Simulation of Ideal Gas

Pranav Abbaraju

Assumptions Using KMT

- Small particles in constant random linear motion
- No interactive forces
- Average kinetic energy depends on temperature
- Maxwell-Boltzmann distribution for total velocities (but each vector component follows normal curve)

Additional Assumptions

- Ignore particle-particle collisions
- Argon gas

What I Want to Verify

Ideal Gas Law: $PV=nRT$

Compressibility Factor: $Z=PV/nRT$ (equals 1 for ideal)

Boyle's Law: $P_1V_1=P_2V_2$

Gay-Lussac's Law: $P_1/T_1=P_2/T_2$

Simulation of Particles in a Box

```
Sim[nParticles_, xParticles_, m_, length_] := {
  Clear[xPos, trajData];

  trajData = {};
  vTotalArray = {};
  F = 0;
  dpTotal = 0;
  nParticlesUsed = nParticles;
  xParticlesUsed = xParticles[[1 ;; nParticlesUsed]];

  For[i = 1, i ≤ nSteps, i++,
   v = {vTherm, vTherm, vTherm};
   vTotal = Sqrt[v[[1]]^2 + v[[2]]^2 + v[[3]]^2];
   AppendTo[vTotalArray, vTotal];

   For[n = 1, n ≤ nParticlesUsed, n++,
    xPos = xParticlesUsed[[n]][[i]];
    xPos += v dt;

    If[xPos[[1]] < 0, xPos[[1]] = -xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
    If[xPos[[1]] > length, xPos[[1]] = 2 * length - xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
    If[xPos[[2]] < 0, xPos[[2]] = -xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
    If[xPos[[2]] > length, xPos[[2]] = 2 * length - xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
    If[xPos[[3]] < 0, xPos[[3]] = -xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];
    If[xPos[[3]] > length, xPos[[3]] = 2 * length - xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];

    AppendTo[xParticlesUsed[[n]], xPos];
   ];

  ];

  F += dpTotal / time;
 }
```

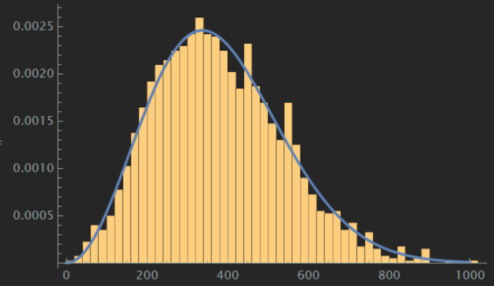
In[3646]:=
```
RT1 = 8.314 * 273; (*units J/mol = (kg m2/s2)/mol*)
M = 39.9 / 1000; (*molar mass: units kg*)
vTherm := Random[NormalDistribution[0, Sqrt[(RT1) / (M)]]] (*units m/s*)
m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)

length = 100;
nParticles = 400;
time = 100; (*units sec*)
dt = .05;
nSteps = Round[time / dt];

xParticles = {};
For[i = 1, i ≤ nParticles, i++,
 AppendTo[xParticles, {{RandomReal[{0, length}], RandomReal[{0, length}], RandomReal[{0, length}]}}];
]

Sim[nParticles, xParticles, m, length];

area = 6 * (length^2);
V1 = length^3;
P1 = F / area;
n = nParticlesUsed / (6.02 * (10^23));
Z = (P1 V1) / (n RT1);

Print["total force: ", F, " N"]
Print["total pressure: ", P1, " Pa"]
Print["compressibility: ", Z]

σ = ((RT1) / M)^0.5;
MaxwellCurve = Plot[PDF[MaxwellDistribution[σ], x], {x, 0, 1000}, PlotRange → {{0, 1000}, {0, .0035}}, PlotStyle → Thick];
vTotalDist = Histogram[vTotalArray, bins = 50, "PDF"];
Show[vTotalDist, MaxwellCurve]
```

total force: 8.98662×10^{-20} N

total pressure: 1.49777×10^{-24} Pa

compressibility: 0.993137

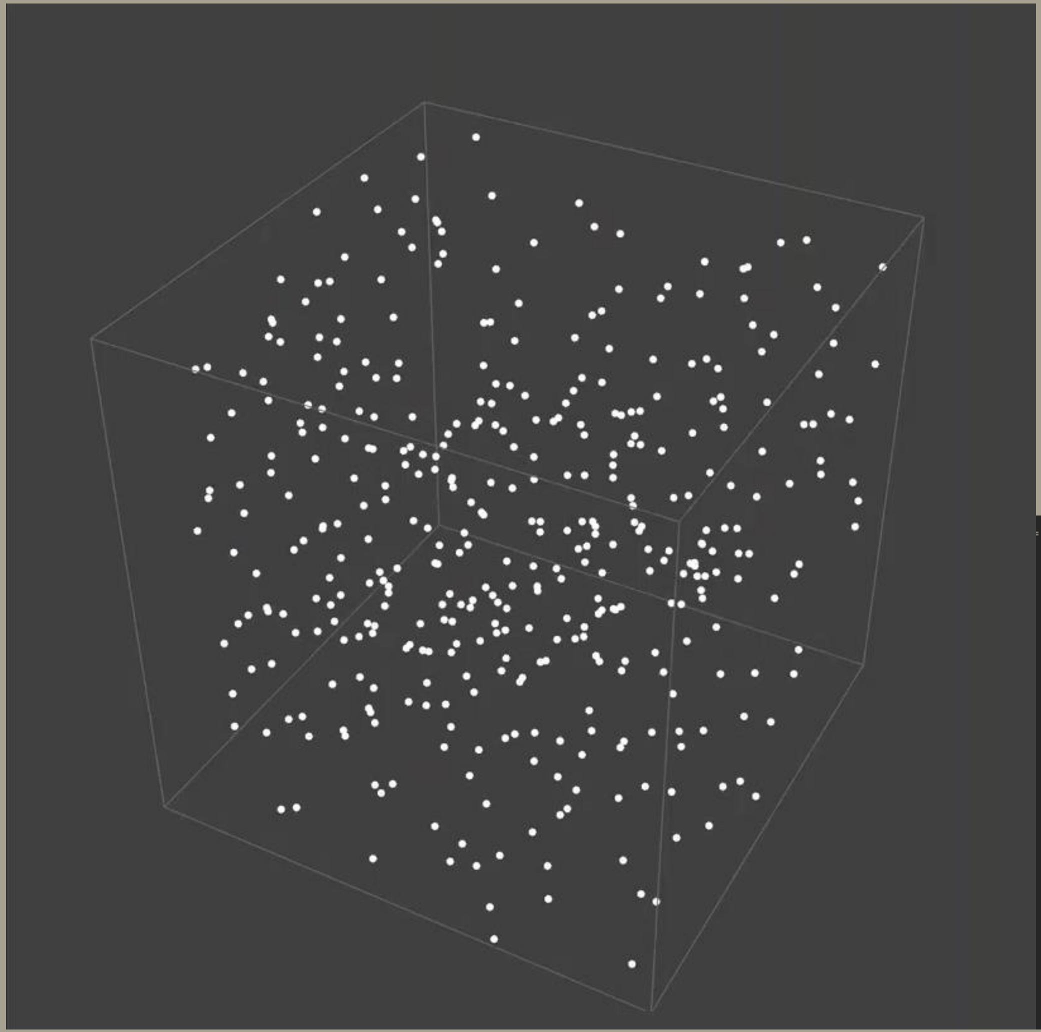
Out[3668]=



Verifying Ideal Gas Law

Conditions:
Argon molar mass, atomic mass
T=273K
Time=100sec
dt=.05sec
Box Length=100m
# of Particles=400

Result:
Z=0.99137
Very close to 1!



Conditions:
T=273K
Box Length=1000m
of Particles=400

```
RT1 = 8.314 * 273; (*units J/mol = (kg m2/s2)/mol*)
M = 39.9 / 1000; (*molar mass: units kg*)
vTherm := Random[NormalDistribution[0, Sqrt[(RT1) / (M)]]] (*units m/s*)
m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)

nParticles = 400;
length = 1000;
time = 100; (*units sec*)
dt = .15;
nSteps = Round[time / dt];

Sim[nParticles, xParticles, m, length];

animationArray = {};
For[i = 1, i ≤ nSteps + 1, i++,
 positionAllParticles = {};
 For[n = 1, n ≤ nParticlesUsed, n++,
  position = xParticlesUsed[[n]][[i]];
  AppendTo[positionAllParticles, position];
 ];
 AppendTo[animationArray, positionAllParticles]
]
v = Map[Point, animationArray];
animation = Animate[Graphics3D[{White, v[[i]]}, PlotRange → {{0, length}, {0, length}, {0, length}}], {i, 1, Length[v], 1}, AnimationRate → 100, RefreshRate → 100,
```

```
In[3678]:= DiffVolume[originalLength_, multiplier_, nRuns_] := {
   pressuresRatioVolume = {};
   lengthMultiplier = multiplier^(1 / 3);
   length = lengthMultiplier * originalLength;

   For[k = 1, k ≤ nRuns, k++,
    RT1 = 8.314 * 273; (*units J/mol = (kg m2/s2)/mol*)
    M = 39.9 / 1000; (*molar mass: units kg*)
    vTherm := Random[NormalDistribution[0, Sqrt[(RT1) / (M)]]]; (*units m/s*)
    m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)

    nParticles = 400;
    time = 100; (*units sec*)
    dt = .05;
    nSteps = Round[time / dt];

    xParticles = {};
    For[i = 1, i ≤ nParticles, i++,
     AppendTo[xParticles, {{RandomReal[{0, length}], RandomReal[{0, length}], RandomReal[{0, length}]}}];
    ];

    Sim[nParticles, xParticles, m, length];

    area = 6 * (length^2);
    V2 = length^3;
    P2 = F / area;
    n = nParticlesUsed / (6.02 * (10^23));
    Z = (P2 V2) / (n RT1);

    pressuresRatioVolume = AppendTo[pressuresRatioVolume, P1 / P2];

    σ = ((RT1) / M)^0.5;
    MaxwellCurve = Plot[PDF[MaxwellDistribution[σ], x], {x, 0, 1000}, PlotRange → {{0, 1000}, {0, .0035}}];
    vTotalDist = Histogram[vTotalArray, bins = 50, "PDF"];
   ];

  }
```

Verifying Boyle's Law: Simulation at Different Volume

[3679]:=
```
DiffVolume[100, 8, 1];

Print["total force: ", F]
Print["total pressure: ", P2]
Print["compressibility: ", Z]
Print["P1/P2 = ", P1 / P2]
Print["V2/V1 = ", V2 / V1]

Show[MaxwellCurve, vTotalDist]
```

total force: 4.34085×10^{-20}

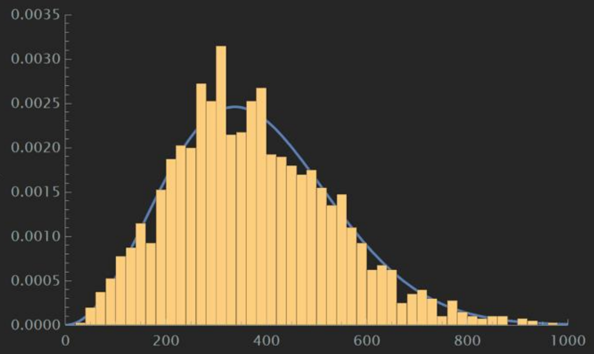
total pressure: 1.80869×10^{-25}

compressibility: 0.959438

P1/P2 = 8.28099

V2/V1 = 8

[3685]=



Condition: V2/V1=8

Result with 1 Run:
P1/P2=8.28099

Result with 10 Runs:
Avg P1/P2=7.93793
Variation from Ideal = 0.77584%

Below is the result if the simulation is run 10 times and the average pressure ratio is taken.

In[3686]:=
```
multiplierVolume = 8;
nRuns = 10;
```

In[3688]:=
```
DiffVolume[100, multiplierVolume, nRuns];
```

In[∘]:=
```
Print[pressuresRatioVolume]
```

{7.89328, 7.26403, 9.67065, 7.92187, 7.69867, 8.26179, 7.91543, 7.13317, 8.76449, 6.85595}

In[∘]:=
```
averagePressureRatioVolume = Mean[pressuresRatioVolume]
Print["variation from ideal: ", Abs[((averagePressureRatioVolume - multiplierVolume) / multiplierVolume ) * 100], "%"]
```

Out[∘]= 7.93793

variation from ideal: 0.77584%

Verifying Gay-Lussac's Law: Simulation at Different Temperature

```
DiffTemp[originalTemp_, multiplier_, nRuns_] := {
  pressuresRatioTemp = {};
  RT2 = multiplier * 8.314 * originalTemp; (*units J/mol = (kg m2/s2)/mol*)

  For[k = 1, k ≤ nRuns, k++,
   M = 39.9 / 1000; (*molar mass: units kg*)
   vTherm := Random[NormalDistribution[0, Sqrt[(RT2) / (M)]]]; (*units m/s*)
   m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)

   length = 1000;
   nParticles = 400;
   time = 100; (*units sec*)
   dt = .05;
   nSteps = Round[time / dt];

   xParticles = {};
   For[i = 1, i ≤ nParticles, i++,
    AppendTo[xParticles, {{RandomReal[{0, length}], RandomReal[{0, length}], RandomReal[{0, length}]}}];
   ];

   Sim[nParticles, xParticles, m, length];

   area = 6 * (length^2);
   V1 = length^3;
   P2 = F / area;
   n = nParticlesUsed / (6.02 * (10^23));
   Z = (P2 V1) / (n RT2);

   pressuresRatioTemp = AppendTo[pressuresRatioTemp, P1 / P2];

   σ = ((RT2) / M)^0.5;
   MaxwellCurve = Plot[PDF[MaxwellDistribution[σ], x], {x, 0, 1800}, PlotRange → {{0, 1800}, {0, .0035}}];
   vTotalDist = Histogram[vTotalArray, bins = 30, "PDF"];
  ];
 }
```

```
In[3721]:= DiffTemp[273, 3, 1];

Print["total force: ", F]
Print["total pressure: ", P2]
Print["compressibility: ", Z]
Print["P1/P2 = ", P1 / P2]
Print["T1/T2 = ", RT1 / RT2]

Show[MaxwellCurve, vTotalDist]
```

total force: 2.69011×10^{-19}

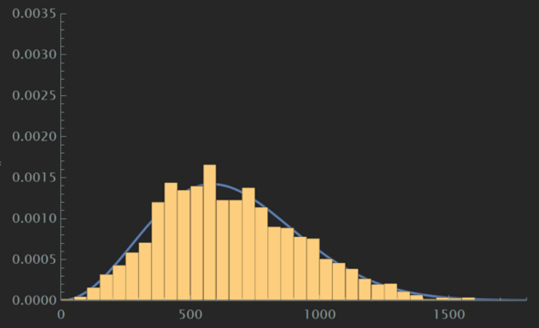
total pressure: 4.48351×10^{-24}

compressibility: 0.990971

P1/P2 = 0.331541

T1/T2 = 0.333333

Out[3727]=




```
In[3728]:= multiplierTemp = 3;
           nRuns = 10;

In[3730]:= DiffTemp[273, multiplierTemp, nRuns];

In[•]:= Print[pressuresRatioTemp]
```

{0.357164, 0.310868, 0.317409, 0.358936, 0.298064, 0.330713, 0.343533, 0.335827, 0.303814, 0.350567}

```
In[•]:= averagePressureRatioTemp = Mean[pressuresRatioTemp]
        Print["variation from ideal: ",
         Abs[((averagePressureRatioTemp - (1 / multiplierTemp)) / (1 / multiplierTemp)) * 100],
         "%"]

Out[•]= 0.33069
```

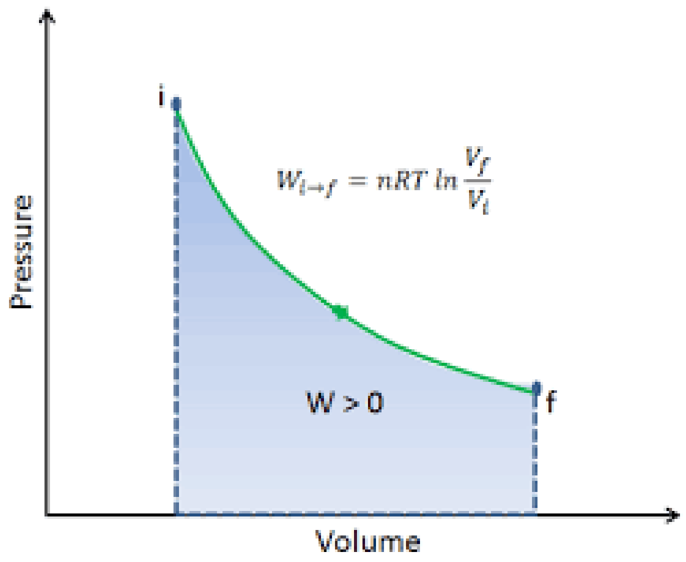
variation from ideal: 0.793144%

Condition: T1/T2=1/3

Result with 1 Run:
P1/P2=0.331541

Result with 10 Runs:
Avg P1/P2=0.33069
Variation from Ideal =
0.7933144%

Isothermal Compression



$$W_{i\to f} = \int_{V_i}^{V_f} p\ dV = \int_{V_i}^{V_f} \frac{nRT}{V}\ dV = nRT \int_{V_i}^{V_f} \frac{1}{V}\ dV = nRT\ ln\frac{V_f}{V_i}$$

Simulation for Isothermal Compression

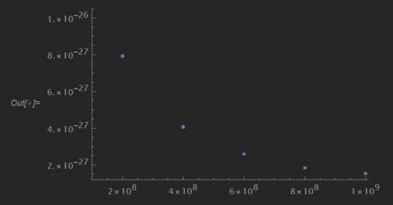
```
In[-]:= IsothermalCE[nParticles_, h1_, h2_, hstep_, m_, Length_] := {
  PVarray = {};
  Zarray = {};

  For[h = h1, h ≥ h2, h += hstep,
   Clear[xPos, trajData];
   xParticles = {};

   trajData = {};
   vTotalArray = {};
   F = 0;
   dpTotal = 0;

   For[o = 1, o ≤ nParticles, o++,
    AppendTo[xParticles, {{RandomReal[{0, Length}], RandomReal[{0, Length}], RandomReal[{0, h}]}}];
   ];

   For[i = 1, i ≤ nSteps, i++,
    v = {vTherm, vTherm, vTherm};
    vTotal = Sqrt[v[[1]]^2 + v[[2]]^2 + v[[3]]^2];
    AppendTo[vTotalArray, vTotal];

    For[n = 1, n ≤ nParticles, n++,
     xPos = xParticles[[n]][[i]];
     xPos += v dt;

     If[xPos[[1]] < 0, xPos[[1]] = -xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
     If[xPos[[1]] > Length, xPos[[1]] = 2 * Length - xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
     If[xPos[[2]] < 0, xPos[[2]] = -xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
     If[xPos[[2]] > Length, xPos[[2]] = 2 * Length - xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
     If[xPos[[3]] < 0, xPos[[3]] = -xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];
     If[xPos[[3]] > h, xPos[[3]] = 2 * h - xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];

     AppendTo[xParticles[[n]], xPos];
    ];

   ];

   F += dpTotal / time;

   area = 4 * Length * h + 2 * Length^2;
   V = h * Length^2;
   P = F / area;
   n = nParticles / (6.02 * (10^23));
   Z = (P V) / (n RT);

   AppendTo[PVarray, {V, P}];
   AppendTo[Zarray, Z];
  ]
 }
```

```
In[•]:= RT = 8.314 * 273; (*units J/mol = (kg m2/s2)/mol*)
M = 39.9 / 1000; (*molar mass: units kg*)
vTherm := Random[NormalDistribution[0, Sqrt[(RT) / (M)]]]; (*units m/s*)
m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)

nParticles = 400;
time = 100; (*units sec*)
dt = .05;
nSteps = Round[time / dt];

length = 1000;
h1 = 1000;
h2 = 100;
hstep = -200;
nRuns = 10;
IsothermalCE[nParticles, h1, h2, hstep, m, length];

In[•]:= Print[PVarray, Zarray]
```

{{1 000 000 000, 1.53888×10^{-27}}, {800 000 000, 1.84273×10^{-27}}, {600 000 000, 2.59661×10^{-27}}, {400 000 000, 4.08045×10^{-27}}, {200 000 000, 7.93431×10^{-27}}}{1.02039, 0.977495, 1.03305, 1.08226, 1.05221}

```
In[•]:= PVplot = ListPlot[PVarray, PlotRange → {{h2 * length^2, 1.01 * h1 * length^2}, {1.2*^-27, 1.05*^-26}}]
```

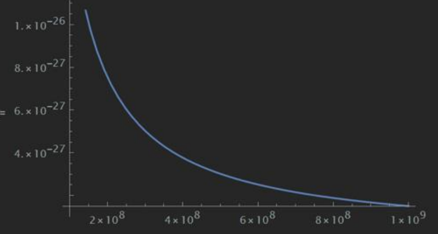
Out[•]=



```
In[•]:= Clear[V, W];
W[V_] = (n RT) / V
w = Integrate[W[V], {V, h2 * length^2, h1 * length^2}];
Plot[W[V], {V, h2 * length^2, h1 * length^2}]
Print["Theoretical work is ", w, " J."]
```

Out[•]= $\frac{1.50812\times10^{-18}}{V}$

Out[•]=

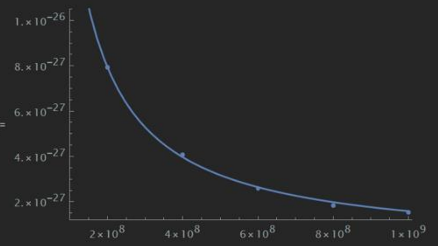


Theoretical work is 3.47258×10^{-18} J.

```
In[•]:= regressionCoefficient = FindFit[PVarray, a / Vfit, a, Vfit];
Wfit[Vfit_] = a / Vfit /. {regressionCoefficient[[1]]}
wEstimate = Integrate[Wfit[x], {x, h2 * length^2, h1 * length^2}];
Show[PVplot, Plot[Wfit[Vfit], {Vfit, h2 * length^2, h1 * length^2}]]
Print["Estimated work calculated using the simulation is ", wEstimate, " J."]
Print["Percent error compared to the theoretical work is ", 100 * Abs[(wEstimate - w) / w], "%"]
```

Out[•]= $\frac{1.58629\times10^{-18}}{Vfit}$

Out[•]=



Estimated work calculated using the simulation is 3.65256×10^{-18} J.

Percent error compared to the theoretical work is 5.18297%

Estimated work using simulation: about 3.653E-18J

Error compared to theoretical integral = about 5.183%

Elastic Collisions of Two Particles

total momentum before = total momentum after

$$m_A v_A + m_B v_B = m_A v'_A + m_B v'_B$$

total kinetic energy before = total kinetic energy after

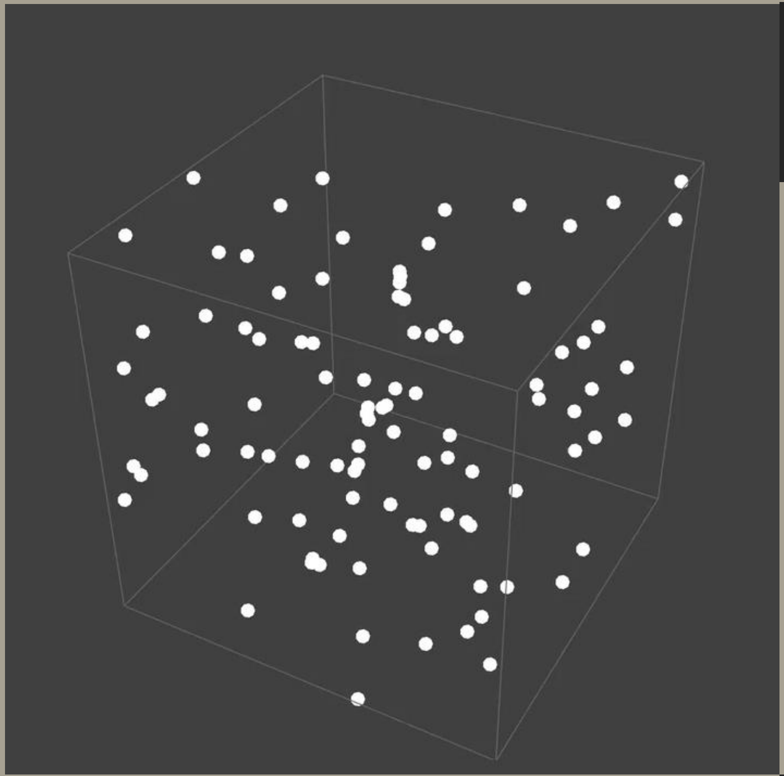
$$\frac{1}{2} m_A v_A^2 + \frac{1}{2} m_B v_B^2 = \frac{1}{2} m_A v_A'^2 + \frac{1}{2} m_B v_B'^2$$

```
In[2714]:= Clear[v1x, v2x, v1y, v2y, v1z, v2z];
  sol = Solve[m v1[[1]] + m v2[[1]] == m v1x + m v2x && 0.5 m v1[[1]]^2 + 0.5 m v2[[1]]^2 == 0.5 m v1x^2 + 0.5 m v2x^2 && m v1[[2]] + m v2[[2]] == m v1y + m v2y && 0.5 m v1[[2]]^2 + 0.5 m v2[[2]]^2 == 0.5 m v1y^2 + 0.5 m v2y^2 &&
     m v1[[3]] + m v2[[3]] == m v1z + m v2z && 0.5 m v1[[3]]^2 + 0.5 m v2[[3]]^2 == 0.5 m v1z^2 + 0.5 m v2z^2 && v1x ≠ v1[[1]], {v1x, v2x, v1y, v2y, v1z, v2z}]

  finalSol = {1, 1, 1, 1, 1, 1};
  For[i = 1, i ≤ Length[sol], i++,
   If[sol[[i]][[1]][[2]] ≠ v1[[1]], finalSol[[1]] = sol[[i]][[1]];];
   If[sol[[i]][[2]][[2]] ≠ v2[[1]], finalSol[[2]] = sol[[i]][[2]];];
   If[sol[[i]][[3]][[2]] ≠ v1[[2]], finalSol[[3]] = sol[[i]][[3]];];
   If[sol[[i]][[4]][[2]] ≠ v2[[2]], finalSol[[4]] = sol[[i]][[4]];];
   If[sol[[i]][[5]][[2]] ≠ v1[[3]], finalSol[[5]] = sol[[i]][[5]];];
   If[sol[[i]][[6]][[2]] ≠ v2[[3]], finalSol[[6]] = sol[[i]][[6]];];
  ]
  finalSolArray = finalSol /. Rule → List
  v1x = finalSolArray[[1]][[2]]
  v2x = finalSolArray[[2]][[2]]
  v1y = finalSolArray[[3]][[2]]
  v2y = finalSolArray[[4]][[2]]
  v1z = finalSolArray[[5]][[2]]
  v2z = finalSolArray[[6]][[2]]
```

```
Out[2719]= -31.2912
Out[2720]= -204.708
Out[2721]= 206.899
Out[2722]= 8.18265
Out[2723]= -235.141
Out[2724]= 148.155

In[2725]:= v1[[1]]
           v2[[1]]
           v1[[2]]
           v2[[2]]
           v1[[3]]
           v2[[3]]
Out[2725]= -204.708
Out[2726]= -31.2912
Out[2727]= 8.18265
Out[2728]= 206.899
Out[2729]= 148.155
Out[2730]= -235.141
```

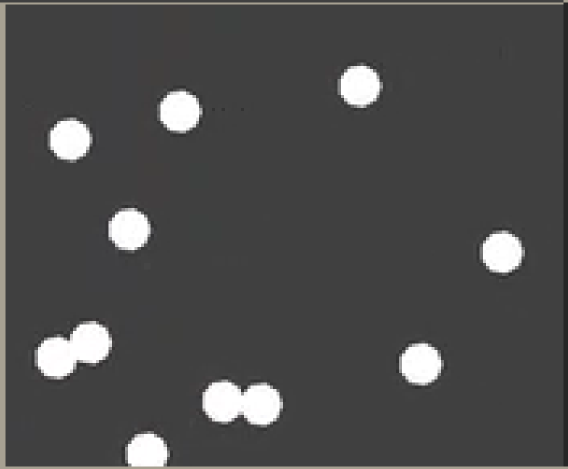
Simulation Including Elastic Collisions of Two Particles at a Time

```
In[2972]:= Sim[nParticles_, m_, Length_] := {
  Clear[xPos, v, xParticles, vParticles];

  F = 0;
  dpTotal = 0;
  collisiondata = {};
  isrunning = {};

  (*------------------------------------------------------------------------------------------*)

  xParticlesLoop = {};
  For[n = 1, n ≤ nParticles, n++,
   AppendTo[xParticlesLoop, {RandomReal[{0, Length}], RandomReal[{0, Length}], RandomReal[{0, Length}]}];
  ];
  xParticles = {xParticlesLoop};

  vTotalArray = {};
  vParticlesLoop = {};
  For[n = 1, n ≤ nParticles, n++,
   vTotal = Sqrt[vAdd[[1]]^2 + vAdd[[2]]^2 + vAdd[[3]]^2];
   AppendTo[vParticlesLoop, {vTherm, vTherm, vTherm}];
   AppendTo[vTotalArray, vTotal]
  ];
  vParticles = {vParticlesLoop};


  (*------------------------------------------------------------------------------------------*)


  For[i = 1, i ≤ nSteps, i++,
   allParticlesAtOneTime = {};
   allVelocitiesAtOneTime = {};


   (*------------------------------------------------------------------------------------------*)


   positionsAtTime = xParticles[[i]];
   distanceArray = {};

   For[k = 1, k ≤ Length[positionsAtTime], k++,
    firstParticlePos = positionsAtTime[[k]];
    distanceWithEachParticle = {};

    For[l = 1, l ≤ Length[positionsAtTime], l++,
     If[l ≠ k,
       secondParticlePos = positionsAtTime[[l]];
       xdist = secondParticlePos[[1]] - firstParticlePos[[1]];
       ydist = secondParticlePos[[2]] - firstParticlePos[[2]];
       zdist = secondParticlePos[[3]] - firstParticlePos[[3]];
       distance = Sqrt[xdist^2 + ydist^2 + zdist^2];
       AppendTo[distanceWithEachParticle, {{k, l}, distance}];
      ];
    ];

    AppendTo[distanceArray, distanceWithEachParticle];
   ];

   collidingParticles = {};
   For[k = 1, k ≤ Length[distanceArray], k++,
    distancesWithParticleCenter = distanceArray[[k]];
    For[l = 1, l ≤ Length[distancesWithParticleCenter], l++,
     If[distancesWithParticleCenter[[l]][[2]] ≤ 2 r,
       AppendTo[collidingParticles, distancesWithParticleCenter[[l]][[1]]];
      ];
    ];
   ];

   collidingParticles1Darray = {};
   For[k = 1, k ≤ Length[collidingParticles], k++,
    For[l = 1, l ≤ Length[collidingParticles[[k]]], l++,
      AppendTo[collidingParticles1Darray, collidingParticles[[k]][[1]]];
      AppendTo[collidingParticles1Darray, collidingParticles[[k]][[1]]];
     ];
   ];
   collidingParticles = DeleteDuplicates[collidingParticles1Darray];

   AppendTo[collisiondata, collidingParticles];

   If[Length[collidingParticles] == 2,
    AppendTo[isrunning, 1];
    particle1 = collidingParticles[[1]];
    particle2 = collidingParticles[[2]];
    v1 = vParticles[[i]][[particle1]];
    v2 = vParticles[[i]][[particle2]];

    vParticles[[i]][[particle1]] = v2;
    vParticles[[i]][[particle2]] = v1;
   ];


   (*------------------------------------------------------------------------------------------*)


   For[n = 1, n ≤ nParticles, n++,
    xPos = xParticles[[i]][[n]];
    v = vParticles[[i]][[n]];
    xPos += v dt;

    If[xPos[[1]] < 0, xPos[[1]] = -xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
    If[xPos[[1]] > Length, xPos[[1]] = 2*Length - xPos[[1]]; dp = Abs[2 m v[[1]]]; dpTotal += dp; v[[1]] = -v[[1]];];
    If[xPos[[2]] < 0, xPos[[2]] = -xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
    If[xPos[[2]] > Length, xPos[[2]] = 2*Length - xPos[[2]]; dp = Abs[2 m v[[2]]]; dpTotal += dp; v[[2]] = -v[[2]];];
    If[xPos[[3]] < 0, xPos[[3]] = -xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];
    If[xPos[[3]] > Length, xPos[[3]] = 2*Length - xPos[[3]]; dp = Abs[2 m v[[3]]]; dpTotal += dp; v[[3]] = -v[[3]];];

    AppendTo[allParticlesAtOneTime, xPos];
    AppendTo[allVelocitiesAtOneTime, v];
    vTotal = Sqrt[v[[1]]^2 + v[[2]]^2 + v[[3]]^2];
    AppendTo[vTotalArray, vTotal];

   ];


   (*------------------------------------------------------------------------------------------*)


   AppendTo[xParticles, allParticlesAtOneTime];
   AppendTo[vParticles, allVelocitiesAtOneTime];

  ];

  F += dpTotal/time;

 }
```



```
{{}, {}, {}, {}, {}, {}, {10, 18}, {}, {}, {}, {}, {}, {}, {}, {22, 31}, {3, 51}, {}, {}, {6, 100}, {}, {}, {},
 {}, {}, {}, {}, {17, 62}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {20, 53}, {}, {}, {}, {}, {}, {}, {}, {},
 {}, {}, {24, 95}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {},
 {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {}, {67, 85}, {}, {5, 84}, {}, {}, {}, {}}
```

Conditions:
T=273K
Box Length=1E-8m
# of Particles=100
time=2E-11 sec, dt=2E-13 sec



```
In[3765]:= RT1 = 8.314 * 273; (*units J/mol = (kg m2/s2)/mol*)
           M = 39.9 / 1000; (*molar mass: units kg*)
           vTherm := Random[NormalDistribution[0, Sqrt[(RT1) / (M)]]] (*units m/s*)
           m = 6.6335209 * (10^(-26)); (*atomic mass of Argon in kg*)
           r = 9.8*^-11; (*atomic radius of Argon in m*)

           length = 1*^-8;
           nParticles = 100;
           time = 2*^-11; (*units sec*)
           dt = 2*^-13;
           nSteps = Round[time / dt];

           Sim[nParticles, m, length];
```