Molecular Dynamics Notes

Enter subtitle here

Enter subsubtitle here

Phil Shea
Enter department here
Enter date here

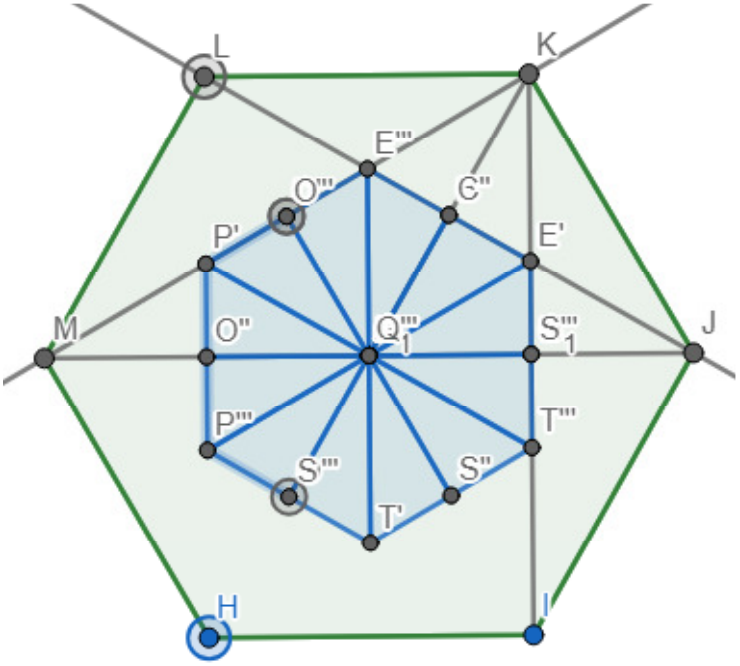
General Parameters

Layout of the Initial Positions

Area of Veroni Cell

The initial position is based on a hexagonal grid oriented to the box sides. Let b_x and b_y be half the sides of the box; the box has corners {{$-b_x$,$-b_y$},{b_x,$-b_y$}, {b_x,b_y},{$-b_x$, b_y}} The box area is then the product of these, and the density is $\rho = N / (4\, b_x b_y)$, where N is the number of particles in the system. Let the hexagonal lattice have particle spacing a (Q_1 - J below). The Veroni cell (the region nearest) around each particle consists of 12 30-60-90 right triangles with sides $a/2$ (Q_1-S_1 in the figure below), $a/(2\sqrt{3})$ (S_1 – E), and hypotenuse $a/\sqrt{3}$ (Q_1 - E; the ‘ marks are from the GeoGebra drawing program indicating how many times the point was used in the drawing).

In[ ]:=



See Fukshansky 2011

The hypotenuse (Q_1 - E) is given by the following:

In[ ]:=
```
Solve[a / 2 == r Cos[30 Degree], r]
```

Out[ ]=
$$\left\{\left\{r \to \frac{a}{\sqrt{3}}\right\}\right\}$$

The short side is therefore

In[ ]:=
```
(a / Sqrt[3]) Sin[30 Degree]
```

Out[ ]=
$$\frac{a}{2\sqrt{3}}$$

The area occupied by one particle is therefore:

In[ ]:=
```
area = 12 (a/2) (a/(2 Sqrt[3])) (1/2)
```

Out[ ]=
$$\frac{\sqrt{3}\ a^2}{2}$$

To achieve a specified density, solve the following:

In[ ]:=
```
Solve[Sqrt[3] a^2/2 == 1/ρ, a]
```

Out[ ]=
$$\left\{\left\{a \to -\frac{\sqrt{2}}{3^{1/4}\sqrt{\rho}}\right\}, \left\{a \to \frac{\sqrt{2}}{3^{1/4}\sqrt{\rho}}\right\}\right\}$$

The paper from 1980 claimed a density of $\rho = 1/\pi$ and an a of about 1.9σ

In[ ]:=
```
a79 = N[Sqrt[2]/(3^(1/4) Sqrt[1/π])]
```

Out[ ]=
```
1.90463
```

The box dimensions can be decided for a desired density. $\rho = N/(4\, b_x\, b_y) = N/(2\sqrt{3}\, b_x^2)$ (where $b_y = \sqrt{3}\, b_x/2$

In[ ]:=
```
Bx[K_,ρ_] = Sqrt[K/(2Sqrt[3] ρ)]
```

Out[ ]=
$$\frac{\sqrt{\frac{K}{\rho}}}{\sqrt{2}\ 3^{1/4}}$$

In[ ]:=
```
Bx[256, 1/π]
```

Out[ ]=
$$\frac{8\sqrt{2\pi}}{3^{1/4}}$$

The positions then will be Nr rows each having Nc particles (this assumes N is composite).

In[ ]:=
```
ClearAll[ placeR];
placeR[n_, a_, x_, y_]:=Table[{x + i a, y},{i,0,n-1}]
```

In[ ]:=
```
placeR[4, .3, .3/4, 0]
```

Out[ ]=
```
{{0.075, 0}, {0.375, 0}, {0.675, 0}, {0.975, 0}}
```

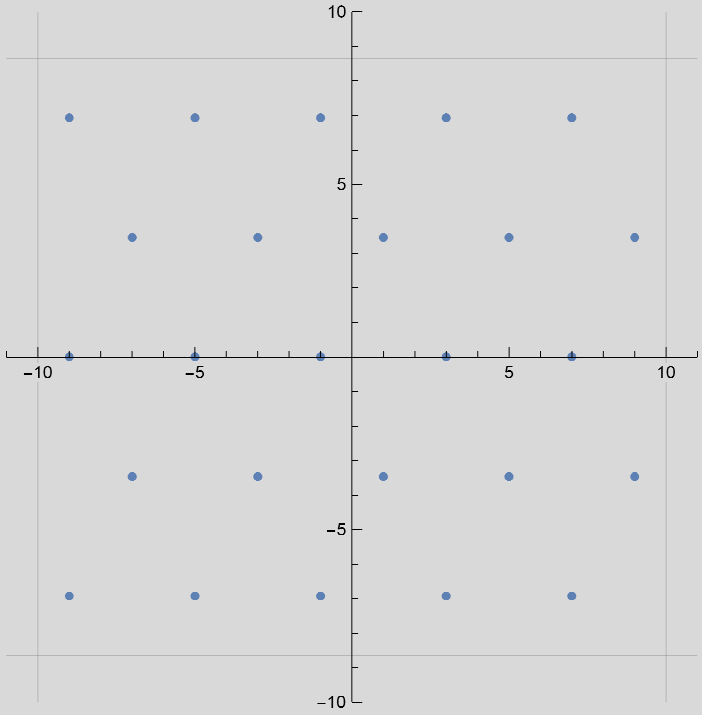
In[ ]:=
```
ClearAll[ fill];
fill::usage="fill[ Bx, By, K] will place K particles in a hexagonal grid in a rectangle of
size 2Bx x 2By.  N should be perfect square such that all the rows have the same number of
particles. Returns a two element list {parms, pnts}; parms is a list {ρ, a, Nc, Nr,Bx,By}
repectivley the density, the particle spacing, the number of rows and columns, and the
input box parmaeters.  pnts is a list of {x, y} coordinates of the K particle positions.";
fill[Bx_, By_, K_]:= Module[{ρ=K/(4 Bx By),a,Nc, Nr,r,c},
a=Sqrt[2/(ρ Sqrt[3])] ;c=Sqrt[3]/2;Nc=Round[2 Bx/a];Nr=Round[K/Nc];
{{ρ, a, Nr, Nc,Bx,By},
Table[placeR[Nc, a, (a/4)(1+2Mod[i,2])-Bx,a c(i+1/2)-By],{i,0,Nr-1}]}];
fill[Bx_, K_]:= fill[Bx, (Sqrt[3]/2)Bx,K]
```

In[ ]:=
```
pnts = fill[10, 10 Sqrt[3] / 2, 25]
```

Out[ ]=
$$\left\{\left\{\frac{1}{8\sqrt{3}}, 4, 5, 5, 10, 5\sqrt{3}\right\},\right.$$
$$\{\{\{-9, -4\sqrt{3}\}, \{-5, -4\sqrt{3}\}, \{-1, -4\sqrt{3}\}, \{3, -4\sqrt{3}\}, \{7, -4\sqrt{3}\}\},$$
$$\{\{-7, -2\sqrt{3}\}, \{-3, -2\sqrt{3}\}, \{1, -2\sqrt{3}\}, \{5, -2\sqrt{3}\}, \{9, -2\sqrt{3}\}\},$$
$$\{\{-9, 0\}, \{-5, 0\}, \{-1, 0\}, \{3, 0\}, \{7, 0\}\},$$
$$\{\{-7, 2\sqrt{3}\}, \{-3, 2\sqrt{3}\}, \{1, 2\sqrt{3}\}, \{5, 2\sqrt{3}\}, \{9, 2\sqrt{3}\}\},$$
$$\left.\{\{-9, 4\sqrt{3}\}, \{-5, 4\sqrt{3}\}, \{-1, 4\sqrt{3}\}, \{3, 4\sqrt{3}\}, \{7, 4\sqrt{3}\}\}\}\right\}$$

In[⁹]:=
```
ListPlot[Flatten[pnts[[2]], 1], AspectRatio → 1,
 PlotRange → {{-11, 11}, {-10, 10}}, GridLines → {{-10, 10}, {-10 √3 / 2, 10 √3 / 2}}]
```

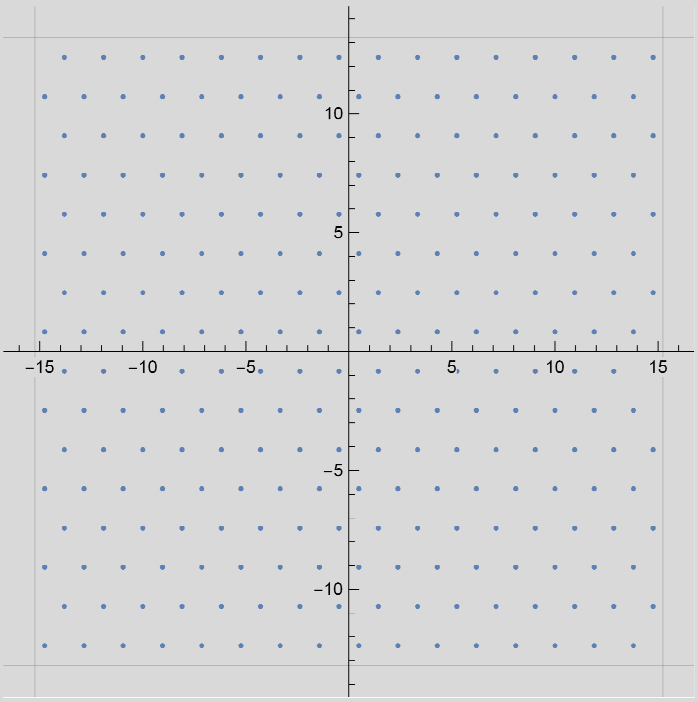
Out[⁹]=



In[⁸]:=
```
ClearAll[plotPnts];
plotPnts[pnts_]:=Module[{Bx=pnts[[1,5]],By=pnts[[1,6]],box},box={{-Bx,Bx},{-By,By}};
ListPlot[Flatten[pnts[[2]],1],AspectRatio→1,PlotRange→1.1box,GridLines→box]]
```

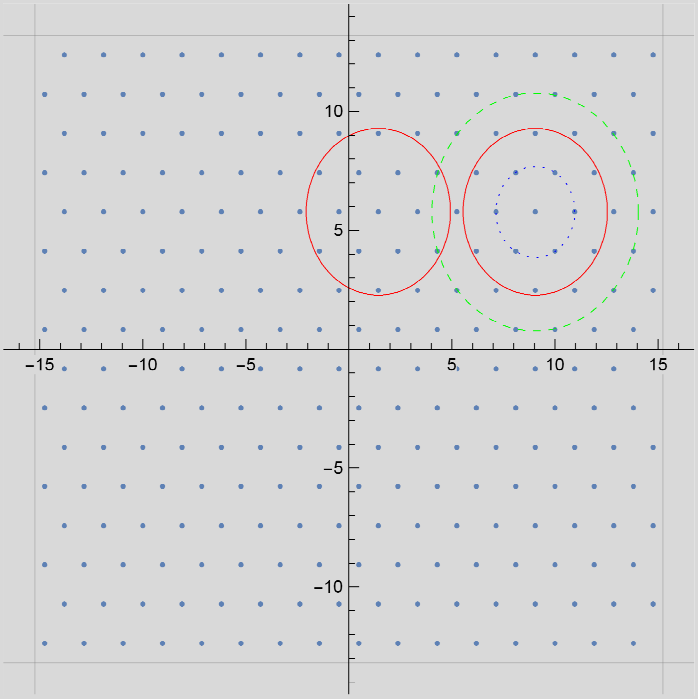
In[ ]:=
```
pnts256 = fill[Bx[256, 1 / π], 256];
p256 = plotPnts[pnts256]
```

Out[ ]=



In[ ]:=
```
p2 = Flatten[pnts256[[2]], 1][[189]];
p3 = Flatten[pnts256[[2]], 1][[189 - 4]];
Show[p256, Graphics[ {Red, Circle[p2, 3.5]}], Graphics[ {Red, Circle[p3, 3.5]}],
 Graphics[{Green, Dashed, Circle[p2, 5]}], Graphics[{Blue, Dotted, Circle[p2, a79]}]]
```

Out[ ]=



The red circles above show that they do not overlap. They could be set to 2*a* so that they would touch. There would be a few more interacting particles then, and at initialization, six particles would be sitting on the skin.

In[ ]:=
```
{3.5^2, (2 a79)^2, 5^2}
```

Out[ ]=
```
{12.25, 14.5104, 25}
```

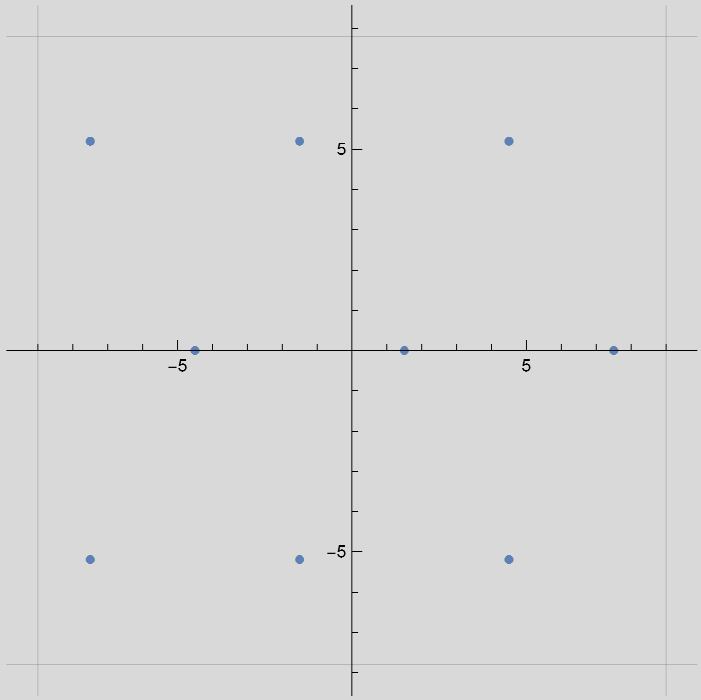
Test Data

In[ ]:=
```
N[fill[9, 9]]
```

Out[ ]=
```
{{0.032075, 6., 3., 3., 9., 7.79423},
 {{{-7.5, -5.19615}, {-1.5, -5.19615}, {4.5, -5.19615}}, {{-4.5, 0.},
   {1.5, 0.}, {7.5, 0.}}, {{-7.5, 5.19615}, {-1.5, 5.19615}, {4.5, 5.19615}}}}
```

In[ ]:=
```
plotPnts[%]
```

Out[ ]=



Potentials

$1/r^n$

The standard repulsive potential is given by:

In[ ]:=
```
ClearAll[rn];
rn[r_, n_]:= 1/r^n
```

Here is the form for a particular n:

In[ ]:=
```
rn[r, 5]
```

Out[ ]=

$$\frac{1}{r^5}$$

And here are some interesting values of that:

In[ ]:=
```
test = {1 / 2, 1, 2, a79, 3, 3.4999, 7 / 2, 4};
Map[ rn[#, 5] &, test]
```

Out[ ]=

$$\left\{32,\ 1,\ \frac{1}{32},\ 0.0398981,\ \frac{1}{243},\ 0.00190424,\ \frac{32}{16\,807},\ \frac{1}{1024}\right\}$$

The force is the derivative with respect to r:

In[ ]:=
```
f[r_] = D[rn[r, 5], r]
```

Out[ ]=
$$-\frac{5}{r^6}$$

Here are the resulting forces at those same interesting values.

In[ ]:=
```
Map[f, test]
```

Out[ ]=
$$\left\{-320, -5, -\frac{5}{64}, -0.10474, -\frac{5}{729}, -0.00272042, -\frac{320}{117\,649}, -\frac{5}{4096}\right\}$$

The skin at 3.5σ yields the following ratios to the mean separation in a hexagonal grid:

In[ ]:=
```
{rn[3.5, 5] / rn[a79, 5], f[3.5] / f[a79]}
```

Out[ ]=
```
{0.0477208, 0.0259687}
```

The number of particles expected in a circular region is $\rho \pi r^2$, but with the density $\rho = 1/\pi$, the expected number of particles is simply r^2

In[ ]:=
```
Block[{ρ = 1 / π}, ρ π {3.5^2, 5^2}]
```

Out[ ]=
```
{12.25, 25}
```

The potential energy (PE) of a particle due to its six nearest neighbors is one half of the PE between each pair (as the PE between particles is shared by the two particles).

In[ ]:=
```
pe6 = 6 rn[a79, 5] / 2
```

Out[ ]=
```
0.119694
```

We can find the nearest points in the following way. Note that when the points are left in symbolic form, the distance returned is the square for some unknown reason. Applying **N[]** to the list and the point yields the desired results.

In[ ]:=
```
Nearest[Flatten[N[pnts256[[2]]], 1] → All, N[p2], {All, 3.5}]
```

Out[ ]=
```
{ <| Element → {9.04697, 5.77309}, Index → 189, Distance → 0. |>,
  <| Element → {8.09466, 7.42254}, Index → 205, Distance → 1.90463 |>,
  <| Element → {9.99928, 7.42254}, Index → 206, Distance → 1.90463 |>,
  <| Element → {7.14235, 5.77309}, Index → 188, Distance → 1.90463 |>,
  <| Element → {8.09466, 4.12364}, Index → 173, Distance → 1.90463 |>,
  <| Element → {10.9516, 5.77309}, Index → 190, Distance → 1.90463 |>,
  <| Element → {9.99928, 4.12364}, Index → 174, Distance → 1.90463 |>,
  <| Element → {6.19003, 7.42254}, Index → 204, Distance → 3.29891 |>,
  <| Element → {9.04697, 9.072}, Index → 221, Distance → 3.29891 |>,
  <| Element → {6.19003, 4.12364}, Index → 172, Distance → 3.29891 |>,
  <| Element → {11.9039, 7.42254}, Index → 207, Distance → 3.29891 |>,
  <| Element → {11.9039, 4.12364}, Index → 175, Distance → 3.29891 |>,
  <| Element → {9.04697, 2.47418}, Index → 157, Distance → 3.29891 |> }
```

Rather than work with a list of associations, we can simply ask for the distances:

In[ ]:=
```
skinlst = Nearest[Flatten[N[pnts256[[2]]], 1] → "Distance", N[p2], {All, 3.5}]
```

Out[ ]=
```
{0., 1.90463, 1.90463, 1.90463, 1.90463, 1.90463,
 1.90463, 3.29891, 3.29891, 3.29891, 3.29891, 3.29891, 3.29891}
```

These can be handed to **rn** to compute the PEs as a list:

In[ ]:=
```
rn[skinlst[[2 ;;]], 5] / 2
```

Out[ ]=
```
{0.019949, 0.019949, 0.019949, 0.019949, 0.019949, 0.019949,
 0.00127973, 0.00127973, 0.00127973, 0.00127973, 0.00127973, 0.00127973}
```

Summing them is easy:

In[ ]:=
```
pe12 = Apply[Plus, %]
```

Out[ ]=
```
0.127373
```

The additional PE of the outer ring of six particles is only a 6% addition.

In[ ]:=
```
dpe = pe12 – pe6
```

Out[ ]=
```
0.00767839
```

In[ ]:=
```
dpe / pe6
```

Out[ ]=
```
0.06415
```

Angular Momentum

When applying random velocities to the particles, the velocities in the x and y directions will each be drawn from a zero-mean Gaussian distribution. The sum of velocities in the x and y direction will therefore also be zero-mean, but will have a substantial variance: for N particles, the variance will be $N\sigma^2$. Therefore, after initialization, we need to measure the momentums in the x and y directions and subtract that average per-particle momentum from each particle.

The same is true for angular momentum. It just won't do for the whole system to be rotating, even a little. It is easy to measure the angular momentum. For a particle with a vector position $\boldsymbol{x}$ and vector velocity $\boldsymbol{v}$, and particle mass m, the angular momentum about the origin is $\boldsymbol{L} = \boldsymbol{x} \times m\,\boldsymbol{v}$. In two dimensions,

In[ ]:=
```
A = {ax, ay, az}; V = {vx, vy, vz}; L = Cross[A, V]
```

Out[ ]=
```
{-az vy + ay vz, az vx - ax vz, -ay vx + ax vy}
```

This is very simple when the z components are zero.

In[ ]:=
```
L /. {az → 0, vz → 0}
```

Out[ ]=
```
{0, 0, -ay vx + ax vy}
```

The total angular momentum is then $L = \sum_{i=1}^{N} x_i \times v_i.$, where we assume $m = 1$. This can also be considered as an average rotation rate ω, and the angular momentum can then also be expressed as $L = \sum_{i=1}^{N} r_i\,\omega = \omega \sum_{i=1}^{N} r_i$, where r_i is the radial component of the particle i in cylindrical coordinates (that is, its distance from the origin). If $S = \sum_{i=1}^{N} r_i$, then the average rotation rate is given by $\omega = L/S$, and therefore the ith particle's share of the rotation is $l_i = \omega\, r_i$.

We can calculate ω, and then say that $l = ax\ vy - ay\ vx$. This is insufficient to calculate a velocity vector, but the velocity vector is orthogonal to the particle vector, so their dot product will be zero.

In[ ]:=
```
Solve[ l == ax vy - ay vx && ax vx + ay vy == 0, {vx, vy}]
```

Out[ ]=
$$\left\{\left\{vx \to -\frac{ay\ l}{ax^2 + ay^2},\ vy \to \frac{ax\ l}{ax^2 + ay^2}\right\}\right\}$$

Numerical Assessments

The value of S is large:

In[ ]:=
```
S = Apply[Plus, Map[Norm, N[Flatten[pnts256[[2]], 1]]]]
```

Out[ ]=
```
2786.81
```

One measurement of L was around 58. ω is then

In[ ]:=
```
58 / S
```

Out[ ]=
```
0.0208123
```

In[ ]:=
```
% / (2 π)
```

Out[ ]=
```
0.00331239
```

In[ ]:=
```
1 / %
```

Out[ ]=
```
301.897
```

So, with that high value of L, it would take a time of almost 302, or 30,000 time steps to make a full rotation. However, only a small rotation would be a problem with the crystal interfaces at the boundaries.

The maximum range is a bit over 19:

In[ ]:=
```
Max[Map[Norm, N[Flatten[pnts256[[2]], 1]]]]
```

Out[ ]=
```
19.2593
```

This means that even the farthest particles will only get less than 1% of the total angular momentum, but this is still more than twice what they would get if the momentum was evenly divided among the 256 particles.

In[ ]:=
```
S / %
```

Out[ ]=
```
144.699
```

Initialization

For the following, see Reif, pg. 249, eq. 7.5.7. The mean kinetic energy of a a particle in any dimension is $\frac{1}{2} k T$, where k is in units of temperature per unit of energy, and T is the temperature in units defined by k. The energy of a particle of mass $m = 1$ is $\frac{1}{2} v^2$ which can be broken into x and y components($\langle \epsilon \rangle = \frac{1}{2} v_X^2 + \frac{1}{2} v_Y^2 = k T$, so that defining $k = 1$, each component has an expected velocity squared of T.

In[12]:=
```
ClearAll[ngt];
ngt[n_, T_] := GammaDistribution[n, 2 T];
{Mean[ngt[n, T]], Variance[ngt[n, T]]}
```

Out[14]=
$$\{2\,n\,T,\ 4\,n\,T^2\}$$

The distribution function with the melting parameters has extreme numbers.

In[6]:=
```
ngt1 = ngt[256, 0.015];
ngtpdf1 = PDF[ngt1, x]
```

Out[7]=
$$\begin{cases} 2.14686 \times 10^{-115}\, e^{-33.3333\,x}\, x^{255} & x > 0 \\ 0 & \text{True} \end{cases}$$

In[9]:=
```
{Mean[ngt1], Variance[ngt1]}
```

Out[9]=
```
{7.68, 0.2304}
```

In[11]:=
```
7.68 / 256 / 2
```

In[43]:=
```
Apply[Plus, Map[#^2 &, RandomVariate[NormalDistribution[0, Sqrt[0.015]], 2 * 10]]]
```

Out[43]=
```
{0.0146742, 0.0000108843, 0.0211145, 0.00155766, 0.0265103, 0.00352063, 0.0000402889,
 0.000943047, 0.00531377, 0.0169177, 0.00349026, 0.000459012, 0.0563798,
 0.0266816, 0.000243944, 0.0634471, 0.00680413, 0.00303738, 0.00221842, 0.0163386}
```

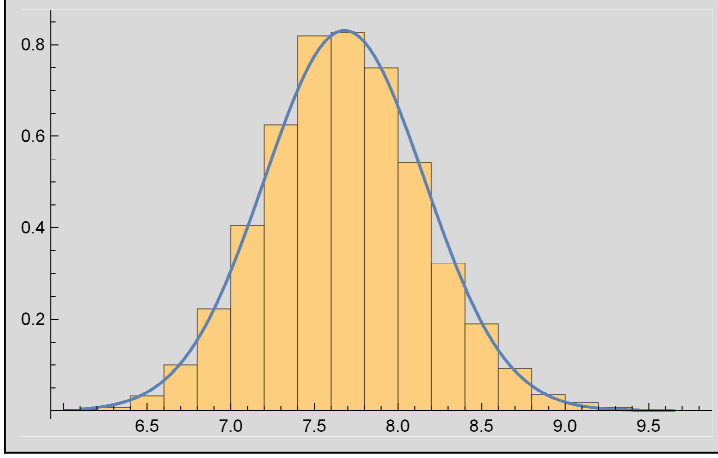
In[44]:=
```
ClearAll[ns];
ns[n_,T_]:=Apply[Plus,Map[#^2&,RandomVariate[NormalDistribution[0,Sqrt[T]],2 n]]]
```

In[62]:=
```
n0 = 256; T0 = 0.015; ns[n0, T0]
```

Out[62]=
```
7.68677
```

In[63]:=
```
data = Table[ns[n0, T0], 10000];
Show[
 Histogram[data, 20, "PDF"],
 Plot[PDF[NormalDistribution[2 n0 T0, 2 √n0 T0], x], {x, Min[data], Max[data]}]]
```

Out[64]=



In[68]:=
```
Through[{Mean, Variance}[data / (2 * 256)]]
```

Out[68]=
$$\{0.0150005,\ 8.62376 \times 10^{-7}\}$$

In[69]:=
```
√%[[2]] / %[[1]]
```

Out[69]=
```
0.0619074
```

So around the melting temperature, there is a 6% standard deviation when generating data.

In[72]:=
```
ClearAll[data]; Share[]
```

Out[72]=
```
7120176
```

Application of Virial Theorem

The Virial THeorem for r^{-n} potentials states that the total kinetic energy is related to the total potential energy as $2\langle K_T\rangle = -n\langle U_T\rangle$, where the angle brackets denote averages. Combine this with the pressure equation $P = \rho\langle K_T\rangle\left(1 + \frac{n\langle U_T\rangle}{2\langle K_T\rangle}\right)$, where ρ is the density, the pressure becomes simply $2\rho\langle K_T\rangle$

Enter subsubsection title here

Enter text here. Enter TraditionalForm input for evaluation in a separate cell below:

In[ ]:=
$$\int x \, dx + \sqrt{z}$$

Out[ ]=
$$\frac{x^2}{2} + \sqrt{z}$$

- Enter bulleted item text here.

  Enter item paragraph text here.

  - Enter subitem text here.

    Enter item paragraph text here.

    - Enter subitem text here.

      Enter item paragraph text here.

Enter text here. Enter formula for display in a separate cell below:

$$\int x \, dx + \sqrt{z}$$

Enter text here. Enter an inline formula like this: 2 + 2.

1. Enter numbered item text here.

Enter item paragraph text here.

1.1. Enter numbered subitem text here.

Enter item paragraph text here.

1.1.1. Enter subitem text here.

Enter item paragraph text here.

Enter text here. Enter formula for numbered display in a separate cell below:

$$\int x \, dx + \sqrt{z} \tag{1}$$

Enter text here. Enter Wolfram Language program code below.

In[ ]:=
```
fun[x_]:=1
```

Enter text here. Enter non-Wolfram Language program code below.

```
DLLEXPORT int fun(WolframLibraryData libData, mreal A1, mreal *Res)
{
 mreal R0_0;
 mreal R0_1;
 R0_0 = A1;
 R0_1 = R0_0 * R0_0;
 *Res = R0_1;
 funStructCompile->WolframLibraryData_cleanUp(libData,  1);
 return 0;
}
```