AST1100 - Oblig 11

Aleksander Hansen

21. november 2011

1 Problem 1

1) We are going to find the pressure in an ideel gas by using the pressure integral,

$$P = \frac{1}{3}\int_0^\infty pvn(p)dp \tag{1}$$

obtained in the lecture notes. An ideel gas follows the Maxwell-Boltzmann distrubution function,

$$n(p)dp = n\left(\frac{1}{2\pi mkT}\right)^{3/2} e^{-p^2/(2mkT)}4\pi p^2 dp \tag{2}$$

The distrubutionfunction $n(p)dp$ in eq. (1) needs to be normalized, so first we need to check that that is the case for eq. (2), if not we need to find the constant that normalized the distribution. The normalizationcondition is,

$$\int_0^\infty n(p)dp = n$$

We have,

$$\int_0^\infty n(p)dp = \int_0^\infty n\left(\frac{1}{2\pi mkT}\right)^{3/2} e^{-p^2/(2mkT)}4\pi p^2 dp$$

$$= 4\pi n\left(\frac{1}{2\pi mkT}\right)^{3/2}\int_0^\infty e^{-p^2/(2mkT)}p^2 dp$$

The solution to the integral has the general form,

$$\int_0^\infty e^{-\lambda x^2}x^k dx = \frac{1}{2}\lambda^{-\frac{k+1}{2}}\Gamma\left(\frac{k+1}{2}\right),\ k > -1,\ \lambda > 0 \tag{3}$$

Therefore we get,

$$\int_0^\infty n(p)dp = 4\pi n\left(\frac{1}{2\pi mkT}\right)^{3/2}\frac{1}{2}\left(\frac{1}{2mkT}\right)^{-3/2}\Gamma\left(\frac{3}{2}\right)$$

And since $\Gamma\left(\frac{3}{2}\right) = \frac{\sqrt{\pi}}{2}$ almost everything cancels and we are left with,

$$\int_0^\infty n(p)dp = n$$

Which means that eq. (2) is normalized.

We can now proceed to calculate the pressure from eq. (1). Substituting $v = p/m$ for the velocity, we get,

$$\begin{aligned} P &= \frac{1}{3}\int_0^\infty \frac{p^2}{m} n \left(\frac{1}{2\pi mkT}\right)^{3/2} e^{-p^2/(2mkT)} 4\pi p^2 dp \\ &= \frac{4\pi}{3m} n \left(\frac{1}{2\pi mkT}\right)^{3/2} \int_0^\infty e^{-p2/(2mkT)} p^4 dp \end{aligned}$$

The solution to this integral has the same form as above, so we get by using eq. (3),

$$P = \frac{4\pi}{3m} n \left(\frac{1}{2\pi mkT}\right)^{3/2} \frac{1}{2} \left(\frac{1}{2mkT}\right)^{-5/2} \Gamma\left(\frac{5}{2}\right)$$

We know that $\Gamma\left(\frac{5}{2}\right) = \Gamma\left(\frac{3}{2}+1\right) = \frac{3}{2}\Gamma\left(\frac{3}{2}\right) = \frac{3\sqrt{\pi}}{4}$ so,

$$\begin{aligned} P &= \frac{4\pi}{3m} n \left(\frac{1}{2\pi mkT}\right)^{3/2} \frac{1}{2} \left(\frac{1}{2mkT}\right)^{-5/2} \frac{3\sqrt{\pi}}{4} \\ &= \frac{1}{2m} n \left(\frac{1}{2mkT}\right)^{3/2} (2mkT)^{5/2} = \frac{1}{2m} n 2mkT \\ P &= nkT \end{aligned}$$

Not too surprisingly we end up with the ideel gas law, as our n is the number density and not the actual number of particles in the gas.

2a) We shall now do a computer simulation testing the expression above, by calculating the force exerted on a wall in a 10cm cubed box. We achieve this by uniformly distributing the particles in the range 0-10cm for each position component, and using a gaussian distribution with a standard deviation of $\sigma = \frac{kT}{m}$ on the velocities. By calculating the positions of the particles $\Delta t = 10^{-9}s$ in the past and in the future, and by determining if they hit the wall or not, we can sum the force of each particle that did hit and then calculate the pressure on that wall. If we use the assumption that the collisions with the wall are elastic, then we can determine if they hit the wall if they are out of bounds after the the time step Δt. Then their velocities in the direction of the wall, lets say the x-direction, are reversed. The force exerted on the wall by each particle is then, $f = \frac{\Delta p}{\Delta t} = \frac{2p_x}{\Delta t}$. Then we can calculate the pressure by $P = \frac{\sum_i f_i}{A}$, where $A = 0.01m^2$ is the area of the wall.

In the text it says to make arrays for each component of the position and velocity, but that don't make any sens when we are only interested in the pressure on one of the walls, and even if we cared about every wall it still don't make sense since the pressure is the same on every wall and the calculation is the same because of the symmetry of the problem. Therefore I have taken the liberty to reduce this to a one-dimensional problem to save cpu-time.

The python code is appended at the end of this document.

2b) The pressures on the wall, calculated for the different temperatures:

```
python oppg2a.py
The pressure on the wall is 7.66874e-10 Pa, at T = 6000 K
The pressure on the wall is 6.45976e-09 Pa, at T = 50000 K
The pressure on the wall is 2.05022e-06 Pa, at T = 15000000 K
The pressure on the wall is 0.000137715 Pa, at T = 1000000000 K
```

2c) The analytical solution we found in 1) is plotted along with the 4 numerical values in a log-log plot shown in Figure 1. The density is just the number of particles divided by the volume. I think the numerical solution looks pretty good, but we could always do better by increasing the number of particles and reducing the time step.

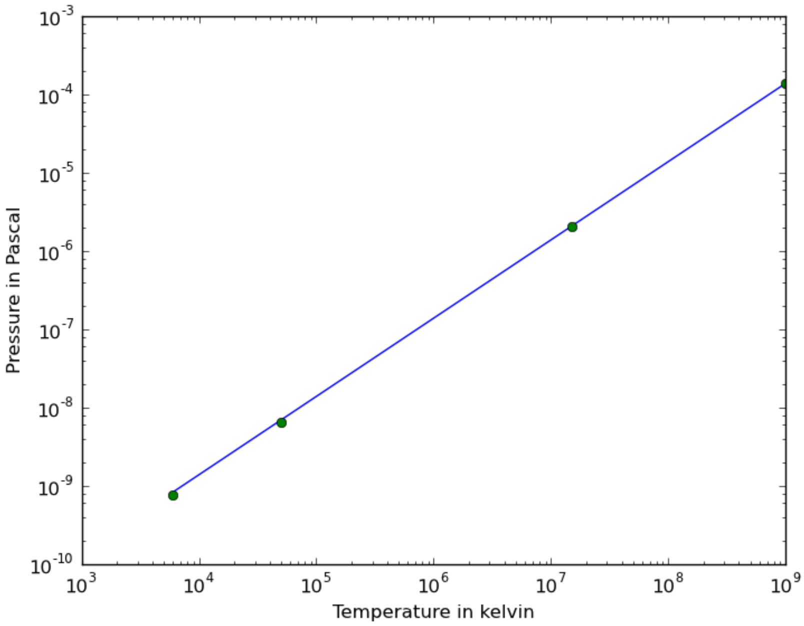
```
from scitools.all import *
import random

def Pressure(T,N,dx,dt):
    k = 1.38e-23  # Boltzmann constant [m^2 kg s^-2 K^-1]
    m = 1.67e-27  # Weight of hydrogen atom [kg]

    sigma = sqrt(k*T/m)

    # Random distribution of the positions and velocities of each particle
    x = zeros(N)
    v = zeros(N)
    for i in range(N):
        x[i] = random.uniform(0,dx)
        v[i] = random.gauss(0,sigma)

    # Calculating the positions of the particles dt in the past and the future
```



Figur 1: Analytical and numerical solution

```
    # and determining if they hit the wall
    F = 0   # Sum of the force
    for i in range(N):
        x_pluss_vdt = x[i] + v[i]*dt
        x_minus_vdt = x[i] - v[i]*dt
        if x_pluss_vdt > 0.1:
            F += 2*m*abs(v[i])/(2*dt)
        if x_minus_vdt > 0.1:
            F += 2*m*abs(v[i])/(2*dt)
    P = F/dx**2   # The pressure on one of the walls
    print 'The pressure on the wall is %g Pa, at T = %d K' % (P, T)
    return P


# Main
N = int(1e7)                # Number of particles
dx = 0.1                    # Size of the cubed box [m]
dt = 1e-9                   # Time step [s]
T = [6e3, 5e4, 15e6, 1e9]   # Temperatures [K]
P = []                      # Pressures corresponding to temperatures in T
```

```
for i in T:
    P.append(Pressure(i,N,dx,dt))

# The analytic expression
k = 1.38e-23  # Boltzmann constant [m^2 kg s^-2 K^-1]
n = N/dx**3   # Density = number of particles/volum
T_analytic = linspace(6e3,1e9,100)
P_analytic = zeros(100)
for i in range(100):
    P_analytic[i] = n*k*T_analytic[i]

# Log-log plot of the analytic solution and the simulated values
loglog(T_analytic,P_analytic)
hold('on')
loglog(T,P,'o')
xlabel('Temperature in kelvin')
ylabel('Pressure in Pascal')
hardcopy('loglog.png')
```