

Introducing Dynamic Walls into Integer Lattice Gas Simulations

David Jedynak

NDSU

May 10th, 2018

Introduction

- ▶ Explore interactions between gas and rigid shapes
- ▶ Build off of existing Lattice Gas Simulation Code

initial goals

- ▶ Non leaking dynamic walls
- ▶ Make complex shapes out of these dynamic walls
- ▶ Reproduce the Feynman tube experiment.

Method 1

Expected value of flow

$$\langle flow \rangle = \text{particle density} * \text{wall velocity}$$

$$0 < flow < \text{min particle density}$$

```

    int tmp_0 = n[vx+x][y+vy][8-v];
int tmp = n[vx+x][y+vy][v];
    int max_random = 1;

    if(tmp > tmp_0){ max_random = tmp_0;}

    else{ max_random = tmp;}

    if(max_random > 0){ flow = (rand()%max_random);

    else{ flow = 0;}

n[x+vx][((y+vy))][v] = n[x][y][8-v] - flow;
n[x][y][8-v] = tmp + flow;

```

Results

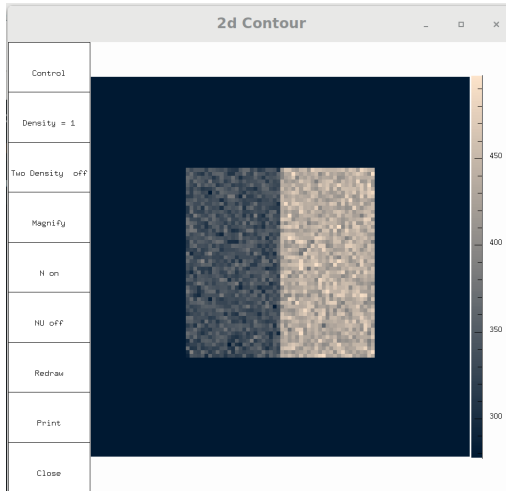


Figure 1:

Results

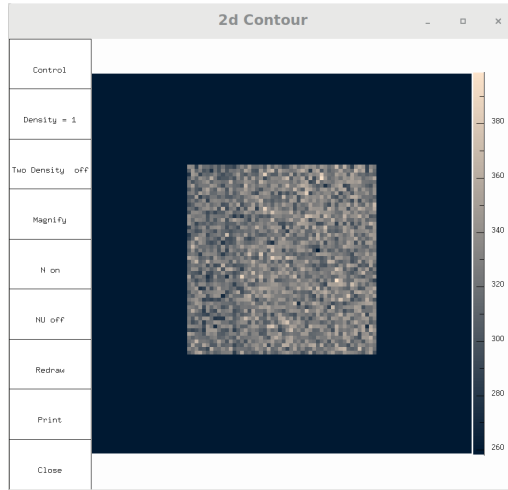


Figure 2:

Results

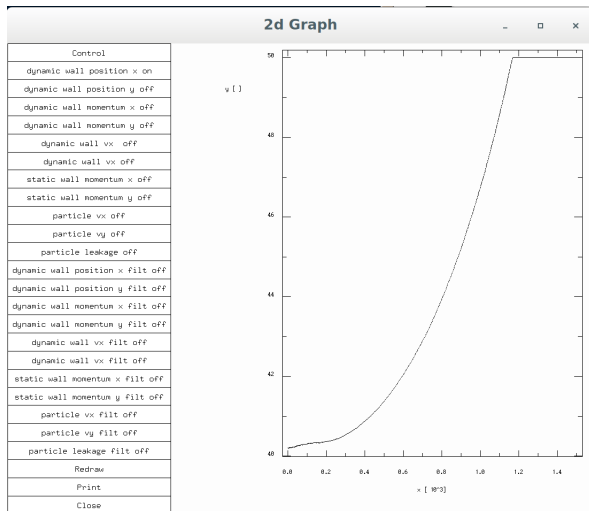


Figure 3:

Results

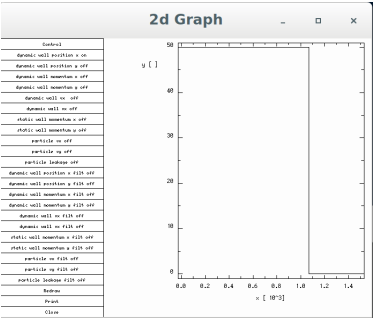


Figure 4:

jtextj

Results

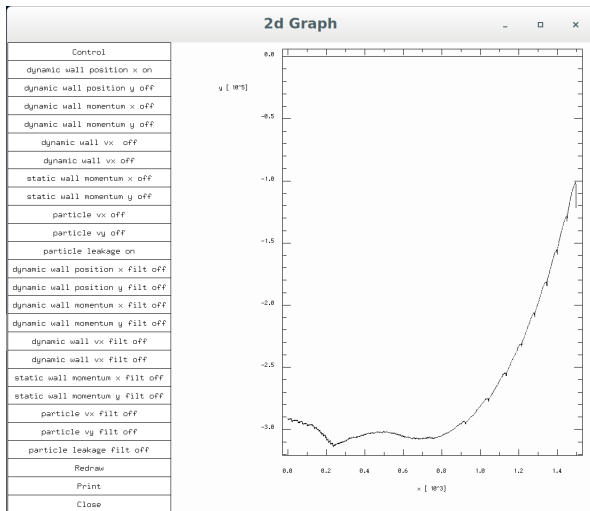


Figure 5:

Method 2

In more detail, the probability that

$$pr * \text{particle density}$$

number of particles will be moved is

$$pr = \frac{\text{Wall } V_x}{1 - (\text{real}(\text{Wall } x) - \text{int}(\text{Wall } x))}$$

.

```

int tmp = n[x_v_b][y_v_b][v];
int iwp = dynamic_wall_position_x;//integer wall po
int flow = 0;//particles to be moved
double pr = dynamic_wall_vx/(1-(dynamic_wall_positio
if(rand()%1000 <= 1000*pr){
    flow = pr*n[x_b][y_b][v];
    n[x_b+1][y_b][v] +=flow;
    n[x_b][y_b][v] -=flow;}

n[x_v_b][y_v_b][v] = n[x_b][y_b][8-v];
n[x_b][y_b][8-v] = tmp;// + flow;

```

Results

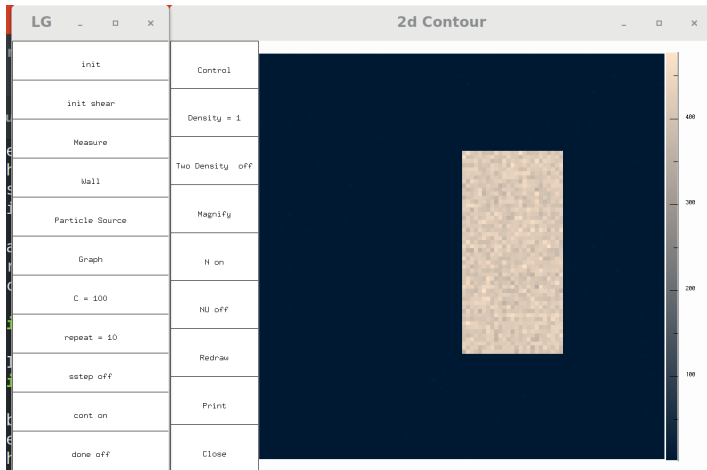


Figure 6:

Results

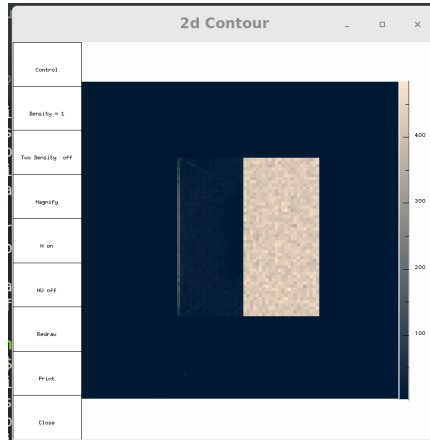


Figure 7:

Results

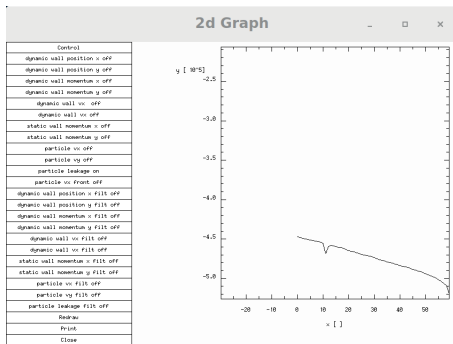


Figure 8:

Results

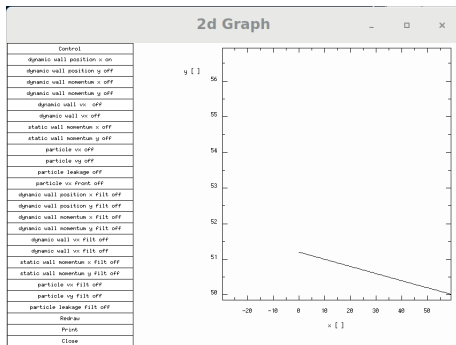


Figure 9:

Method 3

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = \nabla(\rho) + v * \nabla(\nabla(U) + (\nabla(U)T)) \quad (1)$$

The partial for ρ and ρu_i can be set to zero. This gives us:

$$0 = \nabla(\rho) + v * \nabla(\nabla(U) + (\nabla(U)T)) \quad (2)$$

$$\nabla(\rho) = F$$

$$0 = F + v * \nabla(\nabla(U_x)) \quad (3)$$

Solving the differential equation for

$$U_x$$

(mean velocity) above gives us:

$$U_x = \frac{F}{2 * v} * (x(x-L)) \text{ Where } L \text{ is the length of the tube in Lattice sites} \quad (4)$$

```
void moveParticles(){
for(int x = 0;x<xdim;x++){
    for(int y = 0;y<ydim;y++){
        int flip_parts = particle_flip_w*
        ((double)rand())/RAND_MAX)*n[x][y][5];
        n[x][y][3] += flip_parts;
        n[x][y][5] -= flip_parts;
        flip_parts = particle_flip_w*
        ((double)rand())/((double)RAND_MAX)*n[x][y][
        n[x][y][6] += flip_parts;
        n[x][y][8] -= flip_parts;
        flip_parts = particle_flip_w*
        ((double)rand())/((double)RAND_MAX)*n[x][y][
        n[x][y][0] += flip_parts;
        n[x][y][2] -= flip_parts;
    }
}
}
```

Results

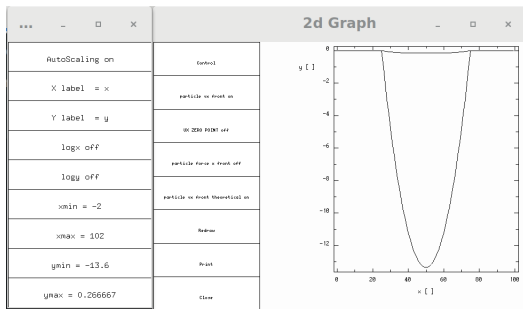


Figure 10:

itext_i

Results

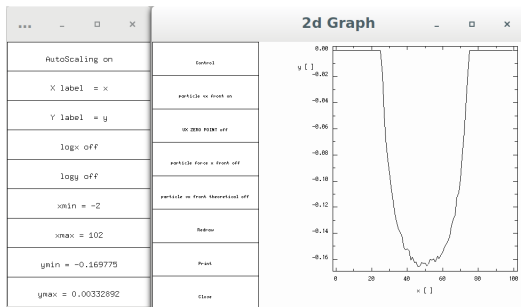


Figure 11:

itext_j

Results

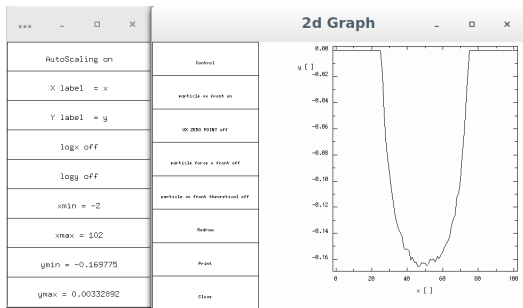


Figure 12:

itext_i

Conclusions and Final thoughts

- ▶ Significant leakage for most walls
- ▶ Partially working
- ▶ Problem depth and complexity
- ▶ Approach 3 Issue might be solvable