

Computer experiments

Static equilibrium properties:

Thermodynamic properties (P,T...) (same as MC)

Structure Properties (g(r))

Dynamic equilibrium properties: Diffusion

Radial distribution function $g(r)$:

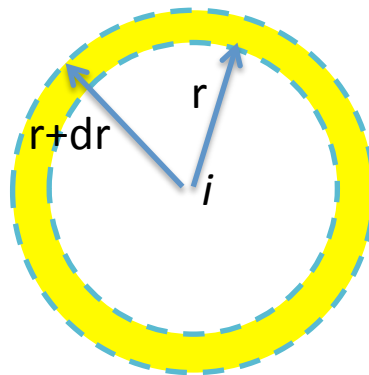
neutron and X-ray scattering experiments yield information about $g(r)$.

$g(r)$ important in theory of liquids: numerical results for $g(r)$ can be compared with theoretical predictions

how to measure $g(r)$ in MD: at any time,

$g(r)$ = ratio between average number density $\rho(r)$ at a distance r from any given atom and the density at a distance r from an atom in an ideal gas at the same overall density.

$g(r) = 1$ in an ideal gas. Any deviation of $g(r)$ from unity reflects correlations between the particles due to the intermolecular interactions.



Radial distribution function $g(r)$:

neutron and X-ray scattering experiments yield information about $g(r)$.

$g(r)$ important in theory of liquids: numerical results for $g(r)$ can be compared with theoretical predictions

how to measure $g(r)$ in MD: at any time,

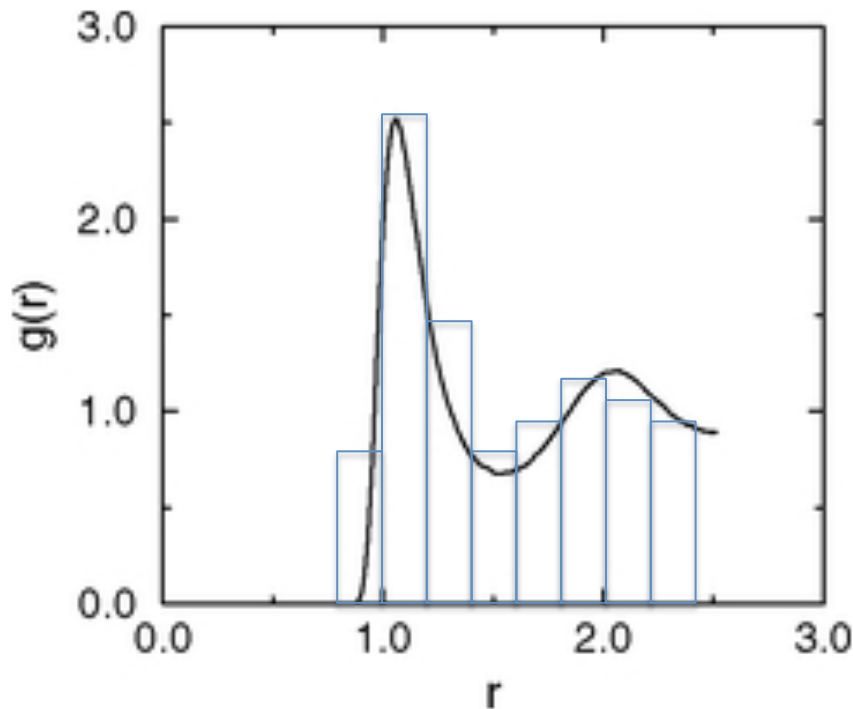
$g(r)$ = ratio between average number density $\rho(r)$ at a distance r from any given atom and the density at a distance r from an atom in an ideal gas at the same overall density.

$g(r) = 1$ in an ideal gas. Any deviation of $g(r)$ from unity reflects correlations between the particles due to the intermolecular interactions.

$$g(r) = \left\langle \frac{\sum_i \# \text{ particles at a distance } (r, r+dr) \text{ from } i}{N_{part}} \times \frac{1}{n_{id} \text{ in } V_r} \right\rangle$$

Calculation of $g(r)$ in MD or MC

The r axis is divided into “ n_{his} ” bins of size $\text{del}g = (\text{box}/2)/n_{\text{his}}$



$g(r) \rightarrow g(k)$ where k is the bin number

We do a histogram:

1- for each pair i,j , we calculate the distance r_{ij}

2- transform r_{ij} to bin#:
 $k = \text{int}(r_{ij}/\text{del}g)$

3- add to $g(k) \rightarrow g(k)+2$

4- Repeat 1-3 for all pairs

For each k , $g(k) = \# \text{ pairs at distance} = k \times \text{del}g$

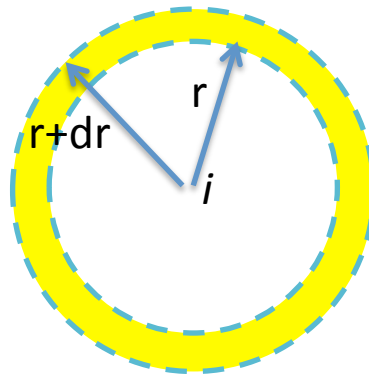
For small values of k , $g(k)=0$ because there are no particles at very short distances ($k \text{ del}g < \sigma$)

Normalization of g:

$$g(r) = \left\langle \frac{\sum_i \# \text{ particles at a distance } (r, r+dr) \text{ from } i}{N_{part}} \times \frac{1}{nid \text{ in } Vr} \right\rangle$$

$$g(r) = \left\langle \frac{g(k)}{N_{part}} \times \frac{1}{nid \text{ in } Vr} \right\rangle$$

$nid \text{ in } Vr = \# \text{ particles for an ideal gas of same density in the yellow volume}$



$$\begin{aligned} Vr &= \frac{4}{3} \pi ((r + dr)^3 - r^3) \\ &= \frac{4}{3} \pi ((k + 1)^3 - k^3) \text{del}g^3 \end{aligned}$$

Subroutine gr (switch)

```
if (switch.eq.0) then
  ngr=0
  delg=box/(2*nhis)
  do i=0,nhis
    g(i)=0
  enddo
else if (switch.eq.1) then
  ngr=ngr+1
  do i=1,npart
    do j=i+1,npart
      xr=x(i)-x(j)
      xr=xr-box*nint(xr/box)
      r=sqrt(xr**2)
      if (r.lt.box/2) then
        ig=int(r/delg)
        g(ig)=g(ig)+2
      endif
    enddo
  enddo
else if (switch.eq.2) then
  do i=1,nhis
    r=delg*(i+0.5)
    vb=((i+1)**3-i**3)*delg**3
    nid=(4/3)*pi*vb*rho
    g(i)=g(i)/(ngr*npart*nid)
  enddo
endif
return
end
```

switch=0 initialization, 1 sample, 2 results

initialization

bin size

nhis is total # bins

sample

loop over all pairs

periodic BC

cut off at Box/2

contribution for particle i and j

determine g(r)

compute distance r

volume between bin l and i+1

ideal gas particles in vb

normalize g(r)

Diffusion

Diffusion is the process whereby an initially nonuniform concentration profile (e.g., an ink drop in water) is smoothed in the absence of flow (no stirring).

Diffusion is caused by the molecular motion of the particles in the fluid.

The macroscopic law that describes diffusion is known as Fick's law,

$$j = -D\nabla c$$

flux $j \sim$ - gradient in the concentration c

D is the diffusion coefficient.

Self-diffusion:

Diffusion of a labeled species among otherwise identical solvent molecules is called self-diffusion

the molecules of the diffusing species are identical to the other molecules but for a label that does not affect the interaction of the labeled molecules with the others.

For instance, this label could be a particular polarization of the nuclear spin of the diffusing species or a modified isotopic composition.

$$\underline{j = -D\nabla c} \quad \underline{\frac{\partial c(r, t)}{\partial t} + \nabla \cdot \mathbf{j}(r, t) = 0.}$$

$$\boxed{\frac{\partial c(r, t)}{\partial t} - D\nabla^2 c(r, t) = 0} \rightarrow \boxed{c(r, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right)}$$

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(r, t) r^2 \quad \frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD$$

$$\begin{aligned} \langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\ &= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\ &= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle. \end{aligned}$$

$$D = \int_0^\infty d\tau \langle v_x(\tau) v_x(0) \rangle$$

$$\mathbf{j} = -D\nabla c \quad \frac{\partial c(\mathbf{r}, t)}{\partial t} + \nabla \cdot \mathbf{j}(\mathbf{r}, t) = 0.$$

$$\frac{\partial c(\mathbf{r}, t)}{\partial t} - D\nabla^2 c(\mathbf{r}, t) = 0 \quad c(\mathbf{r}, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right)$$

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(\mathbf{r}, t) r^2$$

$$\frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD$$

$$\begin{aligned} \langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\ &= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\ &= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle. \end{aligned}$$

$$D = \int_0^\infty d\tau \langle v_x(\tau) v_x(0) \rangle$$

$$j = -D\nabla c \quad \frac{\partial c(r, t)}{\partial t} + \nabla \cdot \mathbf{j}(r, t) = 0.$$

$$\frac{\partial c(r, t)}{\partial t} - D\nabla^2 c(r, t) = 0 \quad c(r, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right)$$

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(r, t) r^2$$

$$\frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD$$

$$\begin{aligned} \langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\ &= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\ &= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle. \end{aligned}$$

$$D = \int_0^\infty d\tau \langle v_x(\tau) v_x(0) \rangle$$

$$j = -D\nabla c \quad \frac{\partial c(r, t)}{\partial t} + \nabla \cdot \mathbf{j}(r, t) = 0.$$

$$\frac{\partial c(r, t)}{\partial t} - D\nabla^2 c(r, t) = 0 \quad c(r, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right)$$

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(r, t) r^2 \quad \frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD$$

$$\begin{aligned} \langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\ &= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\ &= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle. \end{aligned}$$

$$D = \int_0^\infty d\tau \langle v_x(\tau) v_x(0) \rangle$$

$$j = -D\nabla c \quad \frac{\partial c(r, t)}{\partial t} + \nabla \cdot \mathbf{j}(r, t) = 0.$$

$$\frac{\partial c(r, t)}{\partial t} - D\nabla^2 c(r, t) = 0 \quad c(r, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right)$$

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(r, t) r^2 \quad \frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD$$

$$\begin{aligned} \langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\ &= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\ &= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle. \end{aligned}$$

$$D = \int_0^\infty d\tau \langle v_x(\tau) v_x(0) \rangle$$

Diffusion coefficient:

$$D = 1/3 \text{ VACF}$$

Velocity Auto Correlation Function (VACF)

$$VACF = \int_0^{\infty} d\tau \langle v(\tau) \cdot v(0) \rangle = dt \sum_{it=1}^{tsample} \langle v(it - t_0) \cdot v(t_0) \rangle$$

average over particles

One can store the velocities of all the particles on certain times during the MD and do the calculation afterwards.

or we can do the calculation during the MD

Every time we call subroutine SAMPLE

$$VACF = VACF + \sum_{i=1}^{Npart} v_i(it - t_0).v_i(t_0)$$

At the end we normalize:

$$VACF = \frac{VACF}{npart \times ntime}$$



times we called SAMPLE

$$VACF = \sum_{it=1}^{tsample} \langle v(it - t_0) \cdot v(t_0) \rangle \quad \leftarrow VACF(\Delta t, t_0)$$

We also do an average on many values of t_0

$$VACF(\Delta t) = \frac{1}{\#t0s} \sum_i^{\#t0s} VACF(\Delta t, t_{0i})$$

Averages in \rightarrow ensemble (MD)
 \rightarrow t0
 \rightarrow particles.


```

subroutine dif (switch, nsamp)

if (switch.eq.0) then
  ntel=0
  dtime=dt*nsamp
  do i=1,tmax
    ntime(i)=0
    vacf(i)=0
    r2t(i)=0
  enddo
else if (switch.eq.1) then
  ntel=ntel+1
  if (mod(ntel,it0).eq.0) then
    t0 = t0 + 1
    tt0=mod(t0-1,t0max)+1
    time0(tt0)=ntel
    do i=1,npart
      x0(i,tt0)=x(i)
      vx0(i,tt0)=vx(i)
    enddo
  endif
  do t=1,min(t0,t0max)
    delt=ntel-time0(t)+1
    if (delt.lt.tmax) then
      ntime(delt)=ntime(delt)+
+       vx(i)*vx0(i,t)
+       r2t(delt)=r2t(delt)+
+       (x(i)-x0(i,t))**2
    enddo
  endif
enddo
else if (switch.eq.2) then
  do i=1,tmax
    time=dtime*(i+0.5)
    vacf(i)=vacf(i)
+     / (npart*ntime(i))
    r2t(i)=r2t(i)
+     / (npart*ntime(i))
  enddo
endif
return
end

```

```

subroutine dif (switch , nsamp)
if (switch.eq.0) then
  ntel=0
  dtime=dt*nsamp
  do i=1,tmax
    ntime(i)=0
    vacf(i)=0
    r2t(i)=0
  enddo

```

initialize:

ntel: counter

dtime: time between 2
samples

$i = \Delta t$

ntime: # times vacf is
calculated for each i (Δt)

$tmax$: # values of Δt

update velocity autocorr.

update mean-squared displ.

determine results

time
volume velocity autocorr.

mean-squared displacement

```

subroutine dif(switch,nsamp)

if (switch.eq.0) then
  ntel=0
  dtime=dt*nsamp
  do i=1,tmax
    ntime(i)=0
    vacf(i)=0
    r2t(i)=0
  enddo
else if (switch.eq.1) then
  ntel=ntel+1
  if (mod(ntel,it0).eq.0) then
    t0 = t0 + 1
    tt0=mod(t0-1,t0max)+1
    time0(tt0)=ntel
    do i=,npart
      x0(i,tt0)=x(i)
      vx0(i,tt0)=vx(i)
    enddo
  endif
  do t=1,min(t0,t0max)
    delt=ntel-time0(t)+1
    if (delt.lt.tmax) then
      ntime(delt)=ntime(delt)+1
      do i=1,npart
        vacf(delt)=vacf(delt)+
+         vx(i)*vx0(i,t)
+         r2t(delt)=r2t(delt)+
+         (x(i)-x0(i,t))**2
      enddo
    endif
  enddo
else if (switch.eq.2) then
  do i=1,tmax
    time=dtime*(i+0.5)
    vacf(i)=vacf(i)
+     /(npart*ntime(i))
    r2t(i)=r2t(i)
+     /(npart*ntime(i))
  enddo
endif
return
end

```

diffusion; switch = 0 init.
 = 1 sample, and = 2 results
 Initialization
 time counter
 time between two samples
 tmax total number of time step
 number of samples for time i

sample

decide to take a new $t = 0$
 update number of $t = 0$
 see note 1
 store the time of $t = 0$

loop on t_0 s (index by t)

delt: Δt

```

do t=1,min(t0,t0max)
  delt=ntel-time0(t)+1
  if (delt.lt.tmax) then
    ntime(delt)=ntime(delt)+1
    do i=1,npart
      vacf(delt)=vacf(delt)+vx(i) vx0(i,t)
      r2t(delt)=r2t(delt)+(x(i)-x0(i,t))**2
    enddo
  endif
enddo

```

loop on particles

```

subroutine dif(switch,nsamp)
  if (switch.eq.0) then
    ntel=0
    dtime=dt*nsamp
    do i=1,tmax
      ntime(i)=0
      vacf(i)=0
      r2t(i)=0
    enddo
  else if (switch.eq.1) then
    ntel=ntel+1
    if (mod(ntel,it0).eq.0) then
      t0 = t0 + 1
      tt0=mod(t0-1,t0max)+1
      time0(tt0)=ntel
      do i=1,npart
        x0(i,tt0)=x(i)
        vx0(i,tt0)=vx(i)
      enddo
    endif
    do t=1,min(t0,t0max)
      delt=ntel-time0(t)+1
      if (delt.lt.tmax) then
        ntime(delt)=ntime(delt)+1
        do i=1,npart
          vacf(delt)=vacf(delt)+
+          vx(i)*vx0(i,t)
+          r2t(delt)=r2t(delt)+
+          (x(i)-x0(i,t))**2
        enddo
      endif
    enddo
  else if (switch.eq.2) then
    do i=1,tmax
      time=dtime*(i+0.5)
      vacf(i)=vacf(i)
+      / (npart*ntime(i))
      r2t(i)=r2t(i)
+      / (npart*ntime(i))
    enddo
  endif
  return
end

```

diffusion; switch = 0 init.
 = 1 sample, and = 2 results
 Initialization
 time counter
 time between two samples
 tmax total number of time step
 number of samples for time i

sample

decide to take a new $t = 0$
 update number of $t = 0$
 see note 1
 store the time of $t = 0$

store position for given $t = 0$
 store velocity for given $t = 0$

update vacf and r2 for $t = 0$
 actual

average on particles:

do i=1,npart

$\text{vacf(delt)} = \text{vacf(delt)} + \text{vx(i)} \cdot \text{vx0(i,t)}$

$\text{r2t(delt)} = \text{r2t(delt)} + (\text{x(i)} - \text{x0(i,t)})^2$

enddo

deter

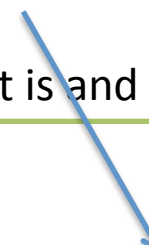
time
 volume velocity autocorr.

mean-squared displacement

$\text{delt} = \Delta t$

t is t0 (discrete)

t is and integer index



```
subroutine dif(switch,nsamp)
```

```
if (switch.eq.0) then
```

```
ntel=0
```

```
dttime=dt*nsamp
```

```
do i=1,tmax
```

```
ntime(i)=0
```

```
vacf(i)=0
```

```
r2t(i)=0
```

```
enddo
```

```
else if (switch.eq.1) then
```

```
ntel=ntel+1
```

```
if (mod(ntel,it0).eq.0) then
```

```
t0 = t0 + 1
```

```
tt0=mod(t0-1,t0max)+1
```

```
time0(tt0)=ntel
```

```
do i=,npart
```

```
x0(i,tt0)=x(i)
```

```
vx0(i,tt0)=vx(i)
```

```
enddo
```

```
endif
```

```
do t=1,min(t0,t0max)
```

```
delt=ntel-time0(t)+1
```

```
if (delt.lt.tmax) then
```

```
ntime(delt)=ntime(delt)+1
```

```
do i=1,npart
```

```
vacf(delt)=vacf(delt)+
```

```
+ vx(i)*vx0(i,t)
```

```
r2t(delt)=r2t(delt)+
```

```
+ (x(i)-x0(i,t))**2
```

```
enddo
```

```
endif
```

```
enddo
```

```
else if (switch.eq.2) then
```

```
do i=1,tmax
```

```
time=dttime*(i+0.5)
```

```
vacf(i)=vacf(i)
```

```
+ / (npart*ntime(i))
```

```
r2t(i)=r2t(i)
```

```
+ / (npart*ntime(i))
```

```
enddo
```

```
endif
```

```
return
```

```
end
```

diffusion; switch = 0 init.

= 1 sample, and = 2 results

Initialization

time counter

time between two samples

tmax total number of time step

number of samples for time i

```
ntel=ntel+1
```

```
if(mod(ntel,it0).eq.0) then
```

```
t0=t0+1
```

```
tt0=mod(t0-1,t0max)+1
```

```
time0(tt0)=ntel
```

```
do i=1,npart
```

```
x0(i,tt0)=x(i)
```

```
vx0(i,tt0)=vx(i)
```

```
enddo
```

```
endif
```

determine results

time

volume velocity autocorr.

mean-squared displacement

new time0

t0: counter

every it0 steps, one time0 is chosen: time0(tt0)

the values of velocities and positions of all particles are stored to be used in the calculation of vacr for many t0s.

tt0: integer, label the new time0

```

subroutine dif(switch,nsamp)
  if (switch.eq.0) then
    ntel=0
    dtime=dt*nsamp
    do i=1,tmax
      ntime(i)=0
      vacf(i)=0
      r2t(i)=0
    enddo
  else if (switch.eq.1) then
    ntel=ntel+1
    if (mod(ntel,it0).eq.0) then
      t0 = t0 + 1
      tt0=mod(t0-1,t0max)+1
      time0(tt0)=ntel
      do i=,npart
        x0(i,tt0)=x(i)
        vx0(i,tt0)=vx(i)
      enddo
    endif
    do t=1,min(t0,t0max)
      delt=ntel-time0(t)+1
      if (delt.lt.tmax) then
        ntime(delt)=ntime(delt)+1
        do i=1,npart
          vacf(delt)=vacf(delt)+
+           vx(i)*vx0(i,t)
          r2t(delt)=r2t(delt)+
+           (x(i)-x0(i,t))**2
        enddo
      endif
    enddo
  else if (switch.eq.2) then
    do i=1,tmax
      time=dtime*(i+0.5)
      vacf(i)=vacf(i)
+       / (npart*ntime(i))
      r2t(i)=r2t(i)
+       / (npart*ntime(i))
    enddo
  endif
  return
end

```

diffusion; switch = 0 init.
 = 1 sample, and = 2 results
 Initialization
 time counter
 time between two samples
 tmax total number of time step
 number of samples for time i

sample

decide to take a new $t = 0$
 update number of $t = 0$
 see note 1
 store the time of $t = 0$

store position for given $t = 0$
 store velocity for given $t = 0$

update vacf and r2, for $t = 0$
 actual time minus $t = 0$

update velocity autocorr.

update mean-squared displ.

normalize

i: label for Δt

$\text{ntime}(i) = \#t_0\text{s for a given } \Delta t$

```

do i=1,tmax
  time=dtime*(i+0.5)
  vacf(i)=vacf(i)/(npart*ntime(i))
  r2t(i)=r2t(i)/(npart*ntime(i))
enddo

```

