

Apéndice A

Código fuente

Las dos simulaciones fueron hechas en el lenguaje de programación de C++, a continuación se exhibirán sus respectivos códigos fuente.

A.1. Simulación gas ideal

```
//Gas Ideal Verlet Optimizado
#include <iostream>
#include<fstream>
#include <cmath>
#include "Vector.h"
#include "random.h"
using namespace std;

ofstream c("GasIdeal.plt");

const double Deltat=0.01;
const double chi=0.193183325037836;

const double g=0,T=700;
const double K=82, Gamma=0, Kcundall=0, mu=0.0, Bm=1.38e-23, mm=6.69e-26;

const double Lx=100,Ly=100;
const int Nx=6,Ny=6,N=Nx*Ny;

enum Restriccion{ninguna,posicion,velocidad,fuerza};

class Cuerpo;
class Colisionador;
//-----class Cuerpo -----
class Cuerpo{
private:
    double m,R; vector3D r,V,F; double theta,omega,tau,I; //
    Restriccion RestriccionX,RestriccionY,RestriccionTheta; double ValorX,ValorY,Valor
public:
```

```

void Inicie(double x0,double y0,double Vx0,double Vy0,double theta0,double omega0,d
void Mueva_r1(double dt); //
void Mueva_V(double dt); //
void Mueva_r2(double dt); //
void Dibujese(void); //

void Fijex(double x0){RestriccionX=posicion; ValorX=x0;};
void Fijey(double y0){RestriccionY=posicion; ValorY=y0;};
void Fijetheta(double theta0){RestriccionTheta=posicion; ValorTheta=theta0;};
void FijeVx(double Vx0){RestriccionX=velocidad; ValorX=Vx0;};
void FijeVy(double Vy0){RestriccionY=velocidad; ValorY=Vy0;};
void Fijeomega(double omega0){RestriccionTheta=velocidad; ValorTheta=omega0;};
void FijeFx(double Fx0){RestriccionX=fuerza; ValorX=Fx0;};
void FijeFy(double Fy0){RestriccionY=fuerza; ValorY=Fy0;};
void Fijetau(double tau0){RestriccionTheta=fuerza; ValorTheta=tau0;};

double Getx(void){return r.x();};
double Gety(void){return r.y();};
double Getm(void){return m;};
double GetEc(void){return m*norma2(V)/2+I*omega*omega/2;};
double GetV(void){return norma(V);};
double GetVx(void){return V.x();};
friend class Colisionador;
};
void Cuerpo::Inicie(double x0,double y0,double Vx0,double Vy0,double theta0,double om
    r.cargue(x0,y0,0); V.cargue(Vx0,Vy0,0); m=m0; R=R0; theta=theta0; omega=omega0; I=2
    RestriccionX=RestriccionY=RestriccionTheta=ninguna;          ValorX=ValorY=ValorTheta=
}
void Cuerpo::Mueva_r1(double dt){ //
    r+=V*chi*dt;          theta+=omega*chi*dt;
    if(RestriccionX==posicion) r.v[0]=ValorX;
    if(RestriccionY==posicion) r.v[1]=ValorY;
    if(RestriccionTheta==posicion) theta=ValorTheta;
}
void Cuerpo::Mueva_V(double dt){ //
    V+=F*(dt/(2*m));          omega+=tau/I*dt/2;
    if(RestriccionX==velocidad) V.v[0]=ValorX;
    if(RestriccionY==velocidad) V.v[1]=ValorY;
    if(RestriccionTheta==velocidad) omega=ValorTheta;
}
void Cuerpo::Mueva_r2(double dt){ //
    r+=V*((1-2*chi)*dt); theta+=omega*(1-2*chi)*dt;
    if(RestriccionX==posicion) r.v[0]=ValorX;
    if(RestriccionY==posicion) r.v[1]=ValorY;
    if(RestriccionTheta==posicion) theta=ValorTheta;
}
void Cuerpo::Dibujese(void){ //
    c<<" , "<<r.x()<<"+"<<R<<"*cos(t),"<<r.y()<<"+"<<R<<"*sin(t) , "

```

```

    <<r.x()<<"+"<<R*cos(theta)/7.0<<"*t,"<<r.y()<<"+"<<R*sin(theta)/7.0<<"*t";
}
//-----class Colisionador -----
class Colisionador{ //
private:
    double dcontacto[N+4][N+4],hold[N+4][N+4];
public:
    void Inicie(void);
    void CalculeFuerzas(Cuerpo *Grano,double dt);
    void Choque(Cuerpo &Cuerpo1,Cuerpo &Cuerpo2,double &dcontacto,double &hold,double &h,
    double GetEp(void);
};
double Colisionador::GetEp(void){
    double Ep=0,h;
    for(int i=0;i<N;i++){
        for(int j=i+1;j<N+4;j++){
            Ep+=Kcundall*pow(dcontacto[i][j],2)/2;
            h=hold[i][j]; if(h>0) Ep+K*pow(h,3.5)/3.5;
        }
    }
    return Ep;
}
void Colisionador::Inicie(void){ //Define los choques entre las particulas y entre las
    for(int i=0;i<N+4;i++)
        for(int j=0;j<N+4;j++)
            dcontacto[i][j]=hold[i][j]=0;
}
void Colisionador::CalculeFuerzas(Cuerpo *Grano,double dt){ //Calcula las fuerzas de
    int i,j;
    for(i=0;i<N+4;i++){
        {Grano[i].F.cargue(0,0,0); Grano[i].tau=0;}
    }
    for(i=0;i<N;i++)
        for(j=i+1;j<N+4;j++)
            Choque(Grano[i],Grano[j],dcontacto[i][j],hold[i][j],dt);
}
void Colisionador::Choque(Cuerpo &Cuerpo1,Cuerpo &Cuerpo2,double &dcontacto,double &h,
    vector3D r12,V12,Vc,n,Rw,t; double h,R1,R2,m1,m2,m12,Vn,Vt,Fn,Ft,Ftmax;
    R1=Cuerpo1.R; R2=Cuerpo2.R;
    r12=Cuerpo1.r-Cuerpo2.r;
    h=R1+R2-norma(r12);

    if(h>0){
        V12=Cuerpo1.V-Cuerpo2.V;
        n=r12/norma(r12); Rw.cargue(0,0,R1*Cuerpo1.omega+R2*Cuerpo2.omega);
        Vc=V12-(Rw^n); Vn=Vc*n; t.cargue(n.y(),-n.x(),0); Vt=Vc*t;

        m1=Cuerpo1.m; m2=Cuerpo2.m; m12=m1*m2/(m1+m2);
        Fn=K*pow(h,1.5)-m12*sqrt(h)*Gamma*Vn;
    }
}

```

```

dcontacto+=Vt*dt;
Ft=-Kcundall*dcontacto;
if(fabs(Ft)>(Ftmax=mu*fabs(Fn))) Ft=Ft/fabs(Ft)*Ftmax;

Cuerpo1.F+=n*Fn;    Cuerpo1.F+=t*Ft;    Cuerpo1.tau+=R1*Ft;
Cuerpo2.F-=n*Fn;    Cuerpo2.F-=t*Ft;    Cuerpo2.tau+=R2*Ft;

if(Cuerpo1.RestriccionX==fuerza) Cuerpo1.F.v[0]=Cuerpo1.ValorX;
if(Cuerpo1.RestriccionY==fuerza) Cuerpo1.F.v[1]=Cuerpo1.ValorY;
if(Cuerpo1.RestriccionTheta==fuerza) Cuerpo1.tau=Cuerpo1.ValorTheta;
if(Cuerpo2.RestriccionX==fuerza) Cuerpo2.F.v[0]=Cuerpo1.ValorX; //
if(Cuerpo2.RestriccionY==fuerza) Cuerpo2.F.v[1]=Cuerpo1.ValorY;
if(Cuerpo2.RestriccionTheta==fuerza) Cuerpo2.tau=Cuerpo1.ValorTheta;
}

if(hold>0 && h<0) dcontacto=0;

hold=h;

}

//-----Funciones de Animacion -----

void InicieAnimacion(void){ //
  //c<<"set terminal gif animate"<<endl;
  //c<<"set output 'GasIdeal.gif'"<<endl;
  c<<"unset key"<<endl;
  c<<"set xrange [0:"<<Lx<<"]"<<endl;
  c<<"set yrange [0:"<<Ly<<"]"<<endl;
  c<<"set size ratio -1"<<endl;
  c<<"set parametric"<<endl;
  c<<"set trange [0:7]"<<endl;
  c<<"set isosamples 12"<<endl;
}
void InicieCuadro(void){
  c<<"plot 0,0 ";
}
void TermineCuadro(void){
  c<<endl;
}

int main(){
ofstream z("vmean.txt");
ofstream a("hgi.txt");
ofstream b("hgi.plt");
b<<"set yrange[0:36]"<<endl;
b<<"plot 'hgi.txt' with boxes"<<endl;
  Cuerpo Grano[N+4];

```

```

Colisionador Cundall;
Crandom Ran2(0);
double t,tdibujo,dx,dy,Rmax,R,m,Alpha,Ec,Ep, VELO, P;
VELO=sqrt(2*Bm*T/mm)/1.57e2;
clog<<"VELO = "<<sqrt(2*Bm*T/mm)/1.57e2<<endl;
P=(N*Bm*T/(Lx*Ly))/1.43e-2;
clog<<"Presion = "<<(N*Bm*T/Lx*Ly)/1.43e-2<<endl;
double vmean;
int i,j,v;

InicieAnimacion();
Cundall.Inicie();
//-----(x0 ,y0,Vx0,Vy0,theta0,omega0,m0,R0);
//PAREDES
//Pared izquierda
Grano[N].Inicie(-10000, Ly/2, 0, 0, 0, 0, 100000, 10000);
Grano[N].Fijex(-10000); Grano[N].Fijey(Ly/2); Grano[N].Fijetheta(0);
//Pared derecha
Grano[N+1].Inicie(Lx+10000, Ly/2, 0, 0, 0, 0, 100000, 10000);
Grano[N+1].Fijex(Lx+10000); Grano[N+1].Fijey(Ly/2); Grano[N+1].Fijetheta(0);
//Pared abajo
Grano[N+2].Inicie(Lx/2, -10000, 0, 0, 0, 0, 100000, 10000);
Grano[N+2].Fijex(Lx/2); Grano[N+2].Fijey(-10000); Grano[N+2].Fijetheta(0);
//Pared arriba
Grano[N+3].Inicie(Lx/2, Ly+10000, 0, 0, 0, 0, 100000, 10000);
Grano[N+3].Fijex(Lx/2); Grano[N+3].Fijey(Ly+10000); Grano[N+3].Fijetheta(0);
//GRANOS
dx=Lx/(Nx+1); dy=Ly/(Ny+1); Rmax=dx/3; if(dx/3 > dy/3) Rmax=dy/3;
for(i=0; i<N; i++){
    Alpha=2*M_PI*Ran2.r(); R=Rmax*(1+Ran2.r())/2; m=pow(R/Rmax, 3);
    Grano[i].Inicie(((i/Nx)+1)*dx, ((i/Ny)+1)*dy, VELO*cos(Alpha), VELO*sin(Alpha), 0, 0, 1
}

cout<<endl;
cout<<Grano[0].GetVx()<<endl;

//RELAJACION
for(t=tdibujo=0; t<60; t+=Deltat, tdibujo+=Deltat){
    if(tdibujo>1/6.0){
        InicieCuadro();
        for(i=0; i<N; i++) Grano[i].Dibujese();
        TermineCuadro();
        tdibujo=0;
        vmean=0;
        for(i=0; i<N; i++) vmean+=Grano[i].GetV();
        z<<t<<" "<<vmean/N<<endl;
    }
}

```

```

//Velocidad Verlet Optimizado
for(i=0;i<N;i++) Grano[i].Mueva_r1(Deltat);
Cundall.CalculeFuerzas(Grano,0.5*Deltat);
for(i=0;i<N;i++) {Grano[i].Mueva_V(Deltat);Grano[i].Mueva_r2(Deltat);}
Cundall.CalculeFuerzas(Grano,0.5*Deltat);
for(i=0;i<N;i++) {Grano[i].Mueva_V(Deltat);Grano[i].Mueva_r1(Deltat);}
}
cout<<Grano[0].GetVx()<<endl;
int np[6]={0}; //Nmero de puntos.
double Vmax=0; //Definicin de la mxima velocidad.
for(i=0;i<N;i++){
    if(Grano[i].GetV()>Vmax)Vmax=Grano[i].GetV();
}

v=int(Vmax);
for(j=0;j<N;j++){
for(i=0;i<6;i++){
if(Grano[j].GetV()<3*(i+1)) {np[i]+=1; break;}
}
}
for(i=0;i<6;i++) a<<3*(i+1)-1.5<<" "<<np[i]<<endl;
system("pause");
return 0;
}

```

A.2. Gas bajo interacciones moleculares

```

//Lennard-Jones Verlet Optimizado
#include <iostream>
#include<fstream>
#include <cmath>
#include "Vector.h"
#include "random.h"
using namespace std;

ofstream c("LJ.plt");

const double Deltat=0.01;
const double chi=0.193183325037836;
const double T=900;
const double Bm=1.38e-23, mm=6.69e-26;

const double Lx=100,Ly=100;
const int Nx=6,Ny=6,N=Nx*Ny;
const double epsilon=2*M_PI/N;

class Cuerpo;
class Colisionador;

```

```
//-----class Cuerpo -----
class Cuerpo{
private:
    double m,R; vector3D r,V,F;    double theta,omega,tau,I;
public:
    void Inicie(double x0,double y0,double Vx0,double Vy0,double theta0,double omega0,d
    void Mueva_r1(double dt);
    void Mueva_V(double dt);
    void Mueva_r2(double dt);
    void Dibujese(void);

    double Getx(void){return r.x();};
    double Gety(void){return r.y();};
    double Getm(void){return m;};
    double GetEc(void){return m*norma2(V)/2+I*omega*omega/2;};
    double GetV(void){return norma(V);};
    double GetVx(void){return V.x();};
    friend class Colisionador;
};
void Cuerpo::Inicie(double x0,double y0,double Vx0,double Vy0,double theta0,double om
    r.cargue(x0,y0,0); V.cargue(Vx0,Vy0,0); m=m0; R=R0; theta=theta0; omega=omega0; I=2
}
void Cuerpo::Mueva_r1(double dt){
    r+=V*chi*dt;          theta+=omega*chi*dt;
}
void Cuerpo::Mueva_V(double dt){
    V+=F*(dt/(2*m));      omega+=tau/I*dt/2;
}
void Cuerpo::Mueva_r2(double dt){
    r+=V*((1-2*chi)*dt);  theta+=omega*(1-2*chi)*dt;
}
void Cuerpo::Dibujese(void){
    c<<" , "<<r.x()<<"+"<<R<<"*cos(t),"<<r.y()<<"+"<<R<<"*sin(t) , "
        <<r.x()<<"+"<<R*cos(theta)/7.0<<"*t,"<<r.y()<<"+"<<R*sin(theta)/7.0<<"*t";
}
//-----class Colisionador -----
class Colisionador{
private:
    double dcontacto[N+4] [N+4],hold[N+4] [N+4],pressure;
public:
    void Inicie(void);
    void CalculeFuerzas(Cuerpo *Grano,double dt);
    void Choque(Cuerpo &Cuerpo1,Cuerpo &Cuerpo2,double &dcontacto,double &hold,double d
    double GetVirial(Cuerpo *Grano);
    double GetPressure(void){return pressure;};
};
double Colisionador::GetVirial(Cuerpo *Grano){
    double virial;
```

```

    virial=0.0;
    vector3D Rij, Normal, Fij;
    double normRij;
    int i,j;
    for(i=0;i<N-1;i++){
        for(j=i+1;j<N;j++){
            Rij = Grano[i].r - Grano[j].r;
            normRij = norma(Rij);
            Normal = Rij/normRij;
            Fij=(24*Normal/normRij)*(2/pow(normRij,13)-1/pow(normRij,7)); //Aqui va el
            virial+=Rij*Fij;
        }
    }
    return 0.5*virial;
}

void Colisionador::Inicie(void){ //
    pressure=0;
    for(int i=0;i<N+4;i++)
        for(int j=0;j<N+4;j++)
            dcontacto[i][j]=hold[i][j]=0;
}

void Colisionador::CalculeFuerzas(Cuerpo *Grano,double dt){ //
    vector3D Rij, Normal;
    double normRij;
    int i,j;
    for(i=0;i<N+4;i++) {Grano[i].F.cargue(0,0,0); Grano[i].tau=0;}
    for(i=0;i<N-1;i++){
        for(j=i+1;j<N;j++){
            Rij = Grano[i].r - Grano[j].r;
            normRij = norma(Rij);
            Normal = Rij/normRij;
            Grano[i].F+=(24*Normal/normRij)*(2/pow(normRij,13)-1/pow(normRij,7)); //Aqui va el
            Grano[j].F-=Grano[i].F;
        }
    }
    for(i=0;i<N;i++)
        for(j=N;j<N+4;j++)
            Choque(Grano[i],Grano[j],dcontacto[i][j],hold[i][j],dt);
}

void Colisionador::Choque(Cuerpo &Cuerpo1,Cuerpo &Cuerpo2,double &dcontacto,double &h
    double hx,hy,Vx,Vy;
    double mass;
    mass=Cuerpo1.m;
    hx=Cuerpo1.r.x();
    hy=Cuerpo1.r.y();
    Vx=Cuerpo1.V.x();
    Vy=Cuerpo1.V.y();

```

```

    if(hx<0.5){
        Cuerpo1.V.cargue(-Vx,Vy,0);
        Cuerpo1.r.cargue(1.0-hx,hy,0);
        pressure+=2*mass*fabs(Vx);
    }
    if(hx>(Lx-0.5)){
        Cuerpo1.V.cargue(-Vx,Vy,0);
        Cuerpo1.r.cargue(2*Lx-1.0-hx,hy,0);
        pressure+=2*mass*fabs(Vx);
    }
    if(hy<0.5){
        Cuerpo1.V.cargue(Vx,-Vy,0);
        Cuerpo1.r.cargue(hx,1.0-hy,0);
        pressure+=2*mass*fabs(Vy);
    }
    if(hy>(Ly-0.5)){
        Cuerpo1.V.cargue(Vx,-Vy,0);
        Cuerpo1.r.cargue(hx,2*Ly-1.0-hy,0);
        pressure+=2*mass*fabs(Vy);
    }
}
//-----Funciones de Animacion -----

void InicieAnimacion(void){
    //c<<"set terminal gif animate"<<endl;
    //c<<"set output 'GasIdeal.gif'"<<endl;
    c<<"unset key"<<endl;
    c<<"set xrange [0:"<<Lx<<]"<<endl;
    c<<"set yrange [0:"<<Ly<<]"<<endl;
    c<<"set size ratio -1"<<endl;
    c<<"set parametric"<<endl;
    c<<"set trange [0:7]"<<endl;
    c<<"set isosamples 12"<<endl;
}
void InicieCuadro(void){
    c<<"plot 0,0 ";
}
void TermineCuadro(void){
    c<<endl;
}

int main(){
    ofstream z("datos.txt");
    ofstream a("histograma.txt");
    ofstream b("hglj.plt");
    b<<"set yrange[0:36]"<<endl;
    b<<"plot 'histograma.txt' with boxes"<<endl;
    Cuerpo Grano[N+4];

```

```

Colisionador Cundall;
Crandom Ran2(0);
double t,tdibujado, dx, dy, Rmax, R, m, Ec, Ep, VELO;
double Alpha[N], vx[N], vy[N], vxsum=0, vysum=0, vxCM, vyCM;
VELO=sqrt(2*Bm*T/mm)/1.57e2;
clog<<"VELO = "<<sqrt(2*Bm*T/mm)/1.57e2<<endl;
double vmean;
int i,j,v;

InicieAnimacion();
Cundall.Inicie();
//-----(x0 ,y0,Vx0,Vy0,theta0,omega0,m0,R0);
//PAREDES
//Pared izquierda
Grano[N].Inicie(-10000,Ly/2,0,0,0,0,100000,10000);
//Pared derecha
Grano[N+1].Inicie(Lx+10000,Ly/2,0,0,0,0,100000,10000);
//Pared abajo
Grano[N+2].Inicie(Lx/2,-10000,0,0,0,0,100000,10000);
//Pared arriba
Grano[N+3].Inicie(Lx/2,Ly+10000,0,0,0,0,100000,10000);
//GRANOS
dx=Lx/(Nx+1); dy=Ly/(Ny+1); Rmax=dx/3; if(dx/3 > dy/3) Rmax=dy/3;
for(i=0;i<N;i++){
    Alpha[i]=2*M_PI*Ran2.r();
    vx[i]=VELO*cos(Alpha[i]); vy[i]=VELO*sin(Alpha[i]);
    vxsum+=vx[i]; vysum+=vy[i];
}
vxCM=vxsum/N; vyCM=vysum/N;
cout<< vxCM<<" "<<vyCM<<endl;
for(i=0;i<N;i++){
    vx[i]=vx[i]-vxCM; vy[i]=vy[i]-vyCM;
    Grano[i].Inicie(((i/Nx)+1)*dx,((i/Ny)+1)*dy, vx[i], vy[i], 0, 0, 1, 0.5);
}
cout<<endl;
//RELAJACION
for(t=tdibujado=0;t<60;t+=Deltat,tdibujado+=Deltat){
    if(tdibujado>1/6.0){
        InicieCuadro();
        for(i=0;i<N;i++) Grano[i].Dibujese();
        TermineCuadro();
        tdibujado=0;
        vmean=0;
        for(i=0;i<N;i++) vmean+=Grano[i].GetV();
        z<<t<<" "<<vmean/N<<" "<<Cundall.GetVirial(Grano)<<" "<<Cundall.GetPressu
    }
    //Velocidad Verlet Optimizado
    for(i=0;i<N;i++) Grano[i].Mueva_r1(Deltat);
}

```

```
Cundall.CalculeFuerzas(Grano,0.5*Deltat);
for(i=0;i<N;i++) {Grano[i].Mueva_V(Deltat);Grano[i].Mueva_r2(Deltat);}
Cundall.CalculeFuerzas(Grano,0.5*Deltat);
for(i=0;i<N;i++) {Grano[i].Mueva_V(Deltat);Grano[i].Mueva_r1(Deltat);}
}
int np[6]={0}; //Nmero de puntos.
double Vmax=0; //Definicin de la mxima velocidad.
for(i=0;i<N;i++){
    if(Grano[i].GetV(>Vmax)Vmax=Grano[i].GetV();
}
v=int(Vmax);
for(j=0;j<N;j++){
    for(i=0;i<6;i++){
        if(Grano[j].GetV(<3*(i+1)) {np[i]+=1; break;}
}
}
for(i=0;i<6;i++) a<<3*(i+1)-1.5<<" "<<np[i]<<endl;
cout<<"Listo!"<<endl;
    cin>>v;
    return 0;
}
```