Temperature Isotherms: HW2

David Jedynak

February 13, 2018

Abstract

This paper will demonstrate how to use the Molecular Dynamics simulation to show how changing density (rho) at several near constant temperatures will effect pressure and what phase the particles are in. A procedure will be given to document how the simulation was used to produce the results in this paper. First, the ideal gas law will be observed by keeping the temperature relatively high and density will change. the particles will not go through a phase change though. Second, the temperature will be set relatively low, and the density will be changed to attempt a change of phase.

1 Introduction

A temperature isotherm is a system at two different states but both are held at the same temperature. The two states can differ in density and pressure, but the temperature will be equivalent.

2 Procedure

1. Open terminal and launch the MD program with the following command

   ```
   $ ./MDA_Therm_Exec
   ```

2. Click graph, init, measure, and Measurements. All of these should bring up a separate window. Arrange these on your desktop so you can see all of them.
3. leave other settings to defaults
4. in the Measurements window, click T, P, Pnid, and rho, all of these should have a dynamic graph once cont is set to on.
5. in the measure window, set T = 2, click "Set T"
6. in the measure window, set rho = 0.5, click "Set rho"
7. in the main window, set cont to on
8. immediately click set T to keep the T at 2
9. repeat step 8 until the T (temperature) on the Measurements graph settles to 2+-0.1.
10. click the cont button in the main window to stop the simulation
11. record the values of P (pressure) Pnid (ideal pressure) and T (temperature) to be plotted later
12. change the value of rho in the measure window to 0.4 and click "set rho"
13. click cont in the main window to resume the simulation
14. repeat steps 6 to 13 with different rho values of [0.5, 0.6, 0.7 ,0.8 ,0.9 ,1.0 ,1.1 ,1.2 ,1.3 ,1.4]
15. plot the values T, P, Pnid vs rho to see how these values changed with respect to changing density
16. repeat steps 5 to 15 with a T value of 0.05 and rho values of [0.5 ,0.4 ,0.3 ,0.2 ,0.15 ,0.1 ,0.08 ,0.06 ,.04 ,0.02 , 0.01]

3 Results

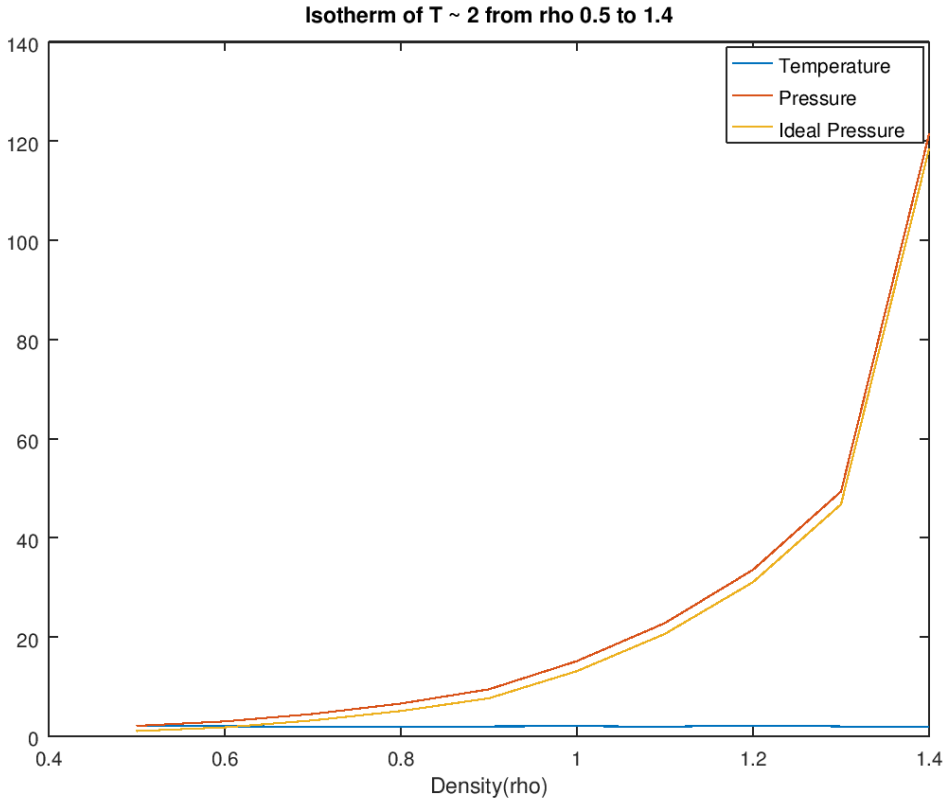

Figure 1:

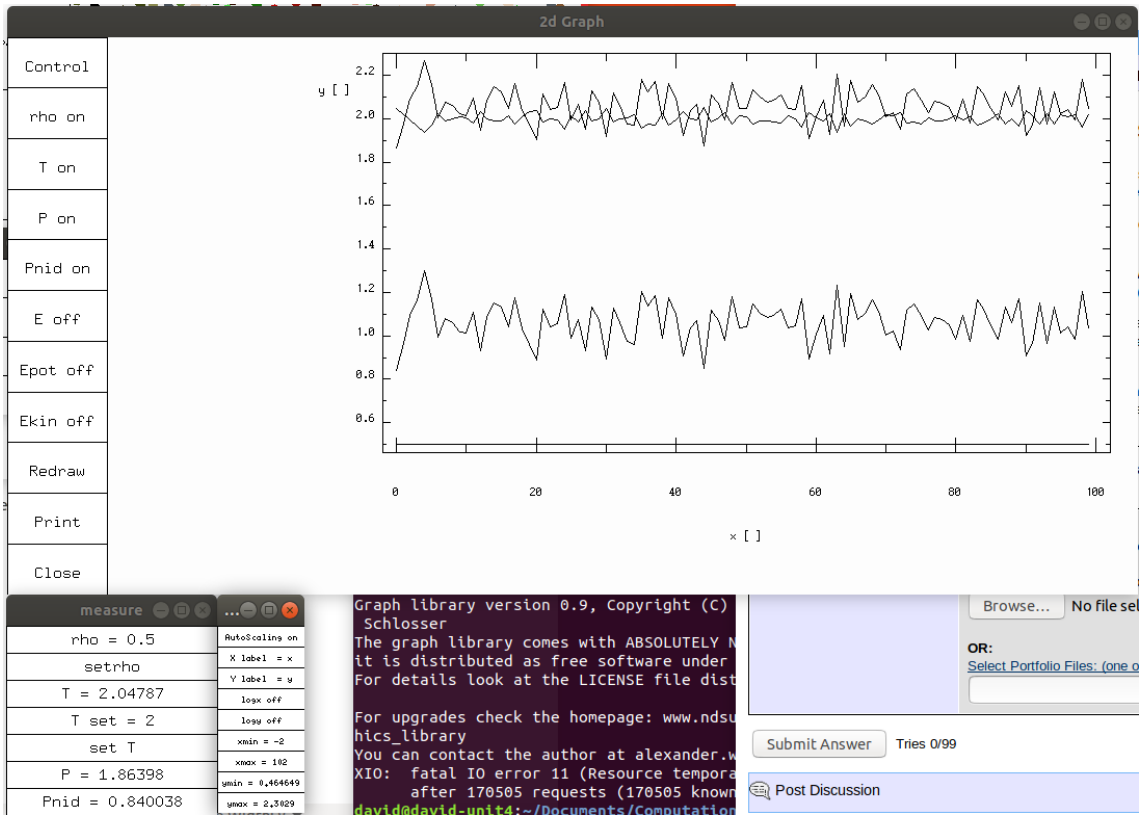

Figure 2: Initial condition for T = 2

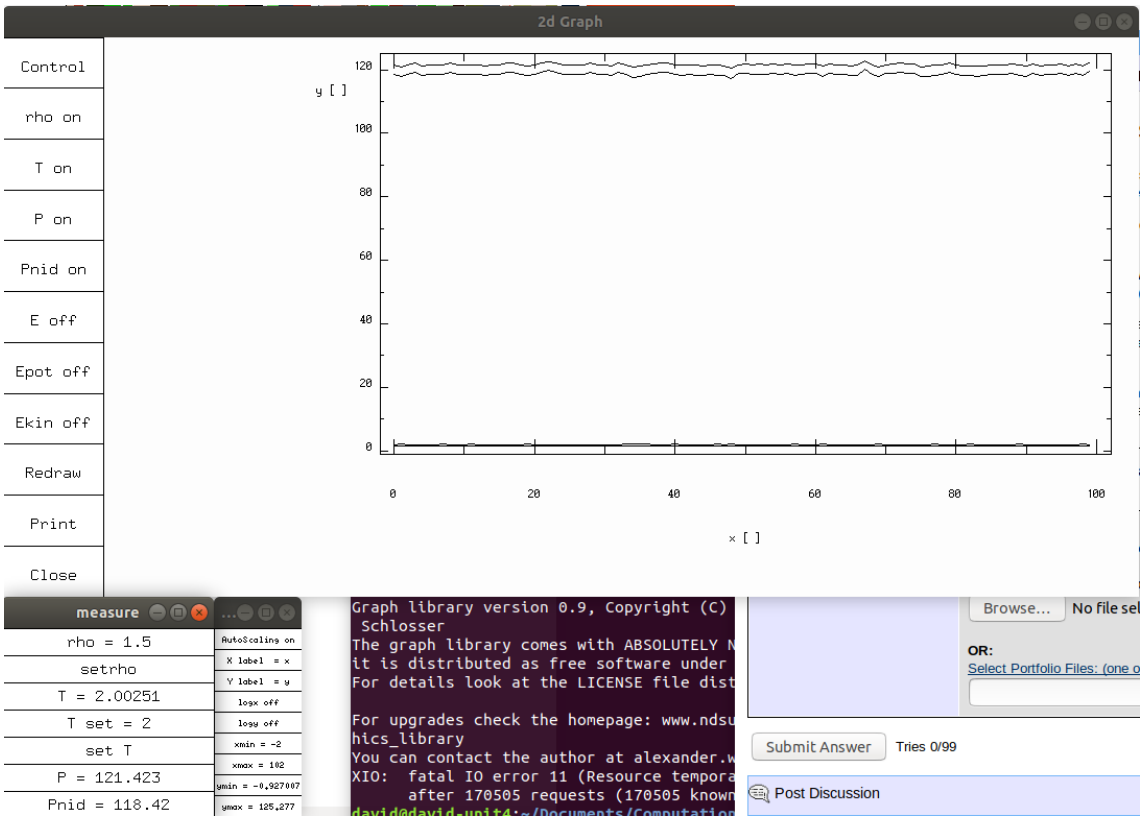

Figure 3: Final condition for T = 2

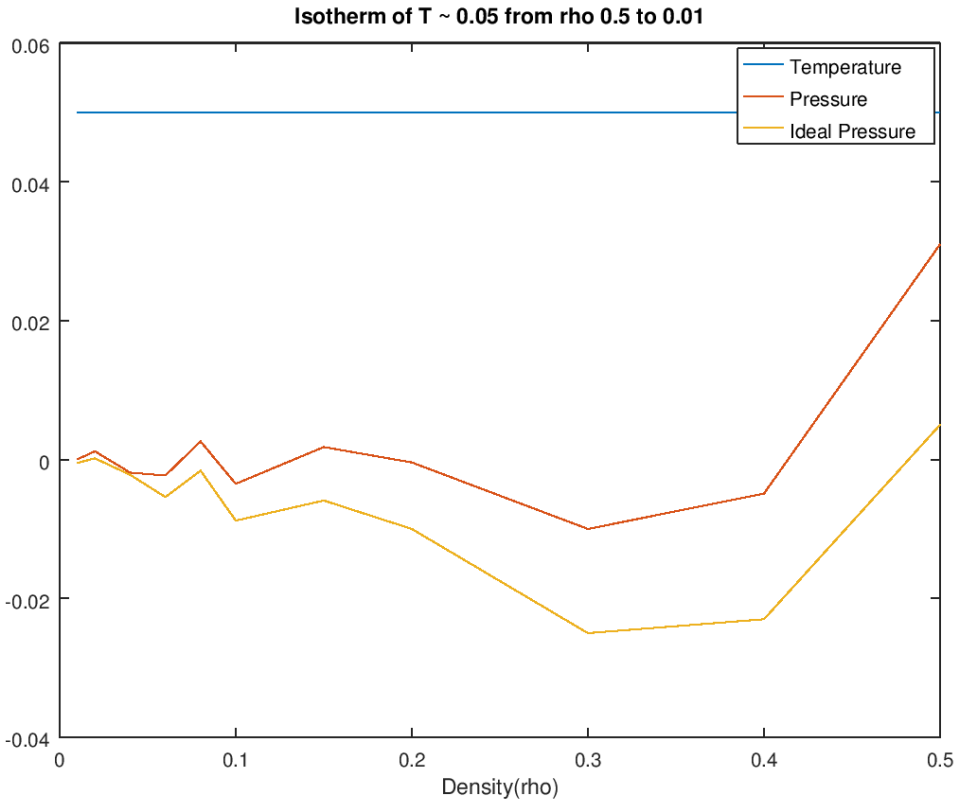
Isotherm of T ~ 0.05 from rho 0.5 to 0.01
Temperature
Pressure
Ideal Pressure
0.06
0.04
0.02
0
-0.02
-0.04
0
0.1
0.2
0.3
0.4
0.5
Density(rho)

Figure 4:

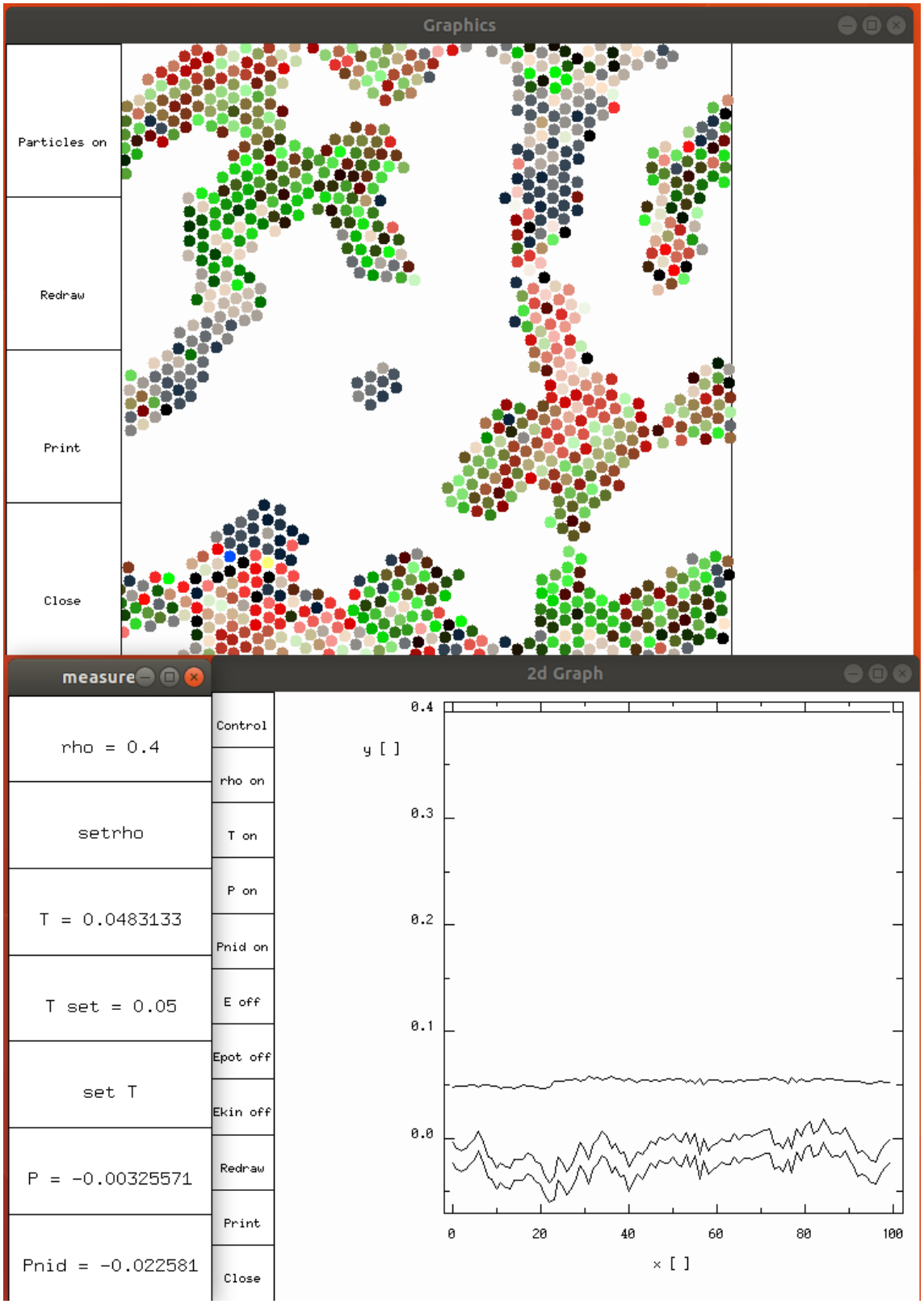

Figure 5: Initial condition for T = 0.05

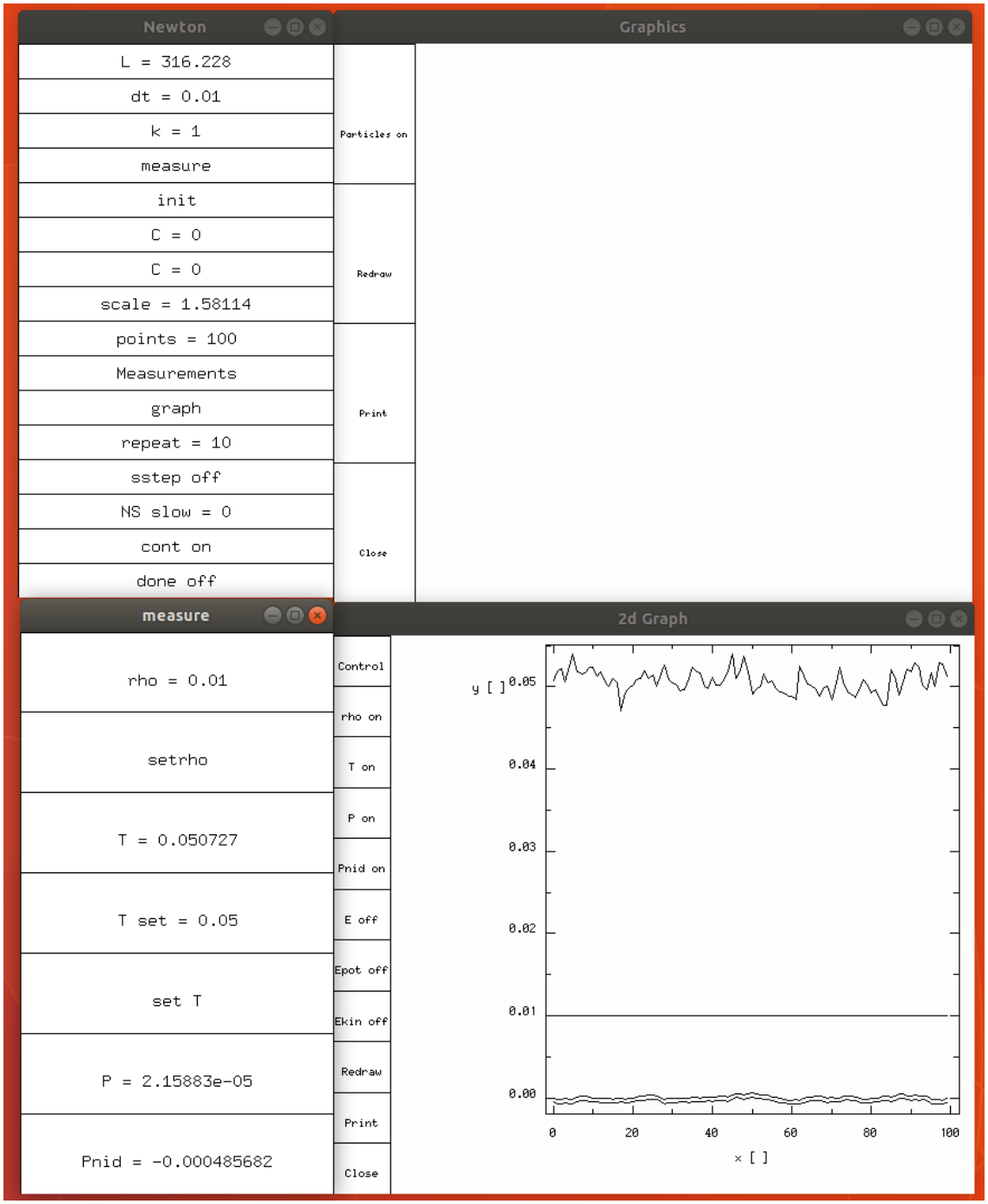

Figure 6: Final condition for T = 0.05

4 Conclusion

Then first isotherm's pressure and had very low error 5 percent, so it did follow the ideal gas law. This can be seen by looking at the difference between the P and Pnid. The second isotherm did not go through a phase change as intended. Perhaps a different starting temperature or density could make the phase change happen, but this configuration did not yield not desired result. Maybe the starting configuration could be determined by looking at the ideal gas law to predict what density and temperature isotherm could produce a phase change. Additionally, in figure 6 the particles are not easily visible, but they are still a solid or a liquid. The image needs to be zoomed in.

5 Simulation Code

```
#include <math.h>
#include <mygraph.h>
#include <unistd.h>
#include <string.h>
#include <time.h>

#define Nmax 1000
#define D 2
#define MeasMax 100
double  L=50; // size of box


double m[Nmax]; // charges of the particles
double x[Nmax][D],v[Nmax][D]; // State of the system

// parameters
double C[D],scalefac=100;
double k=1,x0[Nmax][D],v0[Nmax][D],dt=0.01,vv=1;
double rho[MeasMax],Tset=0,Tmeas[MeasMax], ppnid[MeasMax],pp[MeasMax],Etot[MeasMax],Epot[MeasMax],

int N=Nmax,Measlen=MeasMax,points=100,iterations=0;

void setdensity(){
  double fact=1/L;
  L=pow(N/rho[0],1./D);
  fact*=L;
  for (int n=0;n<N; n++)
    for (int d=0; d<D; d++)
      x[n][d]*=fact;
}

void setTemp(){
  double fact=sqrt(Tset/Tmeas[0]);
  for (int n=0;n<N; n++)
    for (int d=0; d<D; d++)
      v[n][d]*=fact;
}

double density(){
  double r=0;
  r=N;
  for (int d=0; d<D; d++) r/=L;
  return r;
}

double T(double v[N][D]){
  double t=0;
  for (int n=0; n<N; n++)
    for (int d=0; d<D; d++)
      t+=m[n]*v[n][d]*v[n][d];
  return t/N/D;
}

double Pnid(double x[N][D]){
  double p=0;
  for (int n=0; n<N; n++)
```

```
    for (int m=n+1; m<N; m++){
      double dr[D],dR=0;
      for (int d=0; d<D; d++){
        dr[d]=x[m][d]-(x[n][d]-L);
        double ddr;
        ddr=x[m][d]-(x[n][d]);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        ddr=x[m][d]-(x[n][d]+L);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        dR+=dr[d]*dr[d];
      }
      double dR6=dR*dR*dR;
      double dR12=dR6*dR6;
      double Fabs=12/dR12/dR-6/dR6/dR;
      p+=Fabs*dR/D;
    }
  for (int d=0; d<D; d++) p/=L;

  return p;
}

double Ep(double x[N][D],double v[N][D]){
  double EE=0;

  for (int n=0; n<N; n++)
    for (int m=n+1; m<N; m++){
      double dr[D];
      double dR=0;
      for (int d=0; d<D; d++){
        dr[d]=x[m][d]-(x[n][d]-L);
        double ddr;
        ddr=x[m][d]-(x[n][d]);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        ddr=x[m][d]-(x[n][d]+L);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        dR+=dr[d]*dr[d];
      }
      double dR6=dR*dR*dR;
      double dR12=dR6*dR6;
      EE += 1/dR12-1/dR6;
    }
  return 2*EE;
}

void F(double x[N][D], double v[N][D],double FF[N][D]){

  memset(&FF[0][0],0,N*D*sizeof(double));
  for (int n=0; n<N; n++)
    for (int m=n+1; m<N; m++){
      double dr[D],dR=0;
      for (int d=0; d<D; d++){
        dr[d]=x[m][d]-(x[n][d]-L);
        double ddr;
        ddr=x[m][d]-(x[n][d]);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        ddr=x[m][d]-(x[n][d]+L);
        if (fabs(ddr)<fabs(dr[d])) dr[d]=ddr;
        dR+=dr[d]*dr[d];
```

```
      }
      double dR6=dR*dR*dR;
      double dR12=dR6*dR6;
      double Fabs=12/dR12/dR-6/dR6/dR;
      for (int d=0;d<D; d++){
        FF[n][d]-=Fabs*dr[d];
        FF[m][d]+=Fabs*dr[d];
      }
    }
  return;
}

void iterate(double x[N][D],double v[N][D],double dt){
  double ff[N][D];
  F(x,v,ff);
  if (iterations==0)
    for (int n=0;n<N;n++)
      for (int d=0;d<D;d++)
        v[n][d]+=0.5*ff[n][d]/m[n]*dt;
  else
    for (int n=0;n<N;n++)
      for (int d=0;d<D;d++)
        v[n][d]+=ff[n][d]/m[n]*dt;

  for (int n=0;n<N;n++)
    for (int d=0;d<D;d++){
      x[n][d]+=v[n][d]*dt;
      if (x[n][d]<0) x[n][d]+=L;
      else if (x[n][d]>=L) x[n][d]-=L;
    }
  iterations++;
}

void CM(){
  double cm[D],cmv[D],M=0;
  memset(&cm[0],0,D*sizeof(double));
  memset(&cmv[0],0,D*sizeof(double));
  for (int n=0; n<N; n++) M+=m[n];
  for (int d=0; d<D; d++){
    for (int n=0; n<N; n++){
      cm[d]+=m[n]*x[n][d];
      cmv[d]+=m[n]*v[n][d];
    }
    cm[d]/=M;
    cmv[d]/=M;
  }
  for (int n=0; n<N; n++)
    for (int d=0; d<D; d++){
      x[n][d]-=cm[d]+L/2;
      v[n][d]-=cmv[d]+L/2;
    }

}

void setup(){
  int M=pow(N,1./D)+1;
  for (int n=0; n<N; n++){
    m[n]=1;
```

```
    for (int d=0; d<D; d++){
      int nn=n;
      for (int dd=0; dd<d; dd++) nn/=M;
      x0[n][d]=(nn%M)*L/M;
      if (d==1){
        if (x0[n][0]<L/2)
          v0[n][d]=vv;
        else v0[n][d]=-vv;
      }
      else v0[n][d]=0;
    }
  }
}

void init(){
  for (int n=0; n<N; n++){
    x[n][0]=x0[n][0];
    x[n][1]=x0[n][1];
    v[n][0]=v0[n][0];
    v[n][1]=v0[n][1];
  }
  iterations=0;
}

void draw(int xdim, int ydim){
  int size=xdim;
  if (ydim<size) size=ydim;
  scalefac=size/L;

  mydrawline(1,0,size,size,size);
  mydrawline(1,size,0,size,size);
  for (int n=0; n<N; n++){
    int xx=x[n][0]*scalefac;
    int yy=x[n][1]*scalefac;
    myfilledcircle(n+1,xx,size-yy,0.5*scalefac);
  }
}

void Measure(){
  memmove(&rho[1],&rho[0],(MeasMax-1)*sizeof(double));
  rho[0]=density();
  memmove(&Tmeas[1],&Tmeas[0],(MeasMax-1)*sizeof(double));
  Tmeas[0]=T(v);
  memmove(&ppnid[1],&ppnid[0],(MeasMax-1)*sizeof(double));
  ppnid[0]=Pnid(x);
  memmove(&pp[1],&pp[0],(MeasMax-1)*sizeof(double));
  pp[0]=rho[0]*Tmeas[0]+ppnid[0];
  memmove(&Ekin[1],&Ekin[0],(MeasMax-1)*sizeof(double));
  Ekin[0]=N*D*Tmeas[0];
  memmove(&Epot[1],&Epot[0],(MeasMax-1)*sizeof(double));
  Epot[0]=Ep(x,v);
  memmove(&Etot[1],&Etot[0],(MeasMax-1)*sizeof(double));
  Etot[0]=Epot[0]+Ekin[0];
}

int main(){
  struct timespec ts={0,0};
  int cont=0;
```

```
int sstep=0;
int repeat=10;
int done=0;
char name[50],mname[N][50];
setup();
init();
Measure();

DefineGraphN_R("rho",&rho[0],&Measlen,NULL);
DefineGraphN_R("T",&Tmeas[0],&Measlen,NULL);
DefineGraphN_R("P",&pp[0],&Measlen,NULL);
DefineGraphN_R("Pnid",&ppnid[0],&Measlen,NULL);
DefineGraphN_R("E",&Etot[0],&Measlen,NULL);
DefineGraphN_R("Epot",&Epot[0],&Measlen,NULL);
DefineGraphN_R("Ekin",&Ekin[0],&Measlen,NULL);

AddFreedraw("Particles",&draw);
StartMenu("Newton",1);
DefineDouble("L",&L);
DefineDouble("dt",&dt);
DefineDouble("k",&k);
StartMenu("measure",0);
DefineDouble("rho",&rho[0]);
DefineFunction("setrho",setdensity);
DefineDouble("T",&Tmeas[0]);
DefineDouble("T set",&Tset);
DefineFunction("set T",setTemp);
DefineDouble("P",&pp[0]);
DefineDouble("Pnid",&ppnid[0]);
EndMenu();
StartMenu("init",0);
for (int n=0; n<N; n++){
}
if (N<15)
  for (int n=0; n<N; n++){
    sprintf(mname[n],"Particle %i",n);
    StartMenu(mname[n],0);
    DefineDouble("m",&m[n]);
    for (int d=0; d<D; d++){
      sprintf(name,"x[%i]",d);
      DefineDouble(name,&x0[n][d]);
    }
    for (int d=0; d<D; d++){
      sprintf(name,"v[%i]",d);
      DefineDouble(name,&v0[n][d]);
    }
    EndMenu();
  }
DefineDouble("vv",&vv);
DefineFunction("setup",&setup);
DefineFunction("init",&init);
DefineFunction("CM",&CM);
EndMenu();
for (int d=0; d<D; d++){
  DefineDouble("C",&C[d]);
}
DefineDouble("scale",&scalefac);
DefineInt("points",&points);
```

```
  DefineGraph(curve2d_,"Measurements");
  DefineGraph(freedraw_,"graph");
  DefineInt("repeat",&repeat);
  DefineBool("sstep",&sstep);
  DefineLong("NS slow",&ts.tv_nsec);
  DefineBool("cont",&cont);
  DefineBool("done",&done);
  EndMenu();
  while (!done){
    Events(1);
    DrawGraphs();
    if (cont||sstep){
      sstep=0;
      for (int i=0; i<repeat; i++) iterate(x,v,dt);
      nanosleep(&ts,NULL);
      Measure();
    }
    else sleep(1);
  }
}
```

6 Plotting Code

```
% test 1 for T >1

T =    [2.04,2.01,1.99,1.97,2   ,2.02 ,1.98 ,2.07  ,2     ,2];
P =    [2.1 ,3   ,4.5 ,6.6 ,9.47,15.16,22.83,33.57 ,49.366,121.423];
Pnid = [1.05,1.8 ,3.2 ,5.1 ,7.66,13.14,20.65,31.075,46.77 ,118.42];
rho =  [0.5 ,0.6 ,0.7 ,0.8 ,0.9 ,1.0  ,1.1  ,1.2   ,1.3   ,1.4];
% T variation = 0.2
% P variation = 0.1
% Pnid variation = 0.1

plot(rho,T,rho,P,rho,Pnid);
%itle("Isotherm of T ~ 2 from rho 0.5 to 1.4");
xlabel ("Density(rho)");
legend ("Temperature","Pressure","Ideal Pressure");


% test 2

T =    [0.05 ,0.05   ,0.05 ,0.05   ,0.05   ,0.05   ,0.05  ,0.05   ,0.05    ,0.05    ,0.05];
P =    [0.031,-0.0049,-0.01,-.00042,0.0018   ,-0.0035,0.0026,-0.0023,-0.00189,0.001186,2.15*10^-5]
Pnid = [0.005,-0.023 ,-0.025,-0.01 ,-0.0059,-0.0088,-0.0016,-0.0054,-0.0022 ,0.00017,-0.000485];
rho =  [0.5 ,0.4     ,0.3   ,0.2   ,0.15   ,0.1    ,0.08   ,0.06   ,.04    ,0.02     ,   0.01];
% T variation = 0.2
% P variation = 0.1
% Pnid variation = 0.1
figure;
plot(rho,T,rho,P,rho,Pnid);
title("Isotherm of T ~ 0.05 from rho 0.5 to 0.01");
xlabel ("Density(rho)");
legend ("Temperature","Pressure","Ideal Pressure");
```

References

https://www.ndsu.edu/pubweb/~carswagn/LectureNotes/370/index.html