Extending Smoothed Particle Hydrodynamics Demonstration for HPC Education

Todd Hetrick UNC Charlotte
Dr. Tyler Allen, College of Computing and Informatics

Introduction

Background

- TinyTitan is a 2D Smoothed Particle Hydrodynamics HPC Demonstration program originally developed at Oak Ridge Labs
- OpenMPI is used to run the SPH simulation in a distributed environment on a cluster of NVIDIA Jetson Nano embedded computer.

Intention

- A program that can give a visual connection to HPC concepts.
- Dynamic inputs will allow an interactive environment to further understanding of key HPC concepts

Goals

Add Dynamic particle count

- Add the ability to change the particle count while the simulation is running
- Real-time changes will demonstrate dynamic loading and work distribution.
- Does changing the particle count have a balanced impact on node performance?

Improve Foundation.

- Update foundational elements of the code in a way that facilitates future improvements and increases the flexibility of the code base.

Improve Understanding.

- Research and document the flow of particles throughout the SPH compute nodes.

Process

Add Dynamic Particles

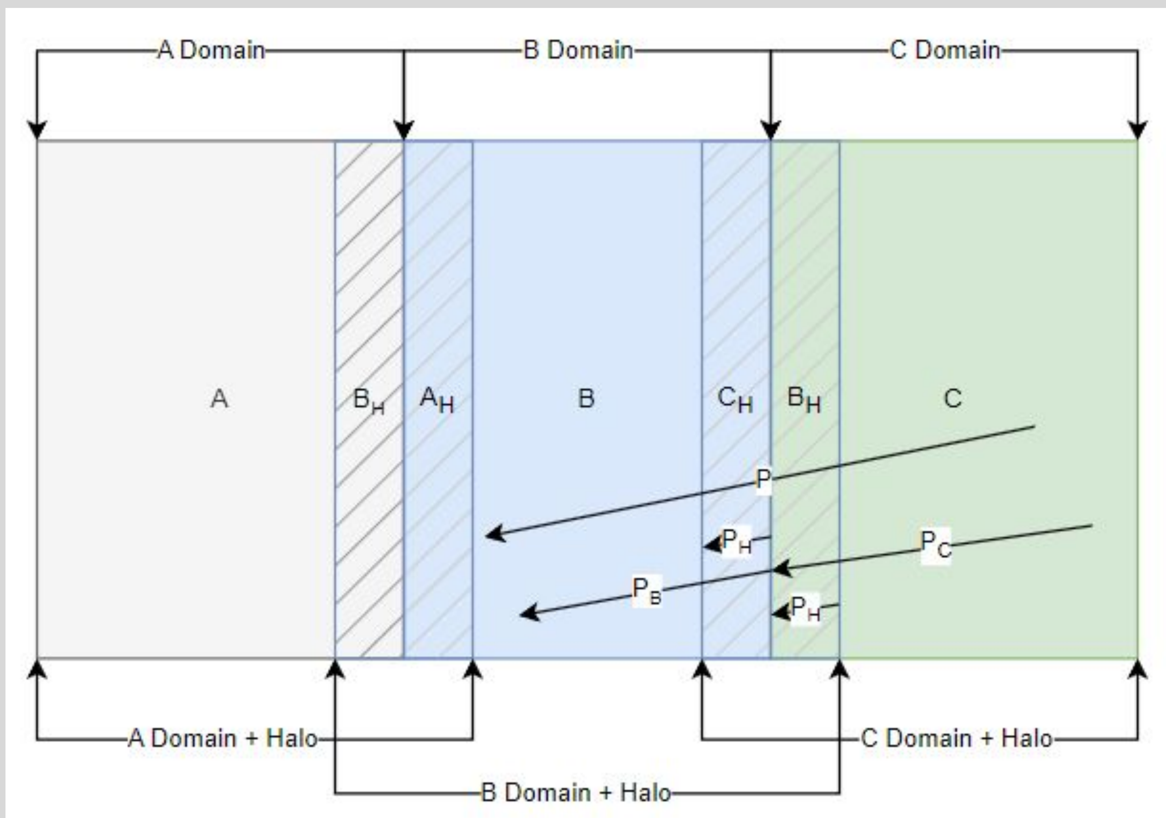
- Modify GUI to allow the user to increase and decrease the number of particles currently in the simulation with keyboard inputs
- Add code to allow these dynamic requests to be communicated to the compute nodes
- Add code to the simulation nodes that increases and decreases the particle count.

Memory management

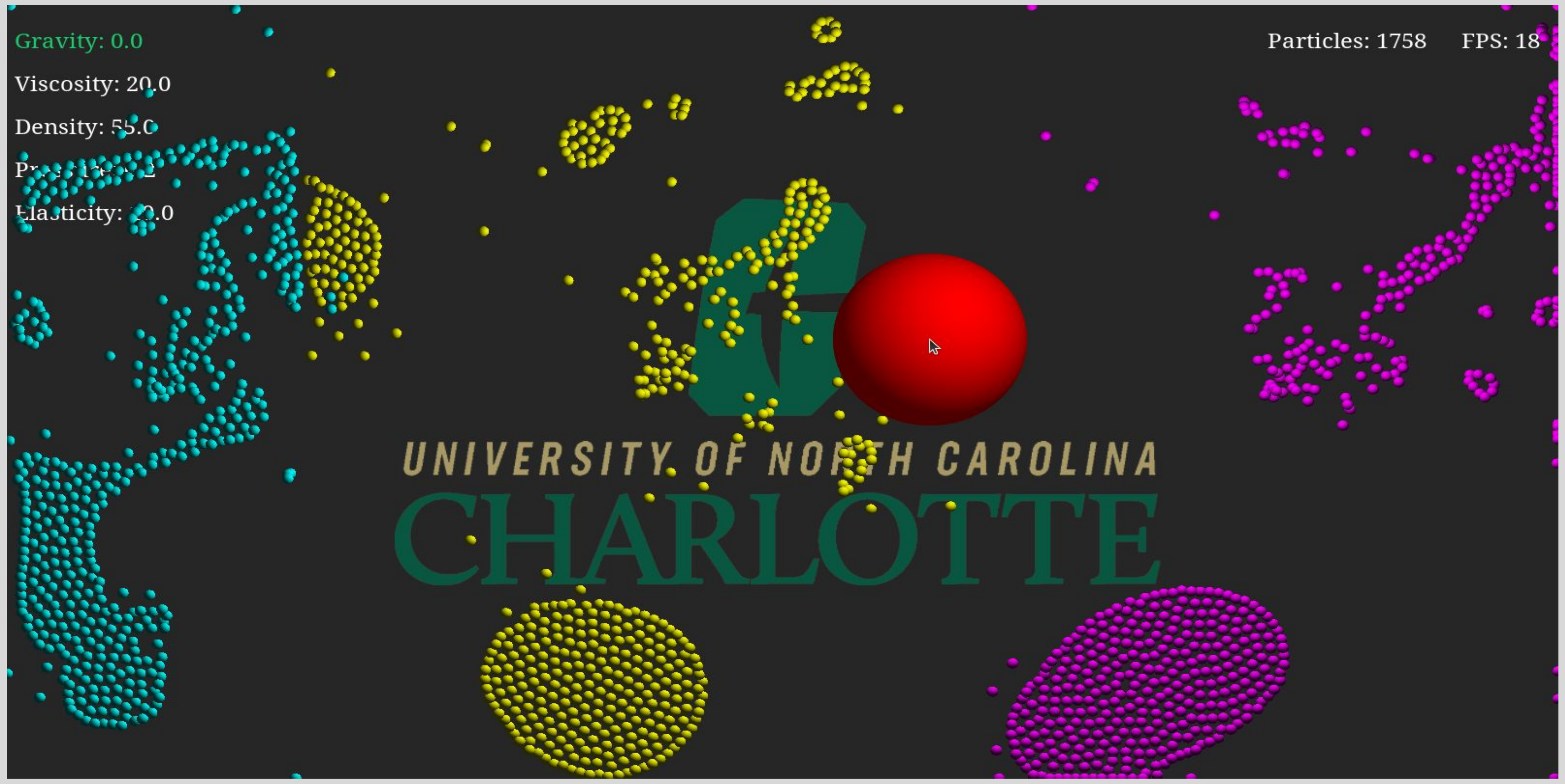
- Address consequences of adding particles to memory structures that were not intended to be dynamic.
- Verify that adding particles does not interfere with halo particles and neighboring calculations.
- Rewrite out of bounds transfer code to eliminate complexity and make particle memory accounting easier.

Testing and verification.

- Observe impact of dynamic particles on simulation FPS.
- Verify particles are redistributing throughout nodes and producing expected results.



particle lifecycle while crossing node boundaries



Challenges

Code Cleanup:

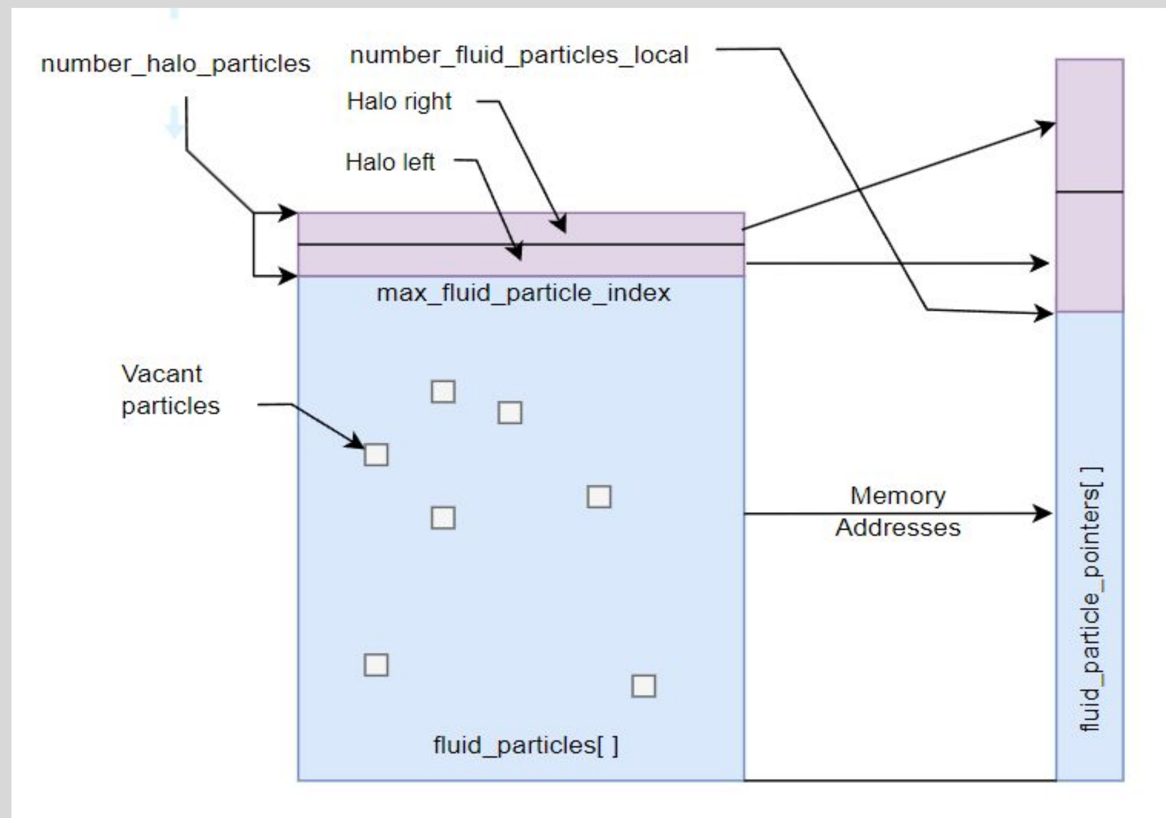
- Several incorrectly initialized variables created a series of cascading errors.

Rogue Particles

- Early attempts to add particles resulted in “rogue particles” that did not behave in the simulation correctly.

Particle transfer complexity

- Out Of Bounds transfers were made using custom MPI type, hiding important particle indexes.



particle memory accounting

Results

Dynamic Visualization

- Particles can be added or removed while the simulation is running.
- Users can see first hand the results on performance when adding particles as the FPS count changes under load

Improved Code

- A simplification of the out of bounds transfer process was implemented.

Conclusions

Dynamic particle loading.

- The ability to change the particle count during runtime opens up a wide array of testing and learning opportunities for HPC education.

Future.

- Other parameters within the simulation may bring additional educational benefit if they are made dynamic.