How to calculate $g(r)$

To calculate $g(r)$ (spherical average) one need to sit on an arbitrary particle i and calculate the number of particles within a radial shell of inner radius r and outer radius $r + \Delta r$. Calling $N_i(r)$ the number of particles found, $g_i(r)$ can be calculated as

$$g_i(r) = \frac{N_i(r)}{N^{ig}(r)} \tag{1}$$

where $N^{ig}(r)$ is the number of pairs that one would have observed if the same system (same number density) would have been an ideal gas, i.e.

$$N^{ig}(r) = \frac{4\pi}{3}\left[(r+\Delta)^3 - r^3)\right]\frac{N-1}{V}$$

(note that we use $(N-1)/V$ to calculate the density, since we know already that one particle is in the origin)

To improve the quality of the results, one can calculate $N_i(r)$ for all particles in the system. In this case one has to put in Eq.1 $N(r) =< N_i(r) >_i$

$$g(r) = \frac{N(r)}{N^{ig}(r)} \tag{2}$$

Let us assume that all particles are within the box. Some codes do not unfold the trajectory and one then need to apply first a fold-back procedure before calculating $g(r)$. For the time being, let's assume all particles are in the box.

So, first of all we need to define a mesh Δr (which must be reasonable to observe the change in $g(r)$). Typically something like $\sigma/20$ is a good choice.

To calculate in an efficient way $N(r)$ one can loop over all pairs of particle $i > j$ and calculate the relative distance, increasing the relative distance counter ($H(r)$ in the following) of that bin.

```
for i=1,N-1
   for j=i+1,N
      calculate rij
       H(rij)=H(rij)+2  (to account for i>j)
    next j
    next i

    H=H/N
    gr=H/N ideal gas
    Save g(r) in a xy file
```

So what is left is only the evaluation of the distance r_{ij}. This has to be done with some care, to account for the periodic boundary conditions. Defining bx the box length and boxh=bx/2 the half-box length, for each direction one has to calculate

```
rijx=x(i)-x(j)
     if (abs(rijx)).gt.boxh) then
          if (rijx>0) rijx=rijx-bx
          else
          rijx=rijx+bx
          end if
      end if

.... same for y and z ....

rij=sqrt(rijx^2+rijy^2+rijz^2)
```

How to calculate $S(q)$

The structure factor $S(\vec{q})$ is defined as

$$S(\vec{q}) = \frac{1}{N} < \rho^*(\vec{q})\rho(\vec{q}) >= \frac{1}{N}|\sum_i e^{i\vec{q}\cdot\vec{r}_i}|^2$$

where

$$\rho(\vec{q}) = \sum_i e^{i\vec{q}\cdot\vec{r}_i}$$

is the Fourier transform of the density $\rho(r) = \sum_i \delta(\vec{r} - \vec{r}_i)$

Let's first calculate $S(\vec{q})$ in a non efficient way, but less prone to mistakes. Later, we will discuss how to make the code more efficient.

Assuming the system is isotropic, we know that we can average together all $\vec{q}$ with the same modulus (which of course means to average together all $\vec{q}$ with modulus between q and $q + \Delta q$).

Due to the periodic boundary conditions, the only accessible $\vec{q}$ are the ones

$$q_x = \frac{2\pi}{b_x} n_x$$

with n_x integer, within $-\infty < n_x < \infty$. In addition, since $\rho(\vec{q})^* = \rho(-\vec{q})$ only half of the $\vec{q}$ space need to be considered.

Here one need to set a value q_{max}, corresponding to a n_{max} beyond which we do not calculate $S(\vec{q})$ anylonger, being "inverse" distances which do not carry any further information (too small). Typically $q_{max} \sim 4\frac{2\pi}{\sigma}$.

```
for nx=-nmax,nmax
for ny=0,nmax
for nz=0,nmax
             calculate q modulus (in mesh units)
             calculate rho(nx,ny,nz) (a complex quantity)
             calculate S(q modulus) (a real quantity)
             S(q modulus)=S(q modulus) + S(nx,ny,nz)
             H(q modulus)=H(q modulus) +1
 next nz
 next ny
 next nz
  S(q modulus)= S(q modulus)/ H(q modulus)
  Save S(q) in a xy file
```

To make the evaluation of $S(\vec{q})$ a little bit faster, one can pre-calculate

$$\rho_x(n_x, i) \equiv e^{i\frac{2\pi}{b_x}n_x x(i)} \quad \rho_y(n_y, i) \equiv e^{i\frac{2\pi}{b_y}n_y x(i)} \rho_z(n_z, i) \equiv e^{i\frac{2\pi}{b_z}n_z x(i)}$$

and then, when requested (remembering that the ρ are complex quantities)

$$\rho(n_x, n_y, n_z) = \sum_i \rho_x(n_x, i)\rho_y(n_y, i)\rho_z(n_z, i)$$

Final consideration

We have not discussed the average over different configurations. This can be done in the same code that calculate $g(r)$ or $S(q)$, adding a loop over the different configurations or it can be done a posteriori by averaging all the produced xy files, one for each configuration. This last strategy allow you to visualize the differences between the different configurations, which sometimes is an useful thing to do.