

Karin Metzgar
ASTE 546
HW 7

To compile use:

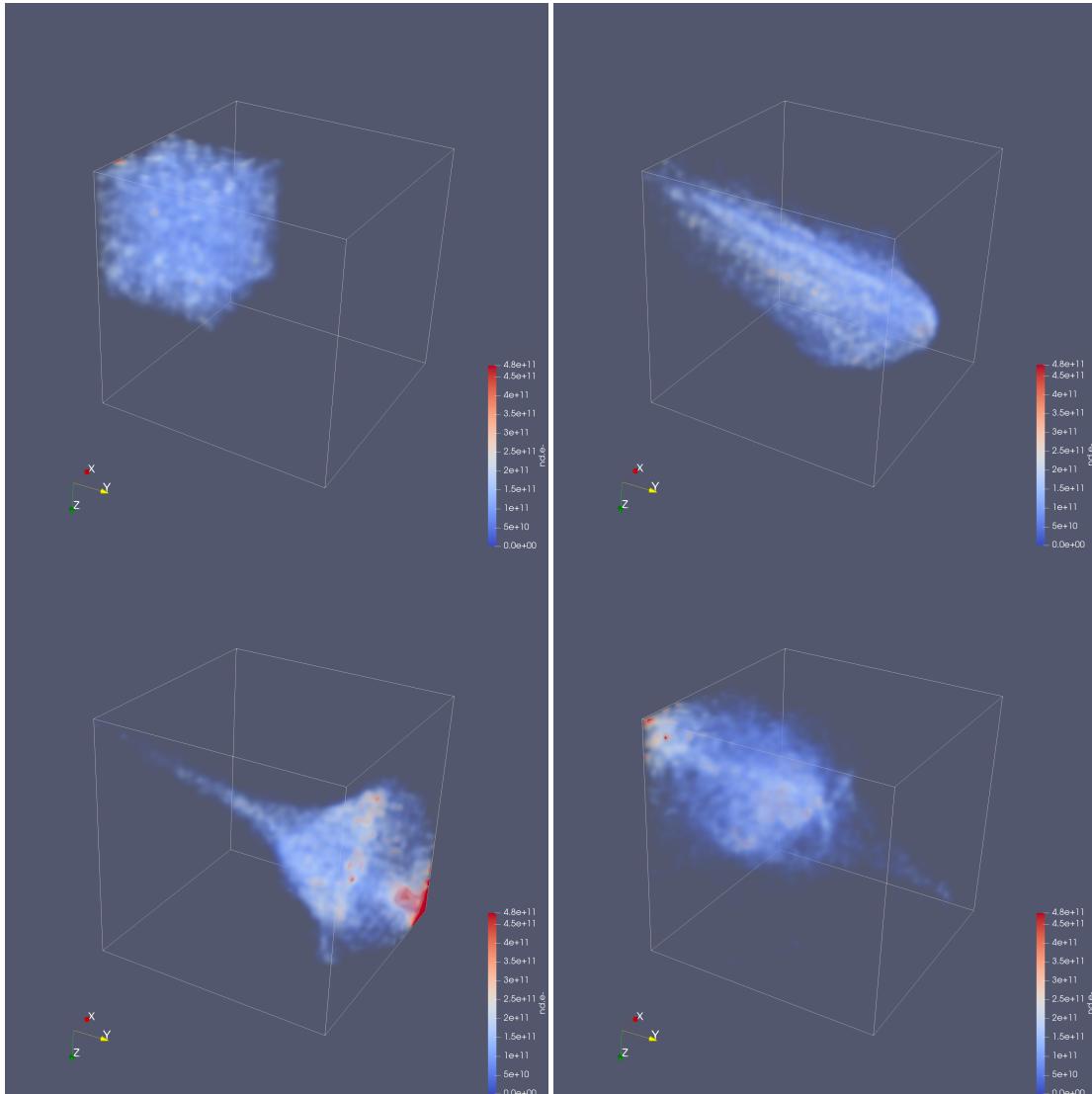
```
clang++ -O3 -std=c++11 Main.cpp Output.cpp PotentialSolver.cpp  
Species.cpp World.cpp -o box
```

without the `-std=c++11` it doesn't include c++11 and won't compile

Either `-O2` optimization flag or the more aggressive `-O3` can be used according to chatGPT, I tried both.

The executable file "box" would not create the output files when running from a double click, I had to back out a folder and run "ls ld box" on the folder titled "box", which produced a pop up for access to the documents folder and then ran the box executable `./box` from Terminal and that solved the permissions issue.

Box Example data:



Box with moat data:

Changes made:

In Main.cpp change the bounds of the domain:

```
//initialize domain
World world(41,41,41);
world.setExtents({-0.2,-0.2,-0.1},{0.2,0.2,0.3});
world.setTime(1e-10,40000);
```

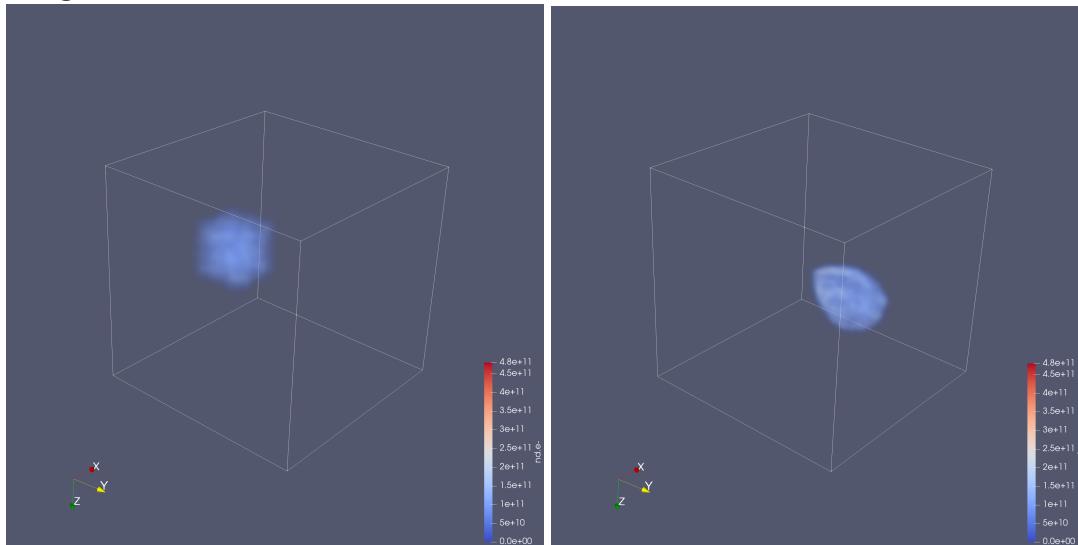
In Main.cpp change the bounds of the species, to reflect the original bounds:

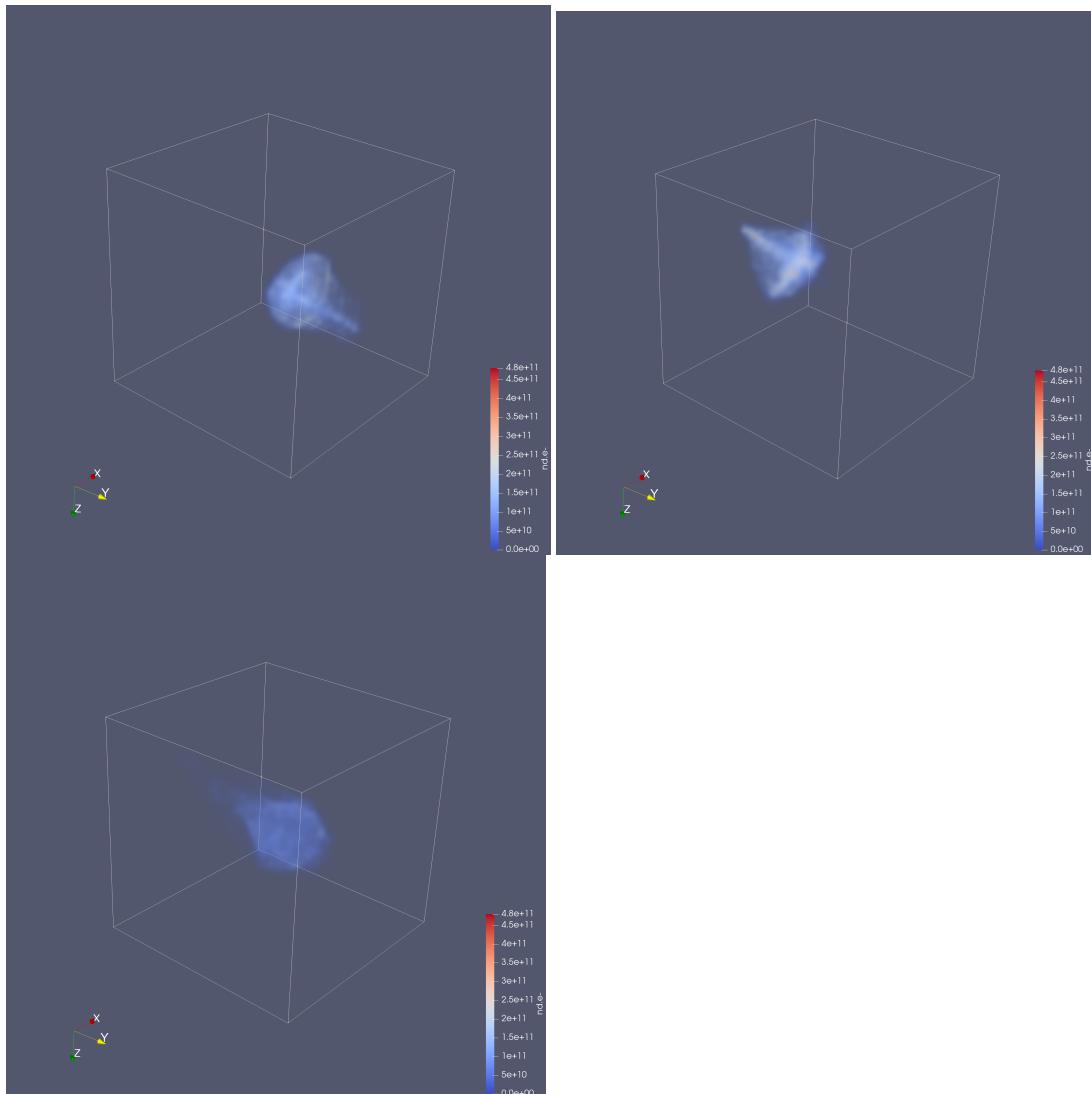
```
//load species in the bounds within the moat / old bounds
double3 oldx0 = {-0.1,-0.1,0.0};
double3 oldxc = {0.1,0.1,0.2};
double3 elecxc = {0.0, 0.0, 0.1};
species[0].loadParticlesBox(oldx0,oldxc,1e11,np_ions); //ions
species[1].loadParticlesBox(oldx0,elecxc,1e11,np_eles); //electrons
```

In Main.cpp I also changed the input to PotentialSolver from 10000 to 1000 – this sped up processing time for me pretty significantly.. hopefully at not too much detriment to data quality.

```
/*initialize potential solver and solve initial potential*/
PotentialSolver solver(world,1000,1e-4);
solver.solve();
```

Images:





I was surprised that the electron dynamics were so different, in the original example they reflect off the corner almost like you'd expect a fluid to (minus the oscillation etc. of course) but they sort of slosh back and forth. In the appended example the behavior was a bit more unexpected to me. They stay in a tighter group at the corners but seem to dissipate in the middle of the box. Some of this may be do to the change I made to the call to PotentialSolver, I am currently compiling resetting that back to 10000.

Original: Simulation took 2.59e+04

Modified: Simulation took 6.17e+03

Videos:

