Welcome

Winterschool on
Theoretical Chemistry and Spectroscopy
6-10 December 2010, Han-sur-Lesse, Belgium

program

- Prof. Dr. Ria Broer (RUG)
  Electronic structure methods for solids (6 hours)

- Prof. Dr. Frank de Groot (UU)
  Theoretical and experimental aspects of X-ray spectroscopy (6 hours)

- Dr. Francesco Buda (UL)
  Ab initio molecular dynamics: ground and excited states (4 hours)

- Dr. Johan Padding (IMCN)
  Statistical mechanics of liquids (6 hours)

- Dr. Bernd Ensing (UvA)
  Free energy and multiscale methods (4 hours)

program

- **Carolin König:** Absorption Spectra of Photosynthetic Complexes from First Principles Calculations.
- **Anna Pavlova:** The role of water in ruthenium catalyzed transfer hydrogenation, a computational study.
- **Alexander Atamas:** Monte Carlo calculations of the free energy of ice-like structures
- **Leendertjan Karssemeijer:** Thermal expansion and lattice dynamics of carbon nanotubes and other graphitic systems derived from LCBOPII
- **Murat Kilic:** Redox properties of aqueous lumiflavin - a first molecular dynamic study
- **Zahid Rashid:** Resonance and Aromaticity of some polycyclic conjugated systems
- **Pawel Tecmer:** Electronic spectroscopy of $UO2^{2+}$, NUO^{+} and NUN: An evaluation of Time-Dependent Density Functional Theory for actinides

program

	9:00-11:00	11:00-13:00	16:00-17:00	17:00-19:00	20:45-22:00
Monday				Ensing	Broer + Presentations I
Tuesday	Broer	Ensing	Broer	Buda	Presentations II
Wednesday	Padding	Broer		De Groot	Presentations III
Thursday	Padding	Buda		De Groot	Excercises
Friday	Padding	De Groot	departure after lunch		

Free energy and Multiscale Methods

Winterschool on
Theoretical Chemistry and Spectroscopy
6-10 December 2010, Han-sur-Lesse, Belgium

Bernd Ensing

Motivation

When is Free Energy relevant?

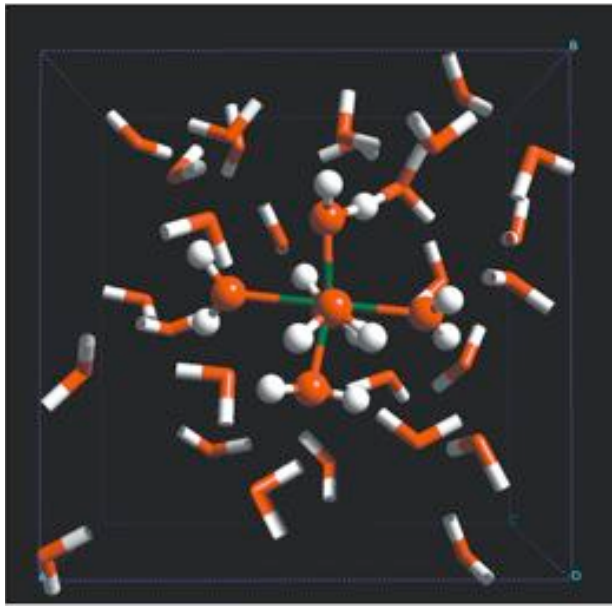
- In principle for all processes
- Except when the number of states is invariant
  - chemical reaction in vacuo at low temperature
  - electronic excitation

- Especially systems with (many) soft modes
  - liquid water
  - bio-molecules
  - polymers
  - soft matter in general

Example: acidity of Fe(III) (aq)

$$Fe^{III}(H_2O)_6^{3+} + H_2O \longleftrightarrow Fe^{III}(H_2O)_5(OH)^{2+} + H_3O^+$$

$$pKa = -\log\left(\frac{[B^-][H_3O^+]}{[HB]}\right) = -\log\left(\frac{(x(H_3O^+))^2[HB]_0}{1 - x(H_3O^+)}\right)$$



Method	$x(H_3O)^+$	kA	pkA
CPMD	0.130	0.034	1.47
static DFT		10,000	-4.00
expt		0.006	2.2

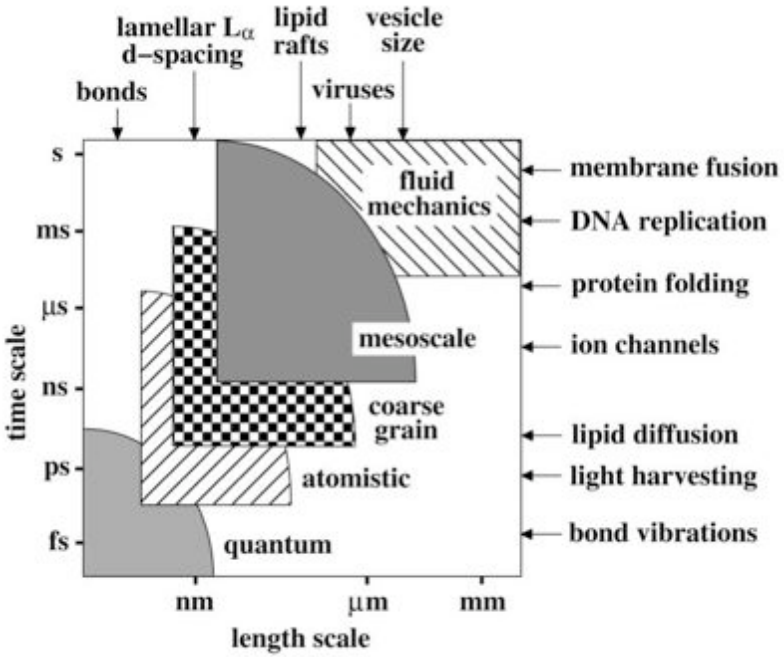
Calculation of the acidity constant of Iron(III) in water requires averaging over many solvent configurations. Optimizing the reactant and product states in a water cluster results in a poor result. Fortunately this process is not a *rare event* on the Car-Parrinello molecular dynamics time scale...

Motivation

What about Multiscale methods?

- Used to save computer time
- Cannot solve Schrödinger's equation for large systems
- Separation of time scales
  (rare event problem)
- Separation of energy scales

- Examples
  - hybrid QM/MM
  - multigrid methods
  - reaction coordinate



Outline

Lecture 1: Integrators

Lecture 2: Multi-time step

Lecture 3: Chemical reactions

Lecture 4: Free energy methods

Integrators

Algorithms to numerically solve ordinary differential equations (ODE)

ODE is an equation that relates the function value to the value(s) of its derivatives

Examples of applications:

- Calculation of an integral
- Solving Schrödingers equation: $E\psi(r) = -\frac{\hbar^2}{2m}\nabla^2\psi(r) + V(r)\psi(r)$
- Geometry optimization
- Molecular dynamics $H = \frac{1}{2}\sum_i m_i \dot{r}_i^2 + V(r)$

Examples of integrators:

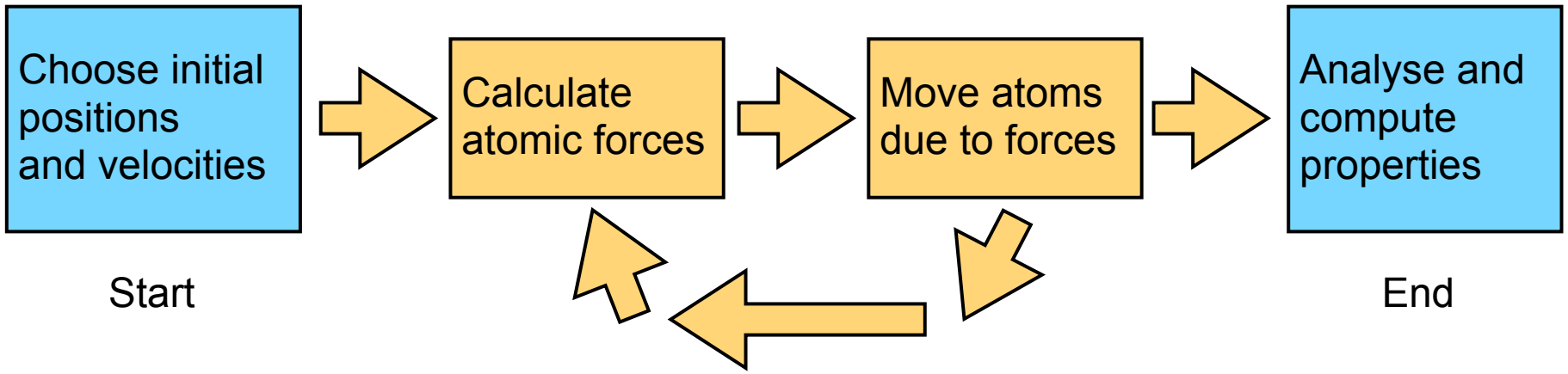
- Euler method
- Runga-Kutta
- Verlet
- Velocity Verlet
- Leap-frog
- Beeman's algorithm
- ...

molecular dynamics

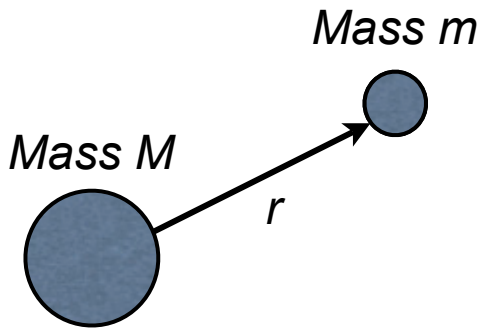
Integration of Newton's equations

$$F = ma$$
$$F = -\nabla V$$

$$-\frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2}$$



Example

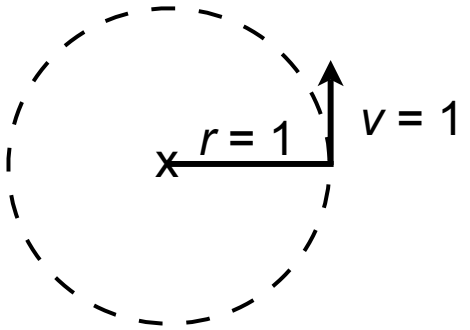


- Newton's Law of Gravitation
  - force is an inverse square law
  - same equations of motion as MD

$$F = -\frac{GMm}{r^2}$$

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{1}{r^2}\hat{\mathbf{r}}$$

- Simple Numerical Model in Reduced Units:
  - Assume Sun is stationary (M >> m)
  - For convenience we use Earth Units
    - GM=1
    - circular orbit for r=v=1
    - each revolution takes 2π time-units



Integrator (1): Euler method

- Truncate Taylor expansion after the acceleration term
  - Local Error: $O(h^3)$ in position and $O(h^2)$ in velocity

$$\begin{aligned} \mathbf{x}(t+h) &= \mathbf{x}(t) + h\mathbf{v}(t) + \frac{h^2}{2}\mathbf{a}(t) \\ \mathbf{v}(t+h) &= \mathbf{v}(t) + h\mathbf{a}(t) \end{aligned}$$

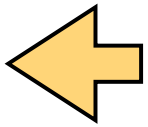
- Euler method is OK for projectiles, but for MD ...

Integrator (1): Euler method

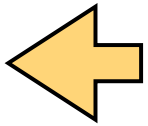
Matlab/Octave script

```
h=0.1;          % timestep
pos=[1 0];      % initial position
vel=[0 1.0];    % initial velocity
plot(1,0,'g-',pos(1),pos(2),'ko',0,0,'ro')

for i=1:100
  x(i)=pos(1);
  y(i)=pos(2);

  plot(x,y,'g-',pos(1),pos(2),'k*',0,0,'r+')
  title(num2str(i*h))
  axis equal;
  axis([-2.0 2.0 -2.0 2.0]);
  pause(0.05);

  r=norm(pos);
  accel=-1/r^2 * pos/r;

  pos=pos + h*vel + 0.5*h*h*accel;
  vel=vel + h*accel;
end
```

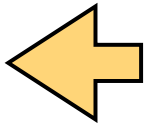
Consider:
- h=0.1 ; steps=100
- h=0.05 ; steps=200



Plotting instructions

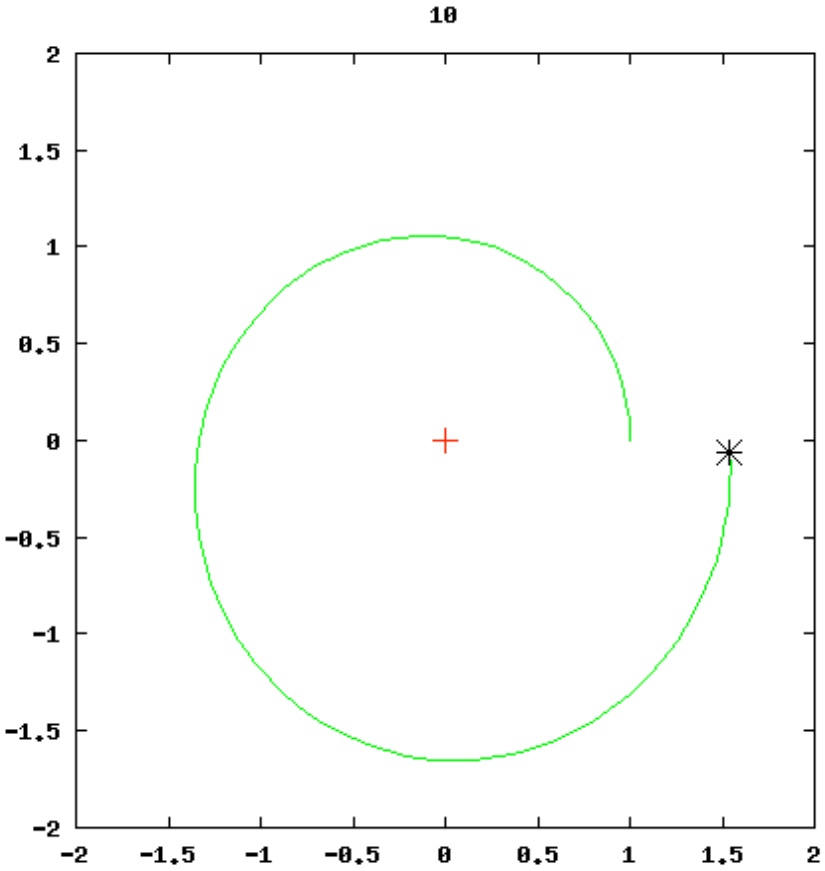


Compute force



Integrate

Integrator (1): Euler method



- Euler method is OK for projectiles, but for MD ...
  - trajectory spirals outward
  - local error accumulates with time (i.e. large global error)
  - reducing timestep *h* just delays the inevitable

Improved integrators

- Symplectic integrators correctly reproduce long-time dynamics
- Common ODE methods such as Runge-Kutta are not suitable
  - error always accumulates in a manner analogous to the Euler example → unreliable long-term behaviour
- A good MD integrator should:
  - be time reversible (thus honouring Newtonian mechanics)
  - conserve phase-space volume (pendulum is illustrative)

Integrator (2): Verlet method

- Combine forward and backward Taylor expansions
  - Local Error: $O(h^4)$ in position and $O(h^2)$ in velocity

$$\begin{aligned} \mathbf{x}(t+h) &= 2\mathbf{x}(t) - \mathbf{x}(t-h) + h^2\mathbf{a}(t) \\ \mathbf{v}(t) &= \frac{1}{2h}[\mathbf{x}(t+h) - \mathbf{x}(t-h)] \end{aligned}$$

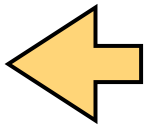
- Velocities not required: [x(t–h),x(t)] → [x(t+h)]
- Not self-starting – apply a single Euler step to begin
  - stable even with large time-steps
  - local error does not accumulate (i.e. small global error)

Integrator (2): Verlet method

```
h=0.2;          % timestep
pos=[1 0];      % initial position
vel=[0 1.0];    % initial velocity

for i=1:50
  x(i)=pos(1);
  y(i)=pos(2);

  ...           % plotting instructions

  r=norm(pos);
  accel=-1/r^2 * pos/r;

  if i==1
    next=pos + h*vel + 0.5*h*h*accel;
  else
    next=2*pos - prev + h*h*accel;
    vel=(next-prev)/(2*h);
  end

  Etot(i)=0.5*norm(vel)^2 + 1/norm(pos);

  prev=pos;
  pos=next;
end
```

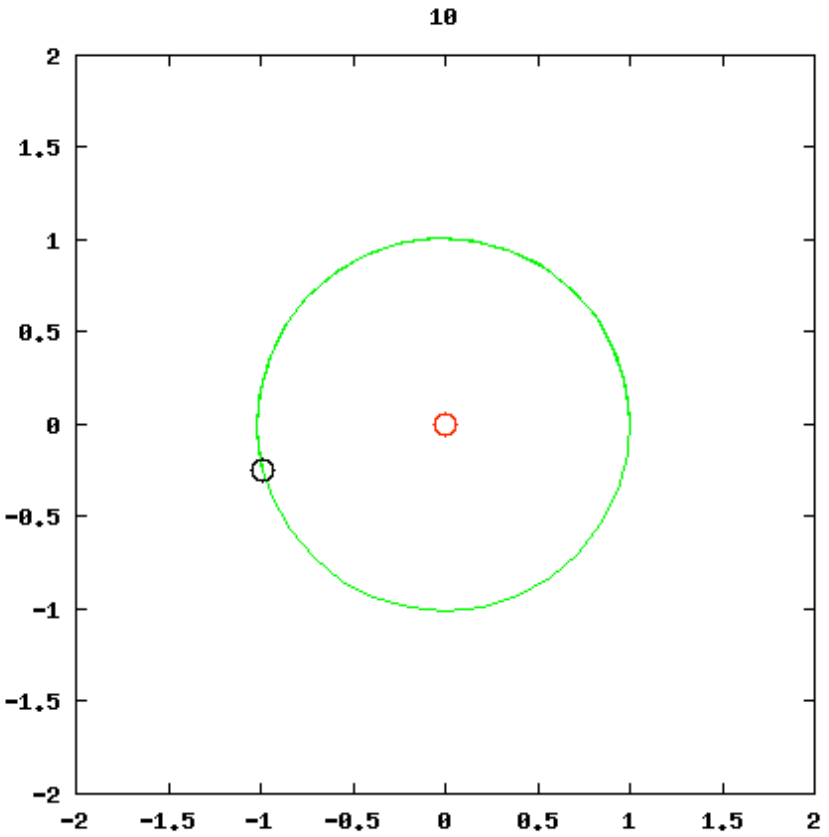
Matlab/Octave script

Consider:
- h=0.1 ; steps=100
- h=0.5 ; steps=100
- h=1.5 ; steps=50



Conserved quantity

Integrator (2): Verlet method



$$m\frac{d^2x}{dt^2} = F(x) \qquad \frac{dv}{dt} = \frac{dv}{dx}\frac{dx}{dt}$$

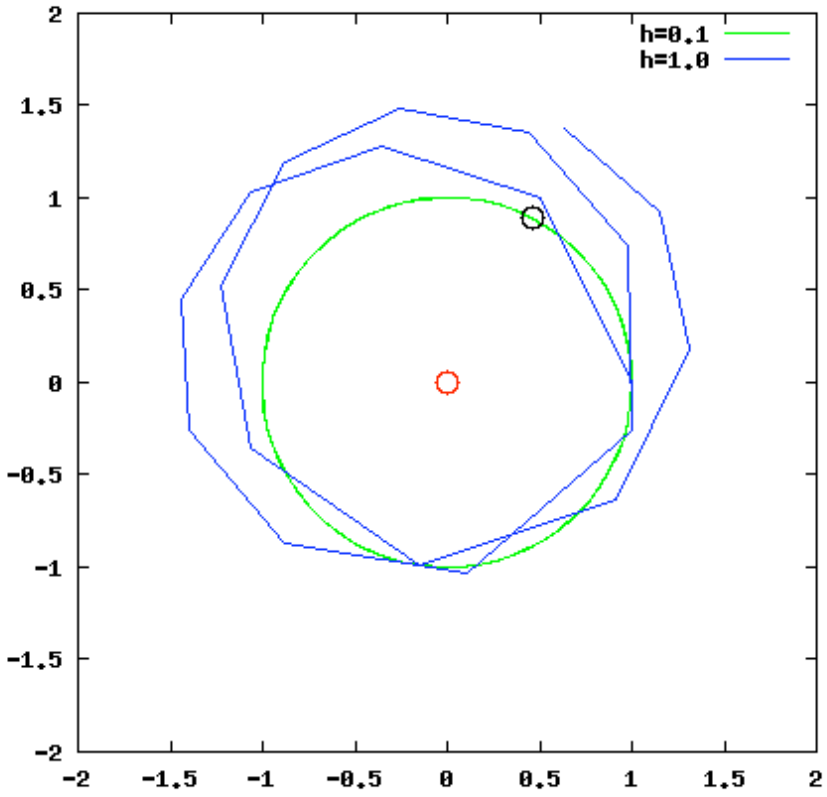
$$mv\frac{dv}{dx} = F(x)$$

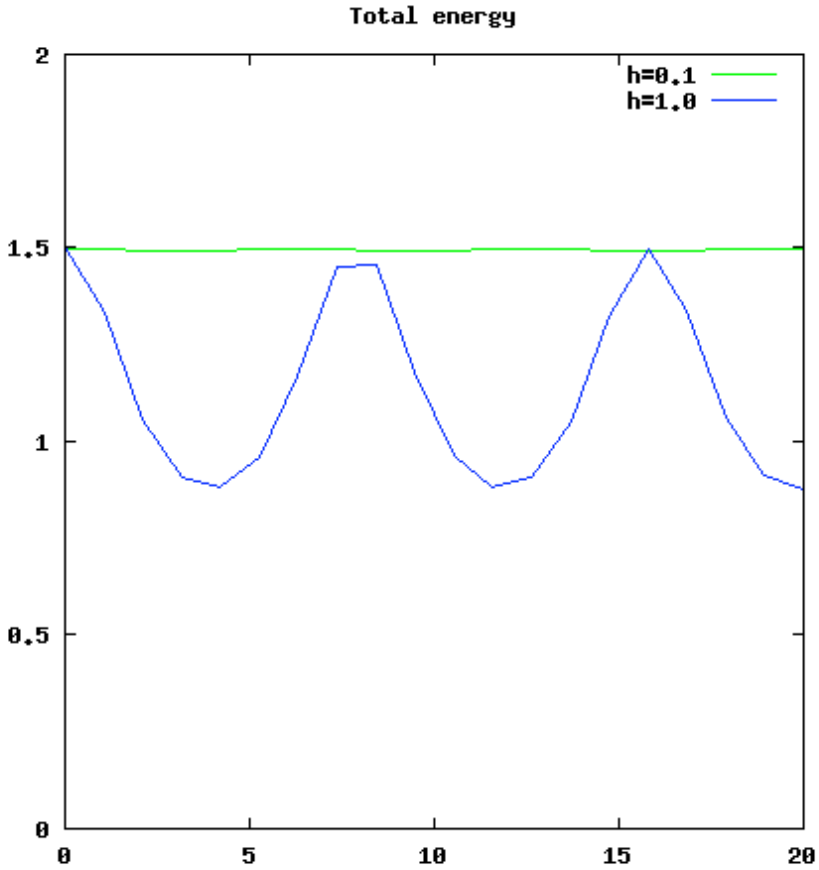
$$\frac{1}{2}mv^2 - \int F(x)dx = \text{constant}$$

$$U(r) \equiv -\int F(r)dr$$

$$= \int \frac{1}{r^2}dr \; = -\frac{1}{r}$$

Integrator (2): Verlet method





conserved quantities

- We can't *exactly* numerically integrate, yet "good" integrators oscillate around the true solution
  - Numerical trajectory 'shadows' the exact orbit, with the proximity to the exact orbit varying with *h*
  - Despite lacking the exact solution, we can gauge the Global Error numerically via conserved quantities
  - Momentum (linear & angular) can be conserved too
  - Linear drift in the conserved quantity is a sign that the equations of motion are not being integrated correctly
- Even with high precision integration, trajectories are extremely sensitive to initial conditions (i.e. chaotic)

Integrator (3): Velocity Verlet

- Resembles Euler method (but with two-step update)
  - Local Error: $O(h^4)$ in position and $O(h^3)$ in velocity

Reduces to the Euler method if **a**(t+h)=**a**(t)

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h\mathbf{v}(t) + \frac{h^2}{2}\mathbf{a}(t)$$

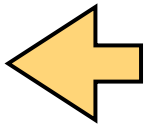
$$\mathbf{v}(t+h) = \mathbf{v}(t) + \frac{h}{2}[\mathbf{a}(t) + \mathbf{a}(t+h)]$$

- Identical Trajectory to Verlet Method
- Uses present state only: [x(t),v(t)] → [x(t+h),v(t+h)]
  - Self-starting
  - Easy to change the time-step

half-step velocities

- Velocity Verlet is often represented in half-steps:

$$\begin{aligned}
\mathbf{v}(t+\frac{1}{2}h) &= \mathbf{v}(t)+\frac{h}{2}\mathbf{a}(t) \\
\mathbf{x}(t+h) &= \mathbf{x}(t)+h\mathbf{v}(t+\frac{1}{2}h) \\
\mathbf{v}(t+h) &= \mathbf{v}(t+\frac{1}{2}h)+\frac{h}{2}\mathbf{a}(t+h)
\end{aligned}$$



Thermostat is applied at this half-step

- In Leap-Frog Verlet the coordinates are defined at full timesteps (t, t+h, t+2h…), while the velocities are defined at half-steps (t-h/2, t+h/2, t+3h/2...).

Equations of motion

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\Delta t^2}{2m}\mathbf{f}(t) + \frac{\Delta t^3}{3!}\frac{d^3\mathbf{r}(t)}{dt^3} + \mathrm{O}\left(\Delta t^4\right)$$

$$\mathbf{r}(t-\Delta t) = \mathbf{r}(t) - \mathbf{v}(t)\Delta t + \frac{\Delta t^2}{2m}\mathbf{f}(t) - \frac{\Delta t^3}{3!}\frac{d^3\mathbf{r}(t)}{dt^3} + \mathrm{O}\left(\Delta t^4\right)$$

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

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

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

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

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

Liouville operator

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

Solution

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

$$i\mathcal{L} \equiv i\mathcal{L}_r + i\mathcal{L}_p = \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$$

$$f(t) = \exp(i\mathcal{L}_r t) f(0)$$

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

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

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

Shift of coordinates

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

$$f(t) = \exp(i\mathcal{L}_p t) f(0)$$

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

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

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

Shift of momenta

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

$$i\mathcal{L}_r \Rightarrow \mathbf{r}(0) \rightarrow \mathbf{r}(0) + \dot{\mathbf{r}}(\mathbf{0})t$$
$$i\mathcal{L}_p \Rightarrow \mathbf{p}(0) \rightarrow \mathbf{p}(0) + \dot{\mathbf{p}}(\mathbf{0})t$$

$$\begin{aligned} f\left(\mathbf{p}^N(t), (\mathbf{r}(t)\right) &= e^{(i\mathcal{L}t)} f\left(\mathbf{p}^N(0), (\mathbf{r}^N(0)\right) \\ &= e^{(i\mathcal{L}_r t + i\mathcal{L}_p t)} f\left(\mathbf{p}^N(0), (\mathbf{r}^N(0)\right) \\ &\neq e^{(i\mathcal{L}_r t)} e^{(i\mathcal{L}_p t)} f\left(\mathbf{p}^N(0), (\mathbf{r}^N(0)\right) \end{aligned}$$

$$= \left(e^{(i\mathcal{L}_p \Delta t/2)} e^{(i\mathcal{L}_r \Delta t)} e^{(i\mathcal{L}_p \Delta t/2)}\right)^P f\left(\mathbf{p}^N(0), (\mathbf{r}^N(0)\right)$$

$$i\mathcal{L}_r \Rightarrow \mathbf{r}(0) \rightarrow \mathbf{r}(0) + \dot{\mathbf{r}}(\mathbf{0})t$$
$$i\mathcal{L}_p \Rightarrow \mathbf{p}(0) \rightarrow \mathbf{p}(0) + \dot{\mathbf{p}}(\mathbf{0})t$$

$$f\left(\mathbf{p}\right.$$

$$= \left(e^{(i}\right.$$

We have *noncommuting* operators!

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

Trotter identity

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

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

$$\frac{A}{P} = \frac{i\mathcal{L}_p t}{P} \qquad \frac{B}{P} = \frac{i\mathcal{L}_r t}{P} \qquad \Delta t = \frac{t}{P}$$

$$i\mathcal{L}_r \Delta t \Rightarrow \mathbf{r} \rightarrow \mathbf{r} + \dot{\mathbf{r}}\Delta t$$
$$i\mathcal{L}_p \Delta t \Rightarrow \mathbf{p} \rightarrow \mathbf{p} + \dot{\mathbf{p}}\Delta t$$

$$\left(e^{(i\mathcal{L}_p \Delta t/2)} e^{(i\mathcal{L}_r \Delta t)} e^{(i\mathcal{L}_p \Delta t/2)} \right)^P$$

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

$$e^{(i\mathcal{L}_r \Delta t)} f' = f\left(\left[\mathbf{p}(0) + \frac{\Delta t}{2}\dot{\mathbf{p}}(0) \right]^N, \left[\mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\frac{\Delta t}{2}) \right]^N \right)$$

$$e^{(i\mathcal{L}_p \Delta t/2)} f'' = f\left(\left[\mathbf{p}(0) + \frac{\Delta t}{2}\dot{\mathbf{p}}(0) + \frac{\Delta t}{2}\dot{\mathbf{p}}(\Delta t) \right]^N, \left[\mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\frac{\Delta t}{2}) \right]^N \right)$$

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

Velocity Verlet:

$$e^{(i\mathcal{L}_p \Delta t/2)} e^{(i\mathcal{L}_r \Delta t)} e^{(i\mathcal{L}_p \Delta t/2)}$$

```
Call force(fx)

Do while (t<tmax)
```

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

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

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

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

$$e^{(i\mathcal{L}_p \Delta t/2)} : \mathbf{v}\left(t + \Delta t\right) \rightarrow \mathbf{v}(t + \Delta t/2) + \frac{\Delta t}{2m}\mathbf{f}(\mathbf{t} + \boldsymbol{\Delta}\mathbf{t})$$

```
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 $\mathbf{r}$ or $\mathbf{p}$ of which the *Jacobian* is one: *Area preserving*

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

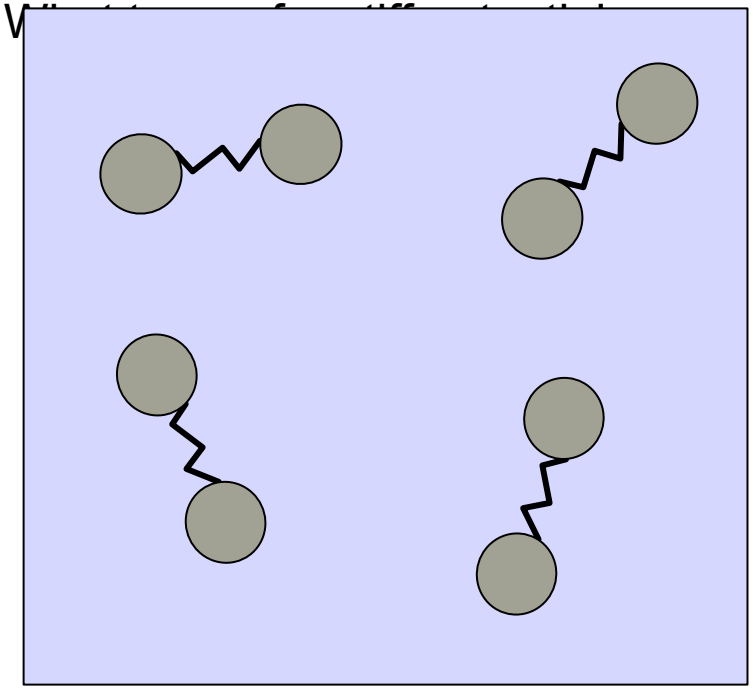
$$\mathbf{p}(t+\Delta t/2)=\mathbf{p}(t+\Delta t/2)$$
$$\mathbf{r}(t+\Delta t)=\mathbf{r}(t)+\frac{\Delta t}{m}\mathbf{p}(t+\Delta t/2)$$
$$J_2=\det\begin{vmatrix}1 & 0\\ \Delta t/m & 1\end{vmatrix}=1$$

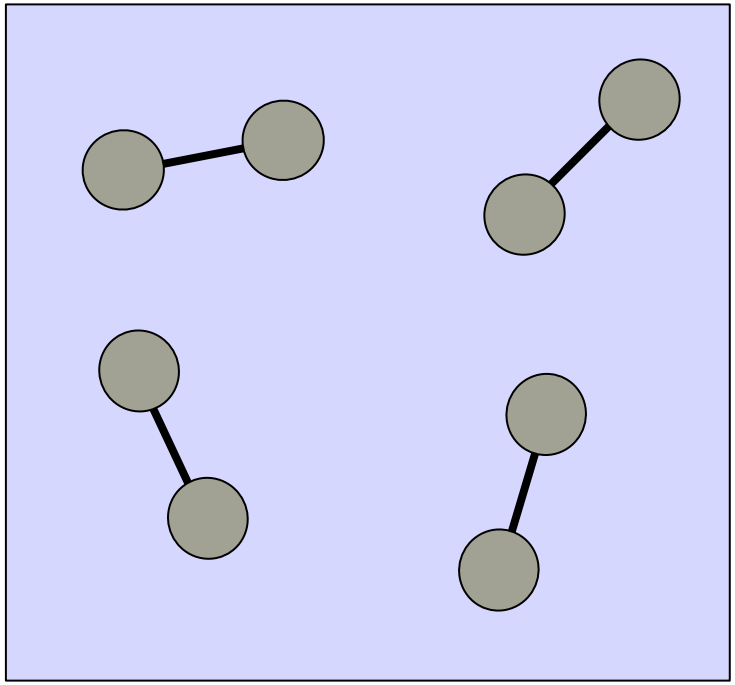
$$\mathbf{p}(t+\Delta t)=\mathbf{p}(t+\Delta t/2)+\frac{\Delta t}{2}\mathbf{F}(\mathbf{r}(t))$$
$$\mathbf{r}(t)=\mathbf{r}(t)$$
$$J_3=\det\begin{vmatrix}1 & \frac{\Delta t}{2}\frac{\partial\mathbf{F}(\mathbf{r})}{\partial\mathbf{r}}\\ 0 & 1\end{vmatrix}=1$$

Other Trotter decompositions are possible!

Multiple time steps

- W





– Very small time step
– Fixed bond-length: constraints (SHAKE/RATTLE/ROLL)
– reversible reference system propagation algorithm (r-RESPA)

Multiple time steps

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

$$i\mathcal{L} \equiv i\mathcal{L}_r + i\mathcal{L}_p = \mathbf{v}\frac{\partial}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m}\frac{\partial}{\partial \mathbf{v}}$$

$$i\mathcal{L} \equiv i\mathcal{L}_{\text{short}} + i\mathcal{L}_{\text{long}}$$

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

Trotter expansion:

$$e^{i(\mathcal{L}_{\text{long}}+\mathcal{L}_{\text{short}}+\mathcal{L}_r)\Delta t} \approx e^{i\mathcal{L}_{\text{long}}\Delta t/2} e^{i(\mathcal{L}_{\text{short}}+\mathcal{L}_r)\Delta t} e^{i\mathcal{L}_{\text{long}}\Delta t/2}$$

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

$$\approx e^{i\mathcal{L}_{\text{long}}\Delta t/2}\left[e^{i\mathcal{L}_{\text{short}}\delta t/2} e^{i\mathcal{L}_r \delta t} e^{i\mathcal{L}_{\text{short}}\delta t/2}\right]^n e^{i\mathcal{L}_{\text{long}}\Delta t/2}$$

$$e^{i(\mathcal{L}_{\rm long}+\mathcal{L}_{\rm short}+\mathcal{L}_r)\Delta t} \approx e^{i\mathcal{L}_{\rm long}\Delta t/2}\left[e^{i\mathcal{L}_{\rm short}\delta t/2}e^{i\mathcal{L}_r\delta t}e^{i\mathcal{L}_{\rm short}\delta t/2}\right]^n e^{i\mathcal{L}_{\rm long}\Delta t/2}$$

$$i\mathcal{L}_{\rm long}\Delta t/2 \Rightarrow v \rightarrow v + F_{\rm long}\Delta t/2m$$

$$i\mathcal{L}_{\rm short}\delta t/2 \Rightarrow v \rightarrow v + F_{\rm short}\delta t/2m$$

$$i\mathcal{L}_r\delta t \Rightarrow r \rightarrow r + v\delta t$$

First

$$e^{(i\mathcal{L}_{\rm long}\Delta t/2)} f\left[r(0), v(0)\right] = f\left[r(0), v(0) + F_{\rm long}(0)\Delta t/2m\right]$$

Now n times:

$$\left[e^{i\mathcal{L}_{\rm short}\delta t/2}e^{i\mathcal{L}_r\delta t}e^{i\mathcal{L}_{\rm short}\delta t/2}\right]^n f\left[r(0), v(0) + F_{\rm long}(0)\Delta t/2m\right]$$

$$e^{(i\mathcal{L}_{long}\Delta t/2)} : \mathbf{v}\left(t+\frac{\Delta t}{2}\right) \rightarrow \mathbf{v}(t) + \frac{\Delta t}{2m}\mathbf{f}_{\rm long}(t)$$

```
Call force(fx_long,f_short)

vx=vx+delt*fx_long/2

Do ddt=1,n
```

$$e^{(i\mathcal{L}_{short}\delta t/2)} : \mathbf{v}\left(t+\frac{\Delta t}{2}+\frac{\delta t}{2}\right) \rightarrow \mathbf{v}(t+\frac{\Delta t}{2}) + \frac{\delta t}{2m}\mathbf{f}_{\rm short}(t)$$

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

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

```
    x=x+ddelt*vx
    Call force_short(fx_short)
```

$$e^{(i\mathcal{L}_{short}\delta t/2)} : \mathbf{v}\left(t+\frac{\Delta t}{2}+\delta t\right) \rightarrow \mathbf{v}\left(t+\frac{\Delta t}{2}+\frac{\delta t}{2}\right) + \frac{\delta t}{2m}\mathbf{f}_{\rm short}(t+\delta t)$$

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

enddo
```

$$e^{(i\mathcal{L}_{long}\Delta t/2)} : \mathbf{v}\left(t+\Delta t\right) \rightarrow \mathbf{v}(t+\frac{\Delta t}{2}) + \frac{\Delta t}{2m}\mathbf{f}_{\rm long}(t+\Delta t)$$

Algorithm 29 (Multiple Time Step)

```
 subroutine                                  Multiple time step, f_long is
+    multi(f_long,f_short)                   the long-range part and f_short
                                             the short-range part of the force
 vx=vx+0.5*delt*f_long                       velocity Verlet with time step Δt
 do it=1,n                                   loop for the small time step
    vx=vx+0.5*(delt/n)*f_short               velocity Verlet with timestep Δt/n
    x=x+(delt/n)2*vx
    call force_short(f_short)                short-range forces
    vx=vx+0.5*(delt/n)*f_short
 enddo
 call force_all(f_long,f_short)              all forces
 vx=vx+0.5*delt*f_long
 return
 end
```

Summary

- Liouville formulation
- Velocity Verlet Algorithm
- Area preserving, time-reversible, energy conserving,...
- Multi-timestep algorithm