several techniques such as streamlines, pathlines, streamsurfaces, LIC etc.
Traditional Methods for Calculating Particle Trajectories

• Advection Method:
  • Euler Method
  • Runge-Kutta Method

• Type of vector field:
  • Steady state (only spatial linear interpolation)
  • Unsteady state (temporal and spatial linear interpolation)
Traditional Methods for Calculating Particle Trajectories

• Steps involved:
  • Save vector field information during simulation.
  • Place seed particle at location $P$.
  • Load vector field/s.
  • Use vector field information available to perform spatial/temporal interpolation and calculate vector $V$ at location $P$.
  • Perform advection.
Outline

• Review: Flow Visualization
• Flow Field Exploration
• Lagrangian-based Flow Visualization
Traditional Explorative Flow Analysis

• User requests analysis that requires one or more particles to be advected.
• The trajectory is calculated by repeated evaluation of the velocity field.

• (If this was not explorative – if you knew the particle locations ahead of time – then this would be trivial: you would calculate the trajectories in situ.)
Traditional Explorative Flow Analysis

**Takeaway**: Trajectory is calculated via repeated evaluations of the velocity field at arbitrary positions and times.

**So**: how do these evaluations happen?
Example

- Velocities Needed:
  - P=P0, T=0
  - P=P1, T=0.1
  - P=P2, T=0.2
  - P=P3, T=0.3
  - P=P4, T=0.4
  - P=P5, T=0.5
  - P=P6, T=0.6

- Time slices saved:
  - T=0
  - T=1

Velocities at times between times slices are calculated through interpolation.

This introduces error.
Supercomputers’ ability to *generate* data is increasing much faster than their ability to *store* data.

<table>
<thead>
<tr>
<th>Machine</th>
<th>FLOPS</th>
<th>I/O</th>
<th>Year of Debut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaguar</td>
<td>1.75 PetaFLOPs</td>
<td>240 GB/s</td>
<td>2009</td>
</tr>
<tr>
<td>Titan</td>
<td>27 PetaFLOPs</td>
<td>240 GB/s -&gt; 1 TB/s</td>
<td>2012</td>
</tr>
<tr>
<td>Summit</td>
<td>&gt;135 PetaFLOPs</td>
<td>1 TB/s</td>
<td>2017</td>
</tr>
</tbody>
</table>

FLOPS – 77x increase  
I/O – 4x increase
Example revisited

• Velocities Needed:
  • $P=P_0$, $T=0$
  • $P=P_1$, $T=0.1$
  • $P=P_2$, $T=0.2$
  • $P=P_3$, $T=0.3$
  • $P=P_4$, $T=0.4$
  • $P=P_5$, $T=0.5$
  • $P=P_6$, $T=0.6$

• Time slices saved:
  – $T=0$
  – $T=10$

Simulation codes reduce the rate they save data when relative I/O drops
→
Analysis will be increasingly inaccurate
What is the value of $V_x(P_0, 5)$?

$V_x(P_0, 5) = 7$
Outline

• Review: Flow Visualization
• Flow Field Exploration
• Lagrangian-based Flow Visualization
Improved Post Hoc Flow Analysis Via Lagrangian Representations

Alexy Agranovsky, David Camp, Christoph Garth, E. Wes Bethel, Kenneth I. Joy, and Hank Childs
Motivation

• Traditional paradigm: store vector field data at select cycles, and visualize the flow field after simulation completes.

• BUT: computation and I/O gap is increasing on supercomputers.
  • Traditional paradigm is in jeopardy since fewer cycles can be saved.
  • Need new approaches.

• Emerging paradigm: allows processing on data as it is generated by the simulation.
  • This is called “in situ processing.”

• The Lagrangian paradigm is perfectly suited for in situ processing.
In Situ Processing

Exascale Simulation → Reduced/Derived data set → Disk → Exploration via post hoc processing

= 1 node of supercomputer
= in situ routine
Eulerian vs Lagrangian

- Physics considers 2 frames of reference
  - Eulerian (Observer is at fixed position)
  - Lagrangian (Observer is moving through space)
Eulerian vs Lagrangian

While traditional flow analysis uses the Eulerian frame of reference, we wanted to explore an approach from the Lagrangian frame of reference.
Lagrangian Approach

• Requires calculation of “basis flows”.
• What is a basis flow?
  • Particles are placed in the volume and displaced by the velocity, forming a trajectory.
  • Basis flows are stored as particle trajectory start and end locations.
• Important: Basis flows can be calculated accurately in situ.
• Basis flows capture the behavior of the flow field over an interval of time.
• Two phase algorithm:
  • Phase 1: in situ extraction of Lagrangian basis flows
  • Phase 2: post-hoc analysis, where new trajectories are interpolated from basis flows

Exascale Simulation

Reduced/Derived data set

Exploration via post hoc processing

= 1 node of supercomputer

= in situ routine
Phase 2: Post-hoc analysis, where new trajectories are interpolated from basis flows

Figure 3: Interpolation using the Lagrangian basis. Position values of pathline $x$ are interpolated from basis pathlines $B_1$ and $B_2$, using weights $\omega_1$ and $\omega_2$, respectively. The weights are calculated based on the location of all pathlines at $t_i$ and then applied at $t_j$ to approximate $x_j$. 
Interpolation Options

• Shepard’s Method
  • Inverse distance weighting
• Moving Least Squares
  • Weighted least squares measure
• Barycentric Coordinates Interpolation
  • Linear interpolation
Phase 2: Reconstructing Flow Field Post Hoc

- Basis flows are used to reconstruct the flow field afterwards.

Phase 1: In Situ - Calculate basis flows and save.
Phase 2: Reconstructing Flow Field Post Hoc

- Basis flows are used to reconstruct the flow field afterwards.

Identify neighborhood at t=0.
Phase 2: Reconstructing Flow Field Post Hoc

- Basis flows are used to reconstruct the flow field afterwards.
Phase 2: Reconstructing Flow Field Post Hoc

• Basis flows are used to reconstruct the flow field afterwards.
Phase 2: Reconstructing Flow Field Post Hoc

- Basis flows are used to reconstruct the flow field afterwards.
Phase 2: Reconstructing Flow Field Post Hoc

- Basis flows are used to reconstruct the flow field afterwards.

**Phase 1:** In Situ - Calculate basis flows and save.

**Phase 2:** Reconstruct flow field using basis flows.
Example

Traditional Advection

Lagrangian-Based Technique

Phase 1

Phase 2
Lagrangian Paradigm

• This is a very different way of calculating particle trajectories
• The traditional (Eulerian) method does repeated vector field evaluations.
• The Lagrangian method uses basis flows, and interpolates.
  • (Although the basis flows are calculated using traditional methods)
# Summarizing Differences

<table>
<thead>
<tr>
<th></th>
<th>Traditional advection</th>
<th>Lagrangian-based method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files represent</td>
<td>Time slice</td>
<td>Time interval</td>
</tr>
<tr>
<td>Files contain</td>
<td>Vector fields</td>
<td>Particle trajectories</td>
</tr>
<tr>
<td>Reducing I/O</td>
<td>Less time slices</td>
<td>Less particles</td>
</tr>
<tr>
<td>Increasing accuracy</td>
<td>More time slices</td>
<td>More particles</td>
</tr>
<tr>
<td>Work required during simulation</td>
<td>None</td>
<td>Advecting particles</td>
</tr>
<tr>
<td>Memory required for method</td>
<td>None</td>
<td>Storing trajectories</td>
</tr>
</tbody>
</table>
Analogy: the narcoleptic

• Imagine both individuals suffer from narcolepsy.
• Each wakes up every ten minutes, is awake for one second, and then falls back asleep.
• Who has more information?
Hypothesis

• When compared with the traditional technique, the Lagrangian technique has the potential to be:
  • More accurate
  • More storage efficient
  • Faster

Note that accuracy and storage both depend on the number of basis flows, and are in tension
How do we calculate the basis flows?

• Basis Flows: Particle trajectories that are calculated in situ.

• Things to consider:
  • Placement
  • Duration

• What did we do when using the Eulerian approach?
  • Uniformly distributed vector field samples.
  • Stored vector field periodically.
Lagrangian Approach 1 – Uniform Seeding

- Step 1: Place seeds uniformly along a grid.
- Step 2: Advect particles using full simulation data for a fixed duration.
- Step 3: Save particle trajectory end locations.
- Step 4: Reset all the seeds uniformly along a grid.

- Pros: This approach is simple to implement, provides good coverage of the flow field, easy to store.
- Cons: Short fixed duration basis flows.
Uniform Seeding - Basis Flows Example

Cycle = 0
(Place particle along a uniform grid.)
Uniform Seeding - Basis Flows Example

Cycle = 10
(Write Cycle, save particle locations.)
Uniform Seeding - Basis Flows Example

Cycle = 10
(Reset. Place particle along a uniform grid.)
Uniform Seeding - Basis Flows Example

Cycle = 20
(Write Cycle, save particle locations.)
Lagrangian Approach 1 – Uniform Seeding

• Under sparse temporal settings post hoc exploration using basis flows calculated by the Uniform Seeding approach showed:
  • Improved accuracy
  • Reduced Storage
  • Faster execution times overall.

• Tests carried out proved that the hypothesis was correct!
Lagrangian Approach – Alternative methods?

• Things to consider when calculating basis flows:
  • Placement
  • Duration

• What kind of basis flows are the best to use?
  • Longer duration
  • Non divergent basis flows

• How can we generate better basis flows? What problems do we encounter? Possible solutions?