Karin Metzgar

HW 5

Part 1: Quiet Start

1. Changing the velocity to represent 0.1eV ions and 1eV electrons, not sure why it took me so many tries to figure out but got it :
2. //Velocities associated with 0.1 eV for ions, 1eV for electrons
3. //double vth\_i = sqrt(( 2.0 \* (EvToK \* 0.1) \* Const::Kb ) / ions.m);
4. //double vth\_e = sqrt(( 2.0 \* (EvToK \* 1.0) \* Const::Kb ) / eles.m );
5. //think I did it wrong
6. double T\_eles, T\_ions;
7. T\_eles = EvToK \* 1.0;
8. T\_ions = EvToK \* 0.1;
9. …
10. //Random Start
11. part->x = world.x0 + rnd()\*(world.xm-world.x0);
12. part->v = 0; //stationary
13. //part->v = vth\_i\*(rnd()+rnd()+rnd()-1.5);
14. part->v = ions.sampleVel(T\_ions); //\*(rnd()+rnd()+rnd()-1.5);
15. ions.np++; //increment counter of particles
16. }
17. …
18. //Random Start
19. eles.part[p].x = world.x0 + rnd()\*(world.xm-world.x0);
20. eles.part[p].v = 0; //stationary
21. //eles.part[p].v = vth\_e\*(rnd()+rnd()+rnd()-1.5);
22. eles.part[p].v = eles.sampleVel(T\_eles); //\*(rnd()+rnd()+rnd()-1.5);
23. eles.np++;
24. …
25. if (world.ts%100==0) {
26. double T\_ions\_ave, T\_eles\_ave;
27. T\_ions\_ave = ((ions.getAveKE() \* 2.0) / Const::Kb) \* (1/EvToK);
28. T\_eles\_ave = ((eles.getAveKE() \* 2.0) / Const::Kb) \* (1/EvToK);
29. diag<<world.ts<<","<<world.ts\*world.dt<<","<<ions.np<<","<<eles.np;
30. diag<<","<<ions.getAveKE()/Const::QE<<","<<eles.getAveKE()/Const::QE;
31. diag<<","<<ions.getCurrent(world)<<","<<-eles.getCurrent(world)<< "," << T\_ions\_ave << "," << T\_eles\_ave <<"\n";
32. }

A graph showing a line

Description automatically generated with medium confidence

For random start I am getting -27 to 3.5 instead of 0 to 3.5 range

Silent Start:

// inject stationary particles

for (int p=0;p<ions.np\_alloc;p++) {

Particle \*part = ions[p];

//Quiet Start

//np\_alloc or N (400000) particles evenly spaced across a length of

//x0 and xm, or 0 and 0.1

if (p<(ions.np\_alloc-1)) {

part->x = world.x0 + (p \* partSpacing);

}

else {

part->x = world.x0 + (p \* partSpacing) - (0.0001\*partSpacing);

}

part->v = 0;

//part->v = vth\_i\*(rnd()+rnd()+rnd()-1.5);

part->v = ions.sampleVel(T\_ions);

ions.np++;

/\*

//Random Start

part->x = world.x0 + rnd()\*(world.xm-world.x0);

part->v = 0; //stationary

//part->v = vth\_i\*(rnd()+rnd()+rnd()-1.5);

part->v = ions.sampleVel(T\_ions);

ions.np++; //increment counter of particles

\*/

}

// inject stationary particles

for (int p=0;p<eles.np\_alloc;p++) {

//Quiet Start

//np\_alloc or N (400000) particles evenly spaced across a length of

//x0 and xm, or 0 and 0.1

if (p<(eles.np\_alloc-1)) {

eles.part[p].x = world.x0 + (p \* partSpacing);

}

else {

eles.part[p].x = world.x0 + (p \* partSpacing) - (0.0001\*partSpacing);

}

eles.part[p].v = 0;

//eles.part[p].v = vth\_e\*(rnd()+rnd()+rnd()-1.5);

eles.part[p].v = eles.sampleVel(T\_eles);

eles.np++;

/\*

//Random Start

eles.part[p].x = world.x0 + rnd()\*(world.xm-world.x0);

eles.part[p].v = 0; //stationary

//eles.part[p].v = vth\_e\*(rnd()+rnd()+rnd()-1.5);

eles.part[p].v = eles.sampleVel(T\_eles);

eles.np++;

\*/

}

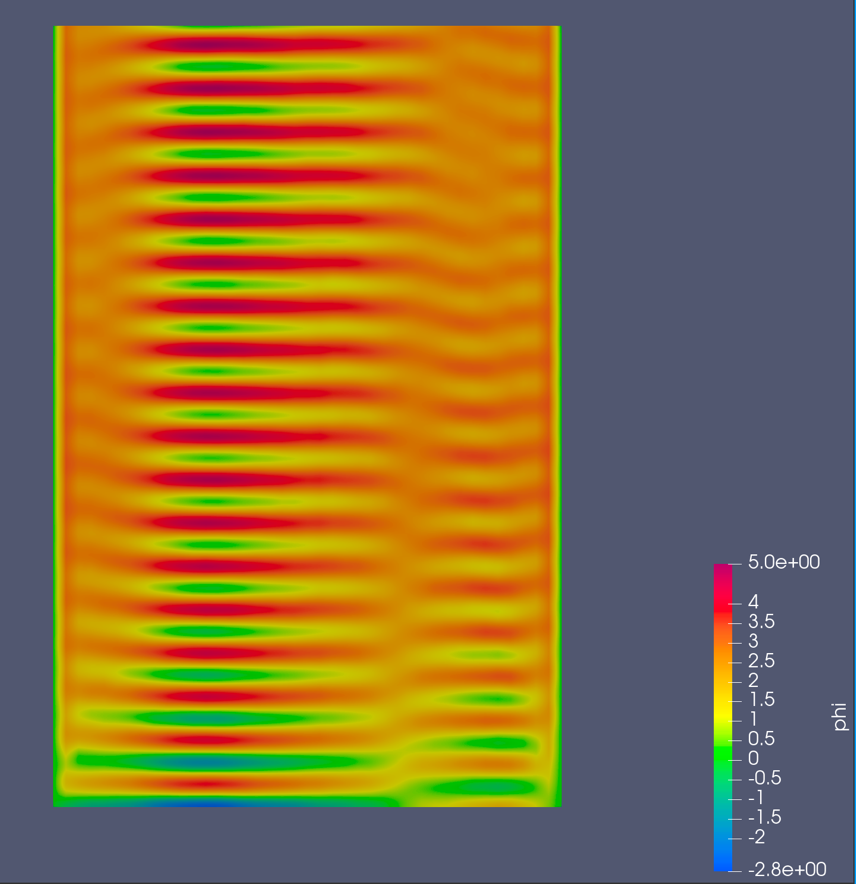
I’m not sure why the range is not from 0 – 3.5 for these, like in the example images.

Silent Start

A screen shot of a heat meter

Description automatically generated

Random Start position



Also not sure why it won’t plot the whole graph, but the maximums are 4 counts apart, which is 4E-11 seconds

///

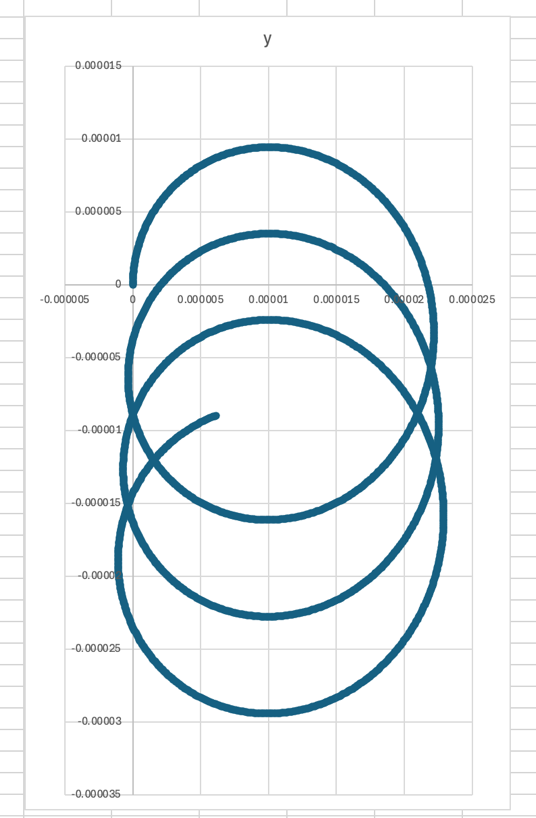
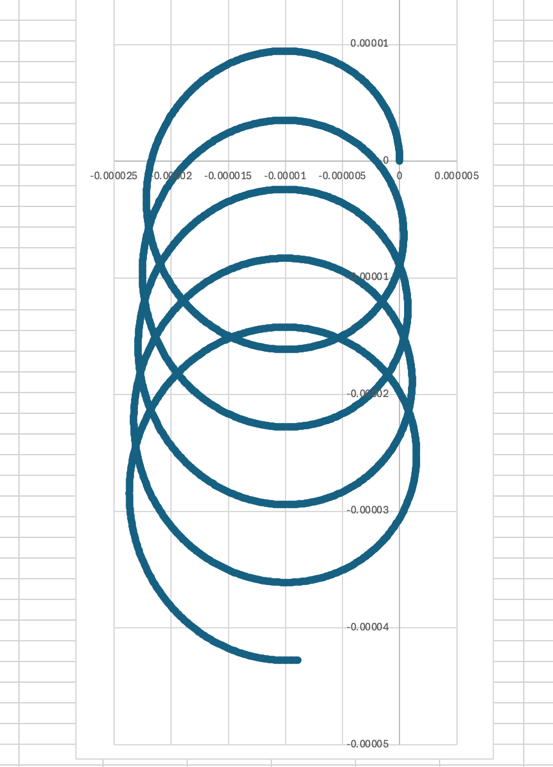
A screenshot of a computer

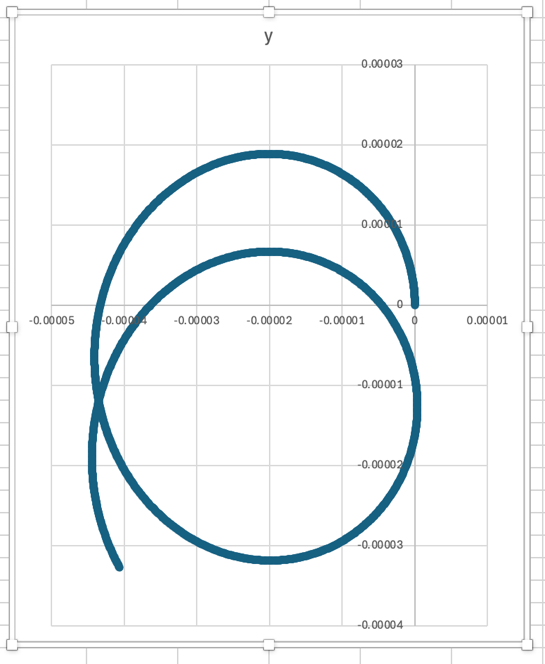
Description automatically generated

Part 2: Boris Push

The particle trajectory seems to increase in radius over time, I’m not sure if this is a result of some numerical instability going on or if my code is incorrect.

Particle 1 on the left, particle 2 on the right, particle 1 with twice the mass on the bottom.





Visualize the data in Paraview

