Molecular Dynamics

Basics (4.1, 4.2, 4.3)

Liouville formulation (4.3.3)

Multiple timesteps (15.3)

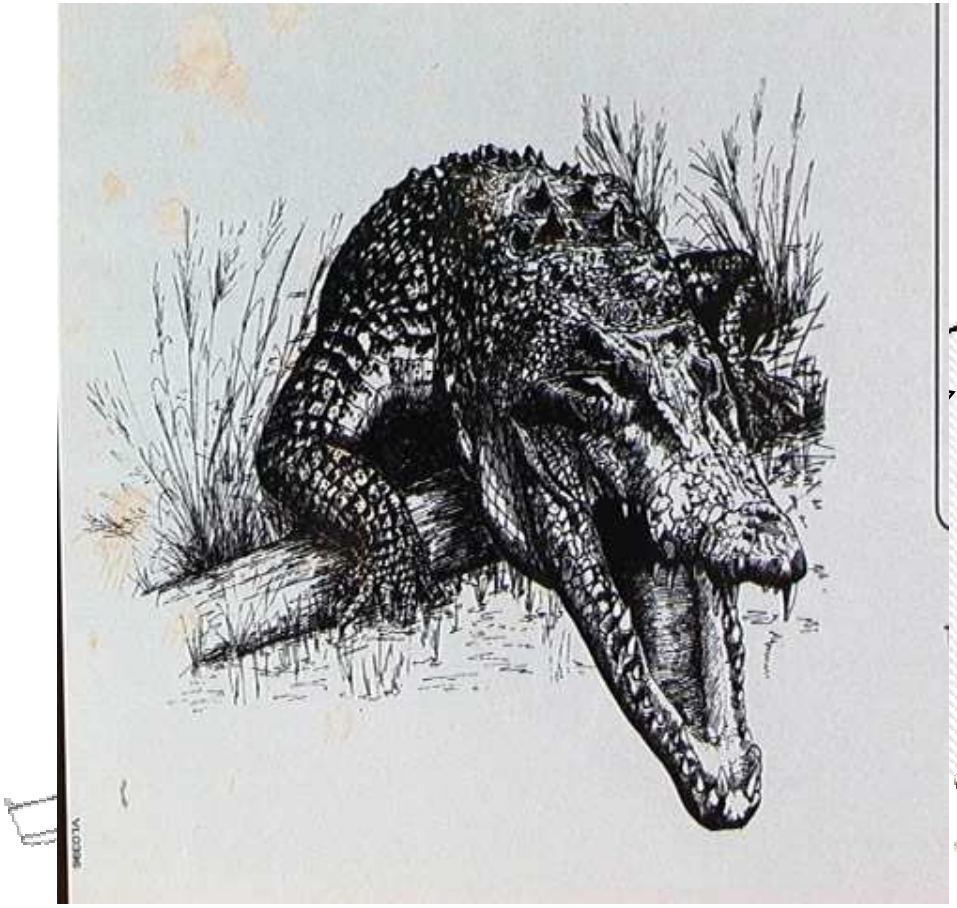
Molecular Dynamics

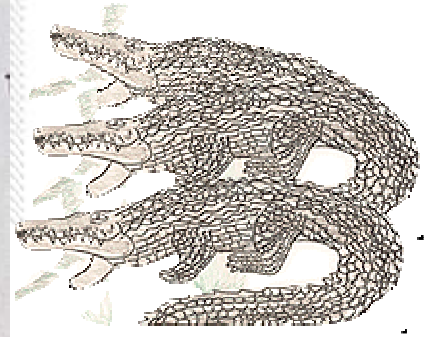
- Theory:

$$\mathbf{F} = m\frac{d^2\mathbf{r}}{dt^2}$$

- Compute the forces on the particles
- Solve the equations of motion
- Sample after some time steps

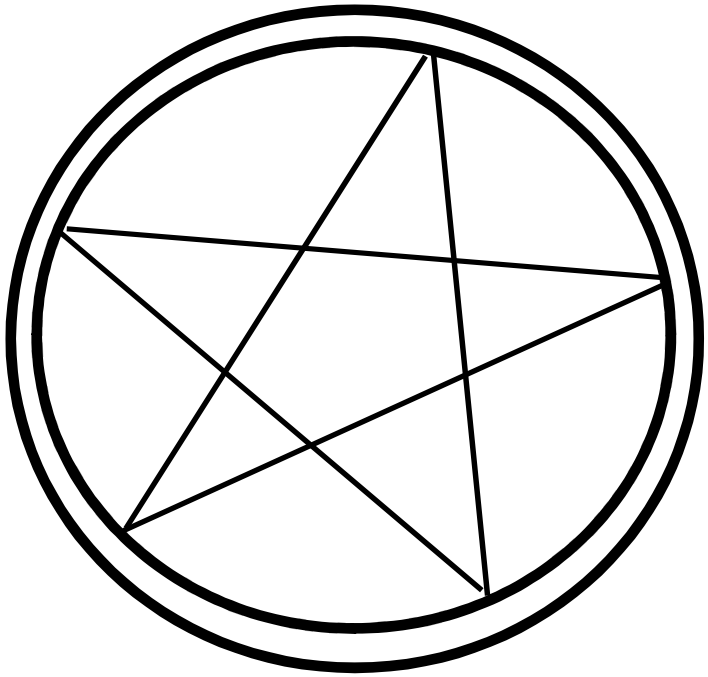
MD Sampling

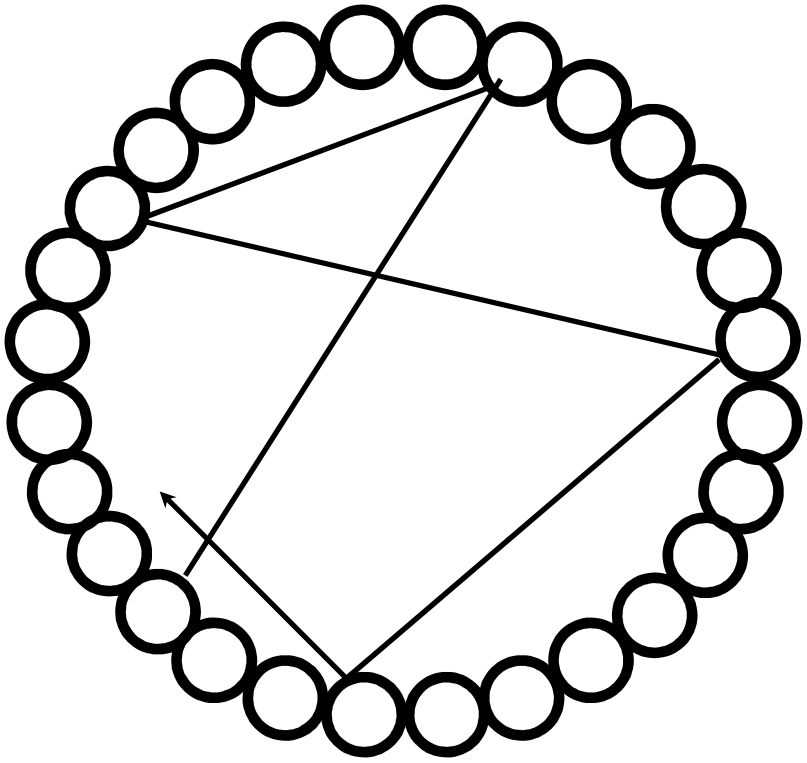




MD Sampling

- Ergodicity



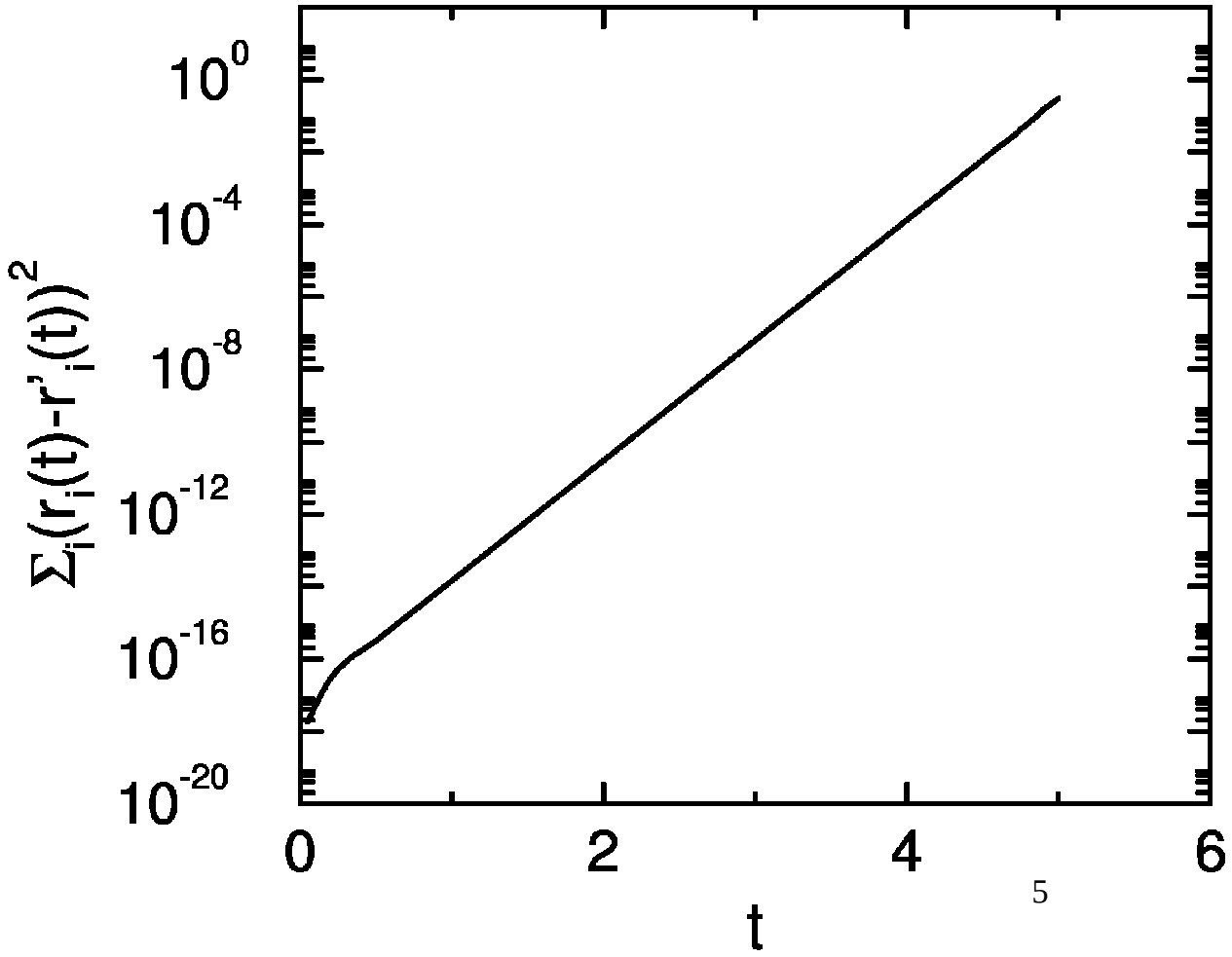


Lyaponov instability

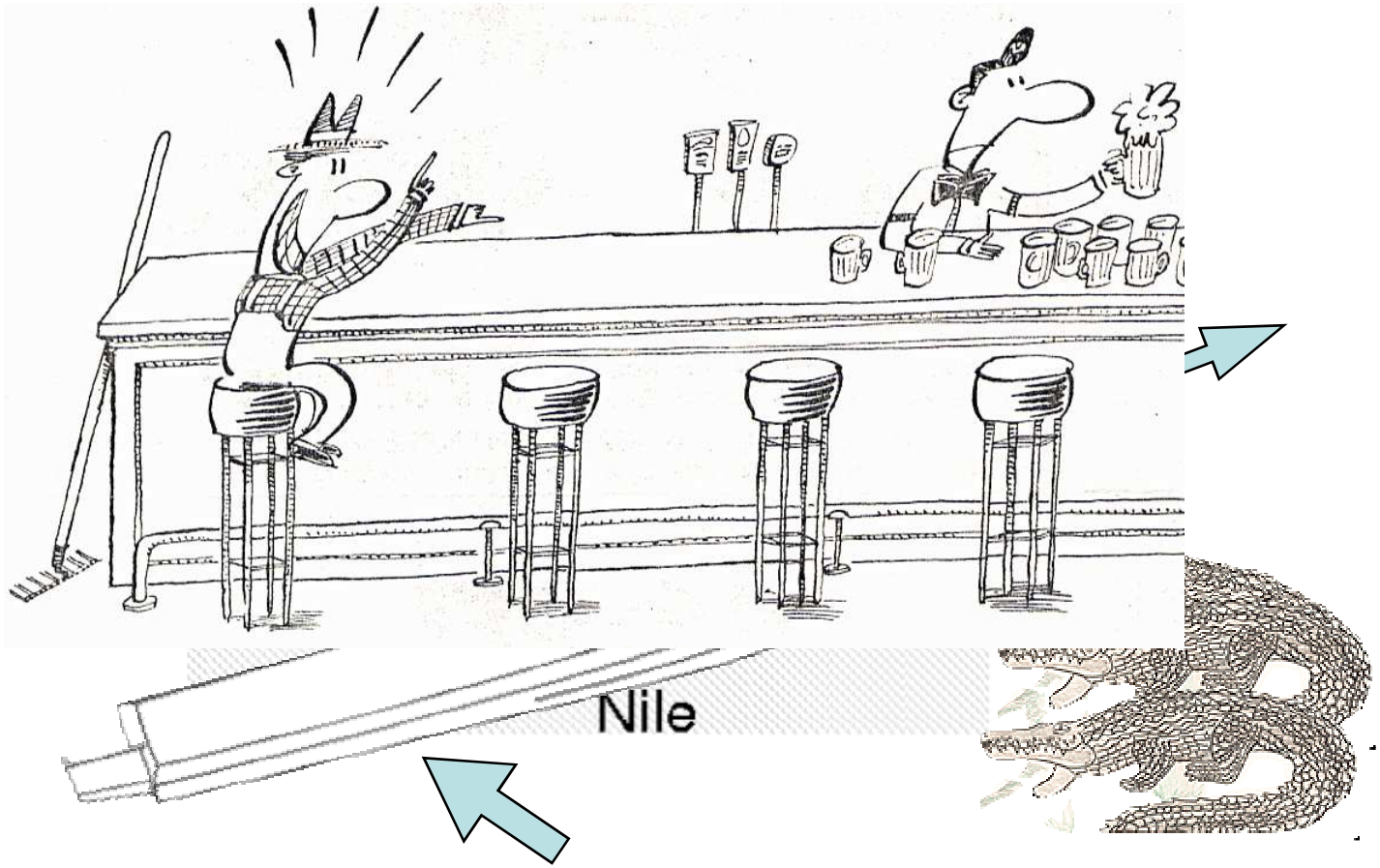
$$\left(\mathbf{r}^N(0), \mathbf{p}^N(0)\right)$$

$$\left(\mathbf{r}^N(0), \mathbf{p}_1(0), \cdots, \mathbf{p}_j(0)+\varepsilon, \mathbf{p}_i(0)-\varepsilon \cdots, \mathbf{p}_N(0)\right)$$

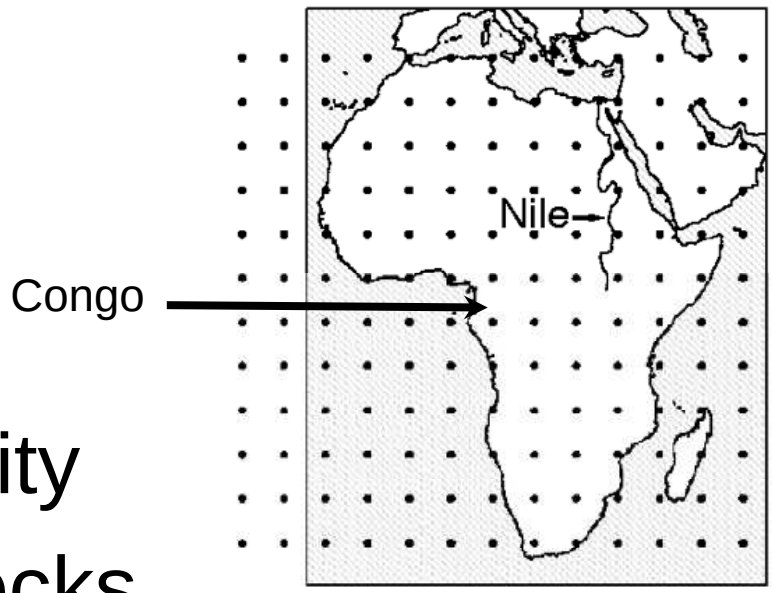
$$\varepsilon = 10^{-10}$$

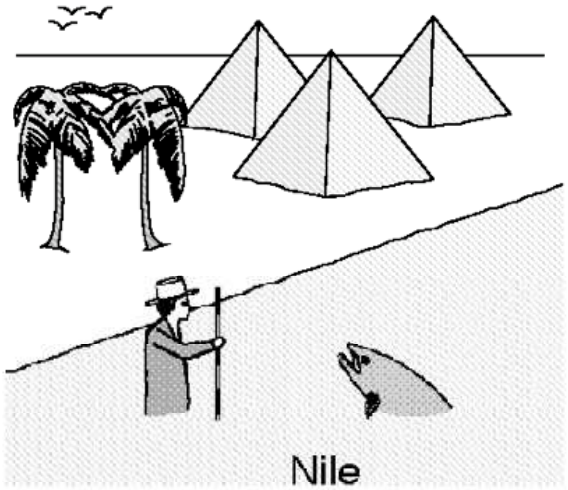


MD Sampling



MD Sampling





- Ergodicity
- bottlenecks
- time scale
- representability

short MD movie

- Ergodicity
- bottlenecks
- time scale
- representability

Algorithm 3 (A Simple Molecular Dynamics Program)

```
program md                          simple MD program

call init                           initialization
t=0
do while (t.lt.tmax)                MD loop
   call force(f,en)                 determine the forces
   call integrate(f,en)             integrate equations of motion
   t=t+delt
   call sample                      sample averages
enddo
stop
end
```

Comment to this algorithm:

1. *Subroutines* `init`, `force`, `integrate`, *and* `sample` *will be described in Algorithms 4, 5, and 6, respectively. Subroutine* `sample` *is used to calculate averages like pressure or temperature.*

Algorithm 4 (Initialization of a Molecular Dynamics Program)

```
subroutine init                initialization of MD program
sumv=0
sumv2=0
do i=1,npart
  x(i)=lattice_pos(i)          place the particles on a lattice
  v(i)=(ranf()-0.5)            give random velocities
  sumv=sumv+v(i)               velocity center of mass
  sumv2=sumv2+v(i)**2          kinetic energy
enddo
sumv=sumv/npart                velocity center of mass
sumv2=sumv2/npart              mean-squared velocity
fs=sqrt(3*temp/sumv2)          scale factor of the velocities
do i=1,npart                   set desired kinetic energy and set
  v(i)=(v(i)-sumv)*fs          velocity center of mass to zero
  xm(i)=x(i)-v(i)*dt           position previous time step
enddo
return
end
```

Algorithm 5 (Calculation of the Forces)

```
subroutine force(f,en)                    determine the force
en=0                                      and energy
do i=1,npart
   f(i)=0                                 set forces to zero
enddo
do i=1,npart-1
  do j=i+1,npart                          loop over all pairs
     xr=x(i)-x(j)
     xr=xr-box*nint(xr/box)               periodic boundary conditions
     r2=xr**2
     if (r2.lt.rc2) then                  test cutoff
       r2i=1/r2
       r6i=r2i**3
       ff=48*r2i*r6i*(r6i-0.5)            Lennard-Jones potential
       f(i)=f(i)+ff*xr                    update force
       f(j)=f(j)-ff*xr
       en=en+4*r6i*(r6i-1)-ecut           update energy
     endif
  enddo
enddo
return
end
```

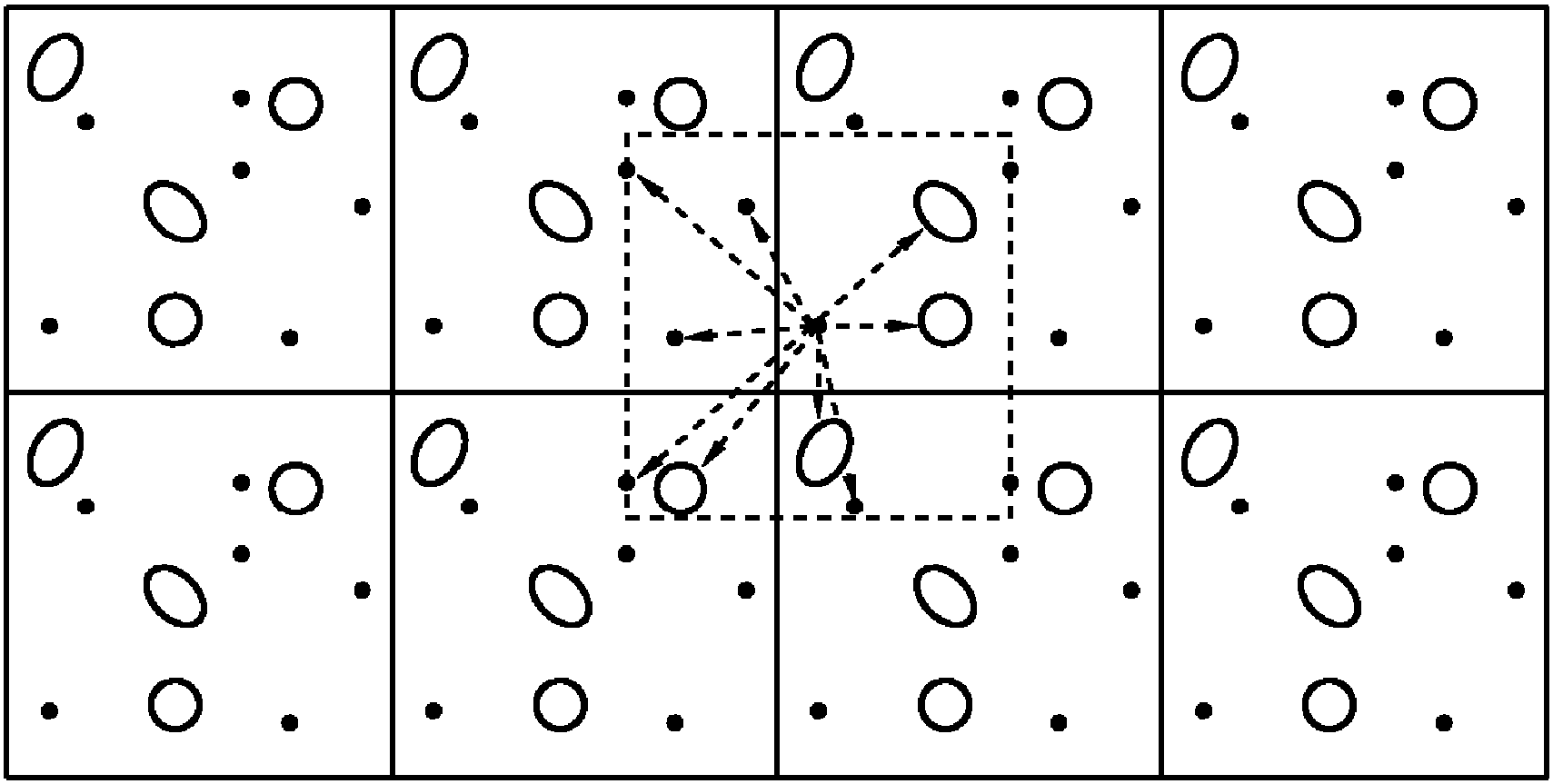
Algorithm 6 (Integrating the Equations of Motion)

```
subroutine integrate(f,en)          integrate equations of motion
sumv=0
sumv2=0
do i=1,npart                        MD loop
   xx=2*x(i)-xm(i)+delt**2*f(i)     Verlet algorithm (4.2.3)
   vi=(xx-xm(i))/(2*delt)           velocity (4.2.4)
   sumv=sumv+vi                     velocity center of mass
   sumv2=sumv2+vi**2                total kinetic energy
   xm(i)=x(i)                       update positions previous time
   x(i)=xx                          update positions current time
enddo
temp=sumv2/(3*npart)                instantaneous temperature
etot=(en+0.5*sumv2)/npart           total energy per particle
return
end
```

Molecular Dynamics

- Initialization
  - Total momentum should be zero (no external forces)
  - Temperature rescaling to desired temperature
  - Particles start on a lattice
- Force calculations
  - Periodic boundary conditions
  - Order NxN algorithm,
  - Order N: neighbor lists, linked cell
  - Truncation and shift of the potential
- Integrating the equations of motion
  - Velocity Verlet
  - Kinetic energy

Periodic boundary conditions



Lennard Jones potentials

- The Lennard-Jones potential

$$u^{LJ}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

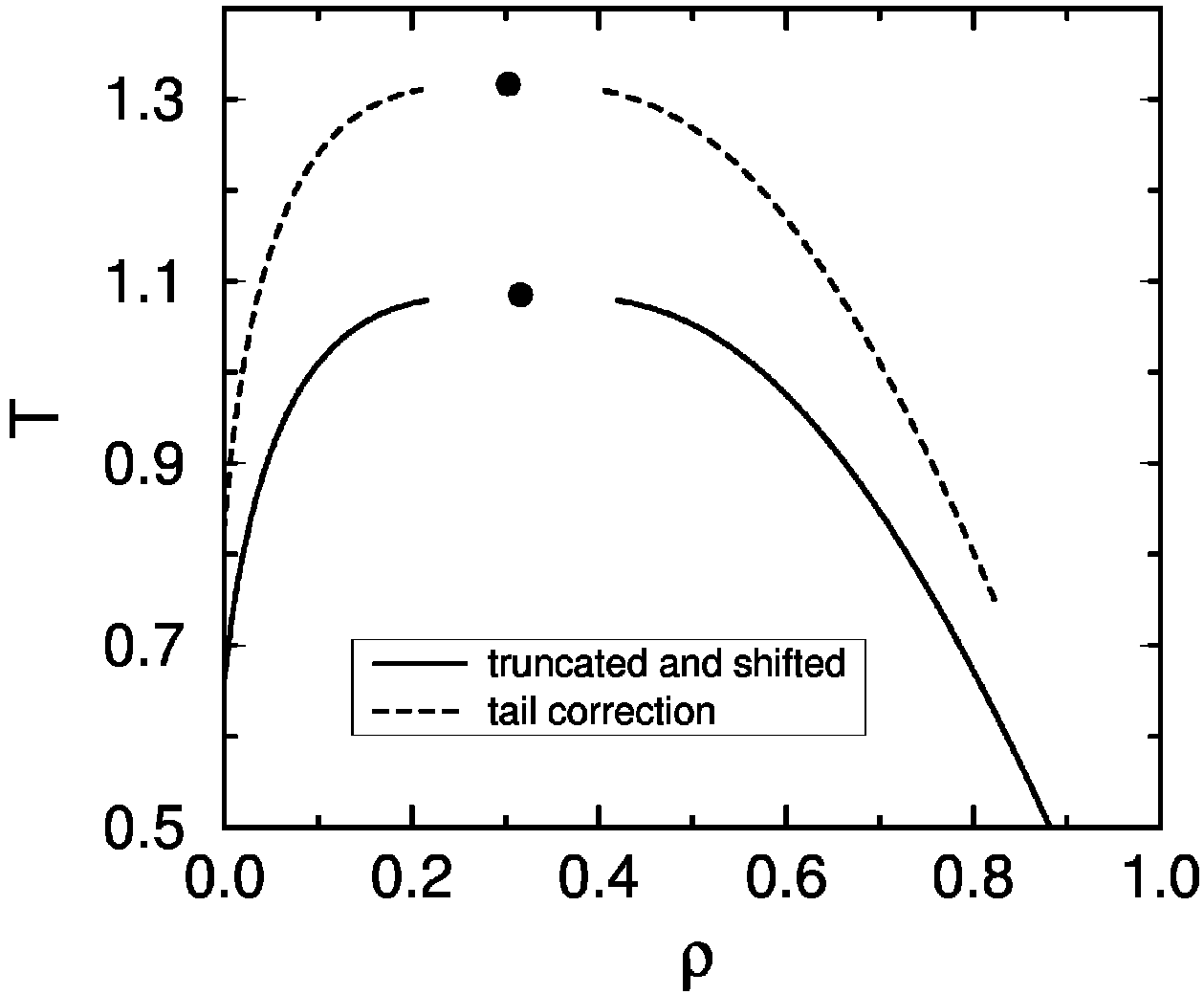
- The truncated Lennard-Jones potential

$$u(r) = \begin{cases} u^{LJ}(r) & r \le r_c \\ 0 & r > r_c \end{cases}$$

- The truncated and shifted Lennard-Jones potential

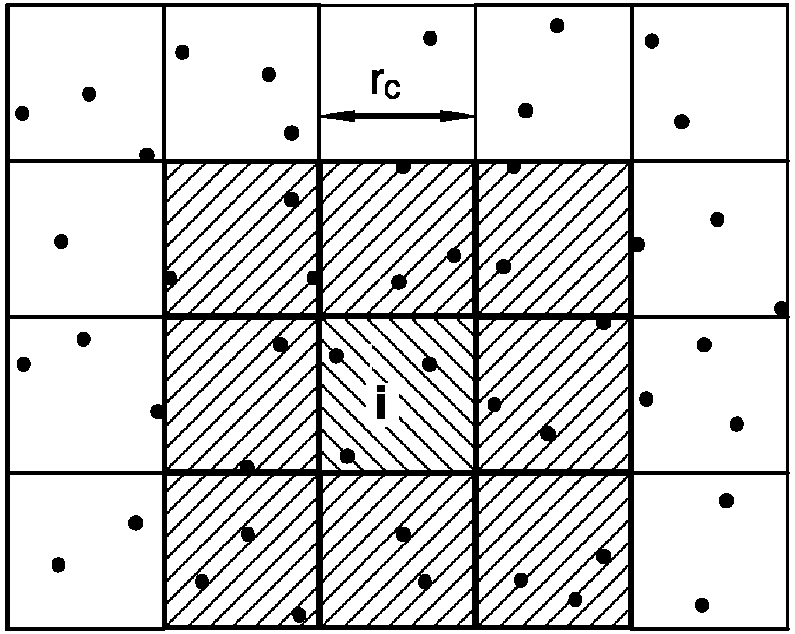
$$u(r) = \begin{cases} u^{LJ}(r) - u^{LJ}(r_c) & r \le r_c \\ 0 & r > r_c \end{cases}$$

Phase diagrams of Lennard Jones fluids

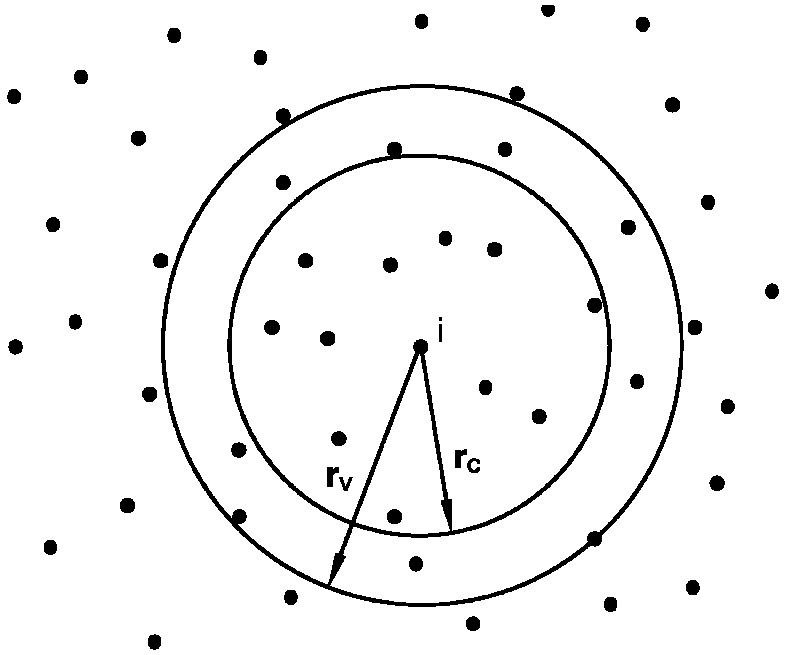


Saving cpu time

Cell list



Verlet-list



Equations of motion

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\Delta t^2}{2m}\mathbf{f}(t) + \frac{\Delta t^3}{3!m}\dot{\mathbf{f}}(t) + \mathcal{O}(\Delta t^4)$$

$$\mathbf{r}(t-\Delta t) = \mathbf{r}(t) - \mathbf{v}(t)\Delta t + \frac{\Delta t^2}{2m}\mathbf{f}(t) - \frac{\Delta t^3}{3!m}\dot{\mathbf{f}}(t) + \mathcal{O}(\Delta t^4)$$

$$\mathbf{r}(t+\Delta t) + \mathbf{r}(t-\Delta t) = 2\mathbf{r}(t) + \frac{\Delta t^2}{m}\mathbf{f}(t) + \mathcal{O}(\Delta t^4)$$

Verlet algorithm

$$\mathbf{r}(t+\Delta t) \approx 2\mathbf{r}(t) - \mathbf{r}(t-\Delta t) + \frac{\Delta t^2}{m}\mathbf{f}(t)$$

Velocity Verlet algorithm

$$\mathbf{r}(t+\Delta t) \approx \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\Delta t^2}{2m}\mathbf{f}(t)$$

$$\mathbf{v}(t+\Delta t) \approx \mathbf{v}(t) + \frac{\Delta t}{2m}\left[\mathbf{f}(t+\Delta t) + \mathbf{f}(t)\right]$$

Liouville formulation

$$f\left(\mathbf{p}^N, \mathbf{r}^N\right)$$

Depends implicitly on t

$$\dot{f} = \dot{\mathbf{r}}\frac{\partial f}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial f}{\partial \mathbf{p}}$$

$$\mathrm{i}L \equiv \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$$

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \mathrm{i}Lf$$

Beware: this solution is equally useless as the differential equation!

Solution

$$f\left(t\right) = \exp\left(\mathrm{i}Lt\right) f\left(0\right)$$

$$\mathrm{i}L \equiv \mathrm{i}L_r + \mathrm{i}L_p = \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$$

Let us look at them separately

$$f(t) = \exp(\mathrm{i}L_r t) f(0)$$

$$= \exp\left(\dot{\mathbf{r}}(0) t \frac{\partial}{\partial \mathbf{r}}\right) f(0)$$

$$= \sum_{n=0}^{\infty} \frac{(\dot{\mathbf{r}}(0)t)^n}{n!} \frac{\partial^n}{\partial \mathbf{r}^n} f(0)$$

Taylor expansion

$$= f\left(\mathbf{p}^N(0), (\mathbf{r}(0) + \dot{\mathbf{r}}(0)t)^N\right)$$

Shift of coordinates

$$\mathbf{r}(0) \to \mathbf{r}(0) + \dot{\mathbf{r}}(0)t$$

$$= \exp\left(\dot{\mathbf{p}}(0) t \frac{\partial}{\partial \mathbf{p}}\right) f(0)$$

$$= \sum_{n=0} \frac{}{n!} \frac{\partial^n}{\partial \mathbf{p}^n} f(0)$$

$$= f\left((\mathbf{p}(0) + \dot{\mathbf{p}}(0)t)^N, \mathbf{r}^N(0)\right)$$

Shift of momenta

$$\mathbf{p}(0) \to \mathbf{p}(0) + \dot{\mathbf{p}}(0)t$$

$$\mathrm{i}L_r \Rightarrow \mathbf{r}(0) \rightarrow \mathbf{r}(0) + \dot{\mathbf{r}}(0)t$$

$$\mathrm{i}L_p \Rightarrow \mathbf{p}(0) \rightarrow \mathbf{p}(0) + \dot{\mathbf{p}}(0)t$$

$$f\left(\mathbf{p}^N(t), \mathbf{r}^N(t)\right) = \left(e^{\mathrm{i}L_p t/2P} e^{\mathrm{i}L_r t/P} e^{\mathrm{i}L_p t/2P}\right)^P f\left(\mathbf{p}^N(0), \mathbf{r}^N(0)\right)$$

We have *noncommuting* operators!

$$e^{A+B} \neq e^A e^B$$

Trotter identity

$$e^{A+B} = \lim_{P\to\infty}\left(e^{A/2P} e^{B/P} e^{A/2P}\right)^P$$

$$e^{A+B} \approx \left(e^{A/2P} e^{B/P} e^{A/2P}\right)^P$$

$$\frac{A}{P} = \frac{\mathrm{i}L_p t}{P} \qquad \frac{B}{P} = \frac{\mathrm{i}L_r t}{P} \qquad \Delta t = \frac{t}{P}$$

$$\mathrm{i}L_r\Delta t \Rightarrow \mathbf{r} \rightarrow \mathbf{r} + \dot{\mathbf{r}}\Delta t$$

$$\mathrm{i}L_p\Delta t \Rightarrow \mathbf{p} \rightarrow \mathbf{p} + \dot{\mathbf{p}}\Delta t$$

$$\left(e^{(\mathrm{i}L_p\Delta t/2)} e^{(\mathrm{i}L_r\Delta t)} e^{(\mathrm{i}L_p\Delta t/2)} \right)^P$$

$$e^{(\mathrm{i}L_p\Delta t/2)} f\left(\mathbf{p}^N(0), \mathbf{r}^N(0)\right) = f\left(\left[\mathbf{p}(0) + \frac{\Delta t}{2}\dot{\mathbf{p}}(0) \right]^N, \mathbf{r}^N(0) \right)$$

$$\left[\mathbf{r}(0) + \Delta t \dot{\mathbf{r}}\left(\frac{\Delta t}{2} \right) \right]^N \Bigg)$$

.....

$$\left] \Bigg)^N \right)$$

Velocity Verlet!

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2m}\mathbf{F}(t)\Delta t^2$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{\Delta t}{2m}\left[\mathbf{F}(t) + \mathbf{F}(t+\Delta t)\right]$$

$$\mathbf{p}(0) \rightarrow \mathbf{p}(0) + \frac{\Delta t}{2}\left[\dot{\mathbf{p}}(0) + \dot{\mathbf{p}}(\Delta t)\right]$$

$$\mathbf{r}(0) \rightarrow \mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\Delta t/2) = \mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(0) + \frac{\Delta t^2}{2m}\mathbf{F}(0)$$

Velocity Verlet:

$$e^{\left(iL_p \Delta t/2\right)} e^{\left(iL_r \Delta t\right)} e^{\left(iL_p \Delta t/2\right)}$$

```
Call force(fx)
Do while (t<tmax)
```

$$e^{\left(iL_p \Delta t/2\right)} : \mathbf{v}\left(t+\frac{\Delta t}{2}\right) \rightarrow \mathbf{v}(t)+\frac{\Delta t}{2m}\mathbf{f}(t)$$

```
vx=vx+delt*fx/2
```

$$e^{\left(iL_r \Delta t\right)} : \mathbf{r}(t+\Delta t) \rightarrow \mathbf{r}(t)+\Delta t \mathbf{v}(t+\Delta t/2)$$

```
x=x+delt*vx
Call force(fx)
```

$$e^{\left(iL_p \Delta t/2\right)} : \mathbf{v}(t+\Delta t) \rightarrow \mathbf{v}(t+\Delta t/2)+\frac{\Delta t}{2m}\mathbf{f}(t+\Delta t)$$

```
vx=vx+delt*fx/2
enddo
```

Liouville Formulation

Velocity Verlet algorithm:

$$\mathbf{p}(t+\Delta t)=\mathbf{p}(t)+\frac{\Delta t}{2}\left[\dot{\mathbf{p}}(t)+\dot{\mathbf{p}}(t+\Delta t)\right]$$

$$\mathbf{r}(t+\Delta t)=\mathbf{r}(t)+\Delta t\dot{\mathbf{r}}(t)+\frac{\Delta t^2}{2m}\mathbf{F}(t)$$

Three subsequent coordinate transformations in either **r** or **p** of which the *Jacobian* is one: *Area preserving*

$$\mathbf{p}(t+\Delta t/2)=\mathbf{p}(t)+\frac{\Delta t}{2}\mathbf{F}(\mathbf{r})$$
$$\mathbf{r}(t)=\mathbf{r}(t)$$
$$J_1=\det\begin{vmatrix}1 & \frac{\Delta t}{2}\frac{\partial\mathbf{F}(\mathbf{r})}{\partial\mathbf{r}}\\ 0 & 1\end{vmatrix}=1$$

$$\mathbf{p}(t+\Delta t/2)=\mathbf{p}(t+\Delta t/2)$$
$$\mathbf{r}(t+\Delta t)=\mathbf{r}(t)+\frac{\Delta t}{m}\mathbf{p}(t+\Delta t/2)$$
$$J_2=\det\begin{vmatrix}1 & 0\\ \Delta t/m & 1\end{vmatrix}=1$$

$$\mathbf{p}(t+\Delta t)=\mathbf{p}(t+\Delta t/2)+\frac{\Delta t}{2}\mathbf{F}(\mathbf{r}(t))$$
$$\mathbf{r}(t)=\mathbf{r}(t)$$
$$J_3=\det\begin{vmatrix}1 & \frac{\Delta t}{2}\frac{\partial\mathbf{F}(\mathbf{r})}{\partial\mathbf{r}}\\ 0 & 1\end{vmatrix}=1$$

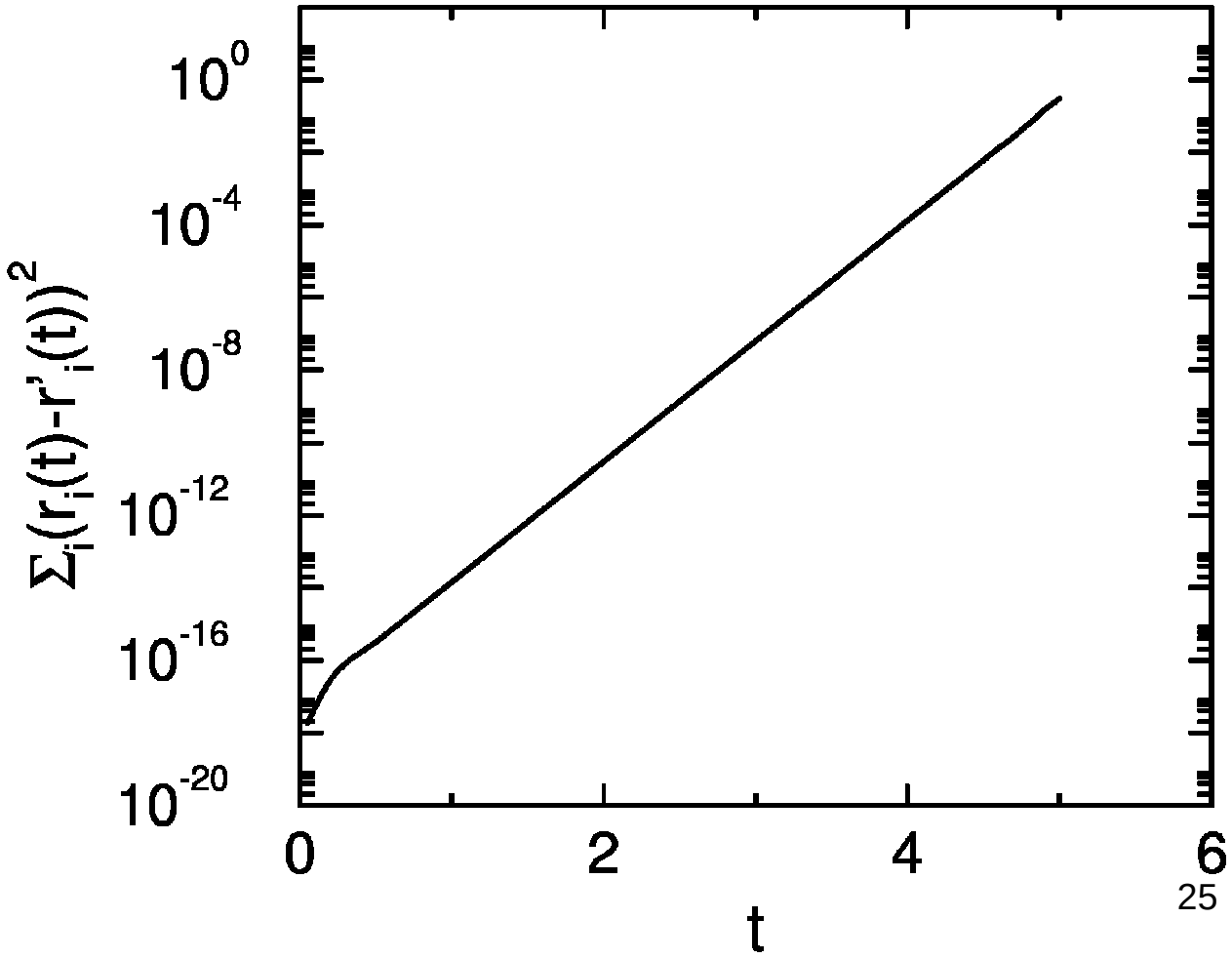
Other Trotter decompositions are possible!

Lyaponov instability

$\left(\mathbf{r}^N(0), \mathbf{p}^N(0)\right)$

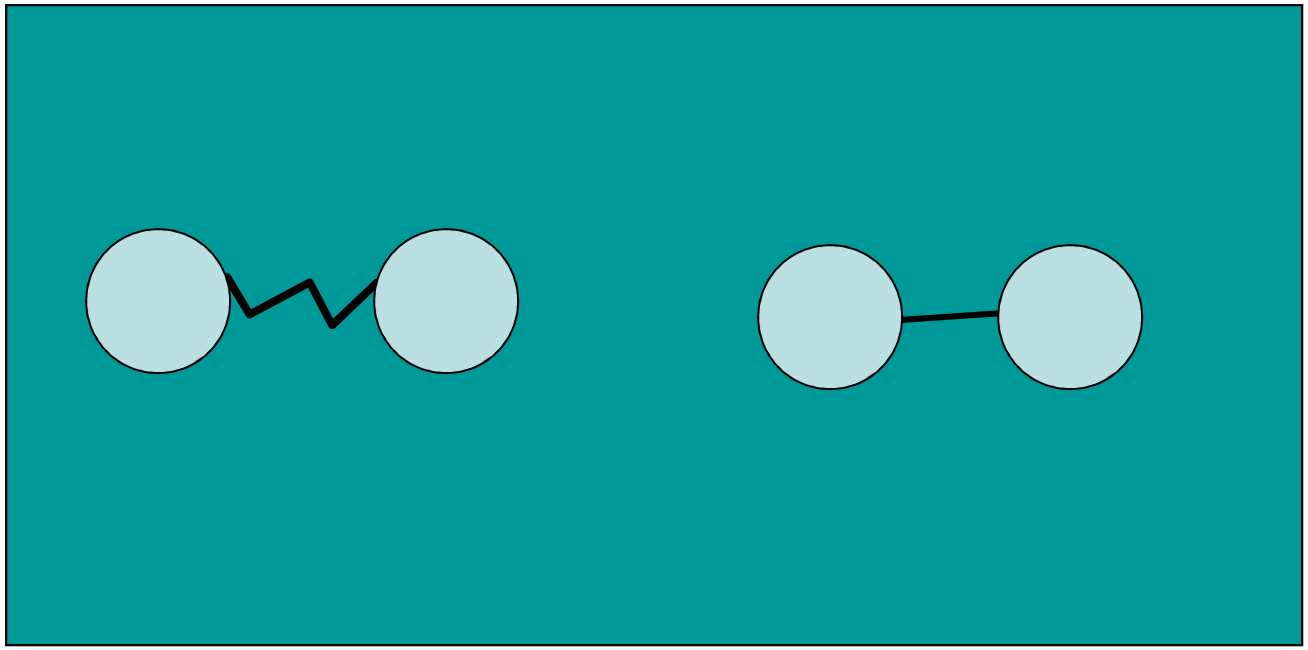
$\left(\mathbf{r}^N(0), \mathbf{p}_1(0), \cdots, \mathbf{p}_j(0)+\varepsilon, \mathbf{p}_i(0)-\varepsilon \cdots, \mathbf{p}_N(0)\right)$

$\varepsilon = 10^{-10}$



Multiple time steps

- What to use for stiff potentials:



  - Fixed bond-length: constraints (Shake)
  - Very small time step

Multiple Time steps

$$\mathbf{F} = \mathbf{F}_{\text{short}} + \mathbf{F}_{\text{long}}$$

$$\mathrm{i}L \equiv \mathrm{i}L_r + \mathrm{i}L_p = \mathbf{v}\frac{\partial}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m}\frac{\partial}{\partial \mathbf{v}}$$

$$\mathrm{i}L_p = \mathrm{i}L_{\text{short}} + \mathrm{i}L_{\text{long}}$$

$$\mathrm{i}L_{\text{short}} = \frac{\mathbf{F}_{\text{short}}}{m}\frac{\partial}{\partial \mathbf{v}} \qquad \mathrm{i}L_{\text{long}} = \frac{\mathbf{F}_{\text{long}}}{m}\frac{\partial}{\partial \mathbf{v}}$$

Trotter expansion:

$$e^{\mathrm{i}\left(L_{\text{long}}+L_{\text{short}}+L_r\right)\Delta t} \approx e^{\mathrm{i}L_{\text{long}}\Delta t/2} e^{\mathrm{i}\left(L_{\text{short}}+L_r\right)\Delta t} e^{\mathrm{i}L_{\text{long}}\Delta t/2}$$

Introduce: $\delta t = \Delta t/n$

$$e^{\mathrm{i}\left(L_{\text{long}}+L_{\text{short}}+L_r\right)\Delta t} \approx e^{\mathrm{i}L_{\text{long}}\Delta t/2}\left[e^{\mathrm{i}L_{\text{short}}\delta t/2} e^{\mathrm{i}L_r\delta t} e^{\mathrm{i}L_{\text{short}}\delta t/2}\right]^n e^{\mathrm{i}L_{\text{long}}\Delta t/2}$$

$$e^{i\left(L_{\text{long}}+L_{\text{short}}+L_r\right)\Delta t} \approx e^{iL_{\text{long}}\,\Delta t/2}\left[e^{iL_{\text{short}}\,\delta t/2}e^{iL_r\delta t}e^{iL_{\text{short}}\,\delta t/2}\right]^n e^{iL_{\text{long}}\,\Delta t/2}$$

$$iL_{\text{long}}\,\Delta t/2 \Rightarrow v \to v + F_{\text{long}}\,\Delta t/2m$$

$$iL_{\text{short}}\,\delta t/2 \Rightarrow v \to v + F_{\text{short}}\,\delta t/2m$$

$$iL_r\delta t \Rightarrow r \to r + v\delta t$$

First

$$e^{iL_{\text{long}}\,\Delta t/2} f\left[r(0), v(0)\right] = f\left[r(0), v(0) + F_{\text{long}}(0)\Delta t/2m\right]$$

Now *n* times:

$$\left[e^{iL_{\text{short}}\,\delta t/2}e^{iL_r\delta t}e^{iL_{\text{short}}\,\delta t/2}\right]^n f\left[r(0), v(0) + F_{\text{long}}(0)\Delta t/2m\right]$$

$$e^{\left(iL_{p\mathrm{Long}}\Delta t/2\right)}:\mathbf{v}\left(t+\frac{\Delta t}{2}\right)\rightarrow\mathbf{v}\left(t\right)+\frac{\Delta t}{2m}\mathbf{f}_{\mathrm{long}}\left(t\right)$$

```
Call force(fx_l
vx=vx+delt*fx_l
Do ddt=1,n
```

$$e^{\left(iL_{p\mathrm{Short}}\delta t/2\right)}:\mathbf{v}\left(t+\right.$$

```
   vx=vx+ddelt*fx_short/2
```

$$e^{\left(iL_{r}\delta t\right)}:\mathbf{r}\left(t+\delta t\right)\rightarrow\mathbf{r}\left(t\right)+\delta t\mathbf{v}\left(t+\Delta t/2+\delta t/2\right)$$

```
   x=x+ddelt*vx
   Call force
```

$$e^{\left(iL_{p\mathrm{Short}}\delta t/2\right)}:\mathbf{v}\left(t+\right.$$

```
   vx=vx+ddel
enddo
```

$$e^{iL_{\mathrm{long}}\Delta t/2}\left[e^{iL_{\mathrm{short}}\delta t/2}e^{iL_{r}\delta t}e^{iL_{\mathrm{short}}\delta t/2}\right]^{n}e^{iL_{\mathrm{long}}\Delta t/2}$$

$$iL_{\mathrm{long}}\,\Delta t/2\Rightarrow v\rightarrow v+F_{\mathrm{long}}\,\Delta t/2m$$

$$iL_{\mathrm{short}}\,\delta t/2\Rightarrow v\rightarrow v+F_{\mathrm{short}}\,\delta t/2m$$

$$iL_{r}\delta t\Rightarrow r\rightarrow r+v\delta t$$

$$e^{iL_{\mathrm{long}}\Delta t/2}\left[e^{iL_{\mathrm{short}}\delta t/2}e^{iL_{r}\delta t}e^{iL_{\mathrm{short}}\delta t/2}\right]^{n}e^{iL_{\mathrm{long}}\Delta t/2}$$

$$iL_{\mathrm{long}}\,\Delta t/2\Rightarrow v\rightarrow v+F_{\mathrm{long}}\,\Delta t/2m$$

$$iL_{\mathrm{short}}\,\delta t/2\Rightarrow v\rightarrow v+F_{\mathrm{short}}\,\delta t/2m$$

$$iL_{r}\delta t\Rightarrow r\rightarrow r+v\delta t$$

Algorithm 29 (Multiple Time Step)

```
 subroutine
+     multi(f_long,f_short)

 vx=vx+0.5*delt*f_long
 do it=1,n
    vx=vx+0.5*(delt/n)*f_short
    x=x+(delt/n)2*vx
    call force_short(f_short)
    vx=vx+0.5*(delt/n)*f_short
 enddo
 call force_all(f_long,f_short)
 vx=vx+0.5*delt*f_long
 return
 end
```

Multiple time step, `f_long` is the long-range part and `f_short` the short-range part of the force

velocity Verlet with time step Δt

loop for the small time step

velocity Verlet with timestep $\Delta t/n$

short-range forces

all forces

GPUs are outpacing CPUs…

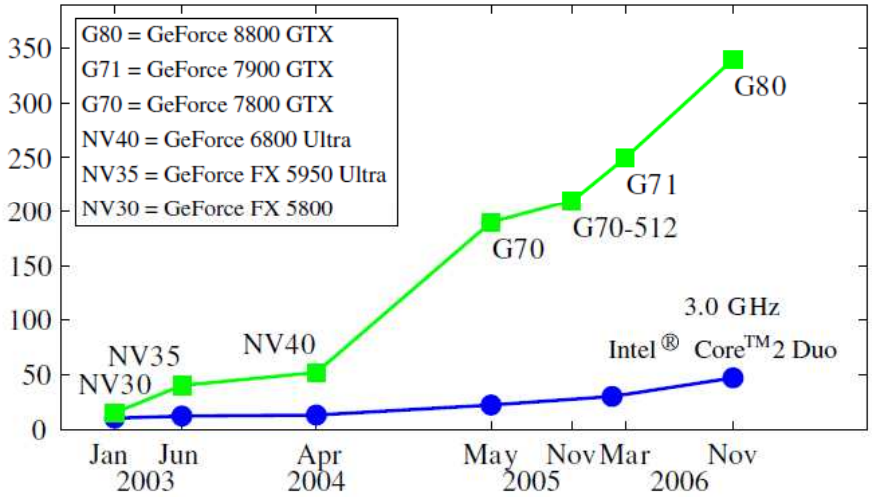
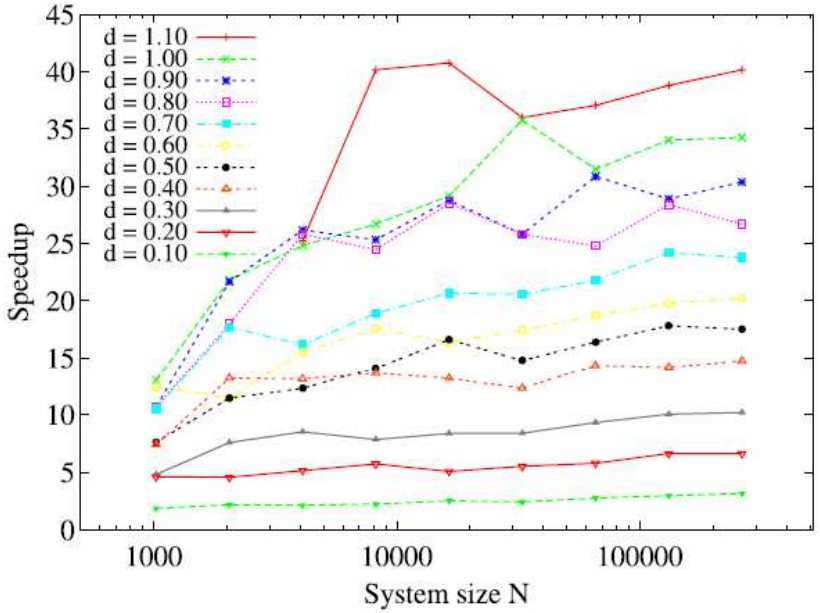
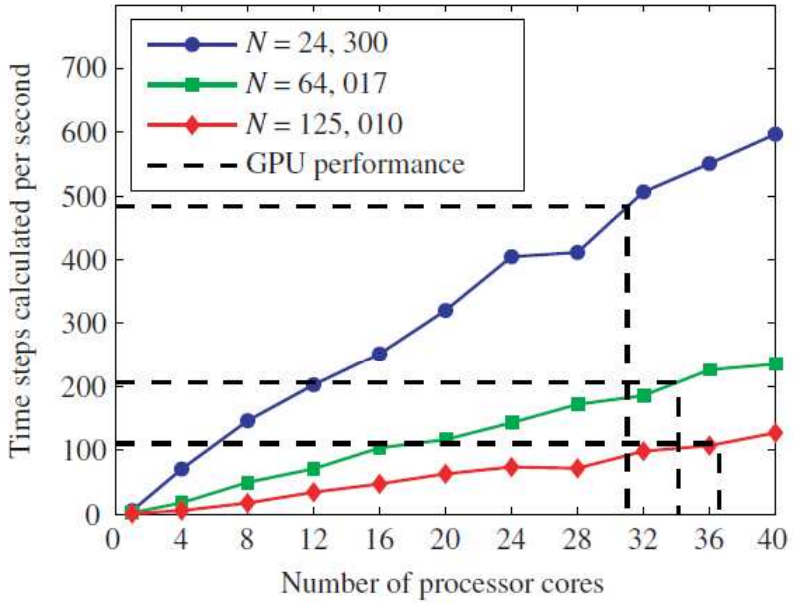


Fig. 1. Performance of CPUs (blue circles) and GPUs (green squares) over the last few years. Figure courtesy of NVIDIA, and adapted from Ref. [1]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Which is good for MD!





van Meel *et al.* Mol. Sim. 31, 259 (2008).

Anderson, Lorentz, and Travesset, J. Comput. Phys. 227, 5342 (2008).