

2.1 The different numbers of molecules in per volume (N/V), the same speed and radius

```
from visual import*
from random import random
print("The size of space is a cubic meter")##output introductory words
num=int(input('please input: quantity: '))##input number
r=float(input('radius: '))##input the radius of balls
v=float(input('velocity: '))##input the velocity of balls
wallR=box(pos=(1,0.5,0.5),size=(0.02,1,1),color=color.white)
wallL=box(pos=(0,0.5,0.5),size=(0.02,1,1),color=color.white)
wallA=box(pos=(0.5,0.5,0),size=(1,1,0.02),color=color.white)
wallB=box(pos=(0.5,0.5,1),size=(1,1,0.02),color=color.white,opacity=0.1)
wallU=box(pos=(0.5,1,0.5),size=(1,0.02,1),color=color.white)
wallD=box(pos=(0.5,0,0.5),size=(1,0.02,1),color=color.white) ##define walls
def ball(x,y,z):
    ball=sphere(pos=vector(x,y,z),color=(random(),random(),random()),radius=r)
    ball.velocity=v*vector(random(),random(),random())
    return(ball)
box=[]
for i in range(num):
    bi=ball(random()*0.8,random()*0.8,random()*0.8)
    box.append(bi)##define balls
def pz_wall(ball):
    if ball.pos.x>wallR.pos.x-0.04:
        ball.velocity.x=-ball.velocity.x
    if ball.pos.x<wallL.pos.x+0.04:
        ball.velocity.x=-ball.velocity.x
    if ball.pos.y<wallD.pos.y+0.04:
        ball.velocity.y=-ball.velocity.y
    if ball.pos.y>wallU.pos.y-0.04:
        ball.velocity.y=-ball.velocity.y
    if ball.pos.z<wallA.pos.z+0.04:
        ball.velocity.z=-ball.velocity.z
    if ball.pos.z>wallB.pos.z-0.04:
        ball.velocity.z=-ball.velocity.z##define collision between balls and wall(s)
def pz_ball(ball1,ball2):
    a=abs(ball1.pos.x-ball2.pos.x)
    b=abs(ball1.pos.y-ball2.pos.y)
    c=abs(ball1.pos.z-ball2.pos.z)
    if a*a+b*b+c*c<4*r*r:
        if a<b+c:
            ball1.velocity.x=-ball1.velocity.x
            ball2.velocity.x=-ball2.velocity.x
```

```

    if b<a+c:
        ball1.velocity.y=-ball1.velocity.y
        ball2.velocity.y=-ball2.velocity.y
    if c<a+b:
        ball1.velocity.z=-ball1.velocity.z
        ball2.velocity.z=-ball2.velocity.z##define collision between balls
t=0
dt=0.01
while True:
    rate(100)
    for i in range(num):
        pz_wall(box[i])
        for j in range(num):
            pz_ball(box[i],box[j])
        box[i].pos+=box[i].velocity*dt##continuous operation

```