Molecular dynamics simulation

how to get things moving

Marcus Elstner and Tomáš Kubař

2019, May 9

Motivation

A (bio)molecule in aqueous solution at ambient conditions

- structure is varying
- energy is fluctuating
- representation with a single, static structure – meaningless
- an interesting process may be going on ☺

State of the system

- (micro)state of a system:
  positions $\vec{r}_i$ and momenta $\vec{p}_i$ of all the atoms
- configuration space – $3N$-dimensional space of coordinates
- phase space – $6N$-dim. space of coords and momenta $\{\vec{r}_i, \vec{p}_i\}$
- trajectory in phase space – sequence of points $\{\vec{r}_i(t), \vec{p}_i(t)\}$
  passed by the system in course of time

State of the system – example I

1D harmonic oscillator:

time course of coordinate and of velocity

$$\begin{aligned} r(t) &= a \cdot \cos[\omega t] \\ v(t) &= -a\omega \cdot \sin[\omega t] \end{aligned}$$

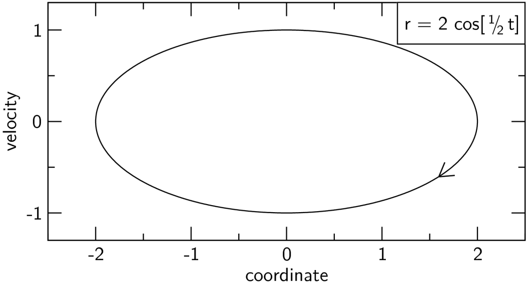
plot of velocity vs. coordinate – in 2D phase space:

elliptic trajectory

$$\left(\frac{x(t)}{a}\right)^2 + \left(\frac{v(t)}{a \cdot \omega}\right)^2 = 1$$

State of the system – example I

1D harmonic oscillator:

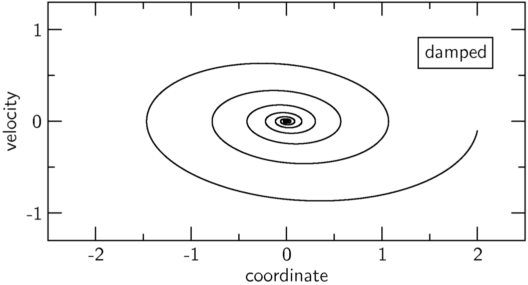


$$E_{\mathrm{tot}} = E_{\mathrm{kin}} + E_{\mathrm{pot}} = \frac{1}{2} m \omega^2 a^2$$

conservative system – total energy remains conserved (constant)

State of the system – example I

1D harmonic oscillator:



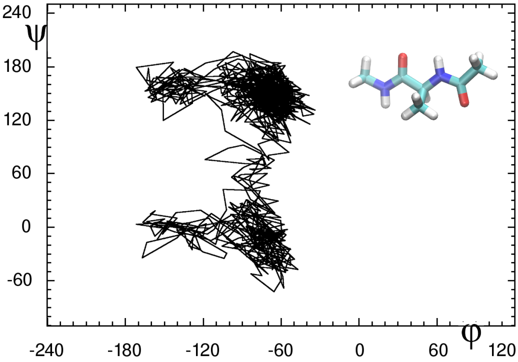
with friction or other damping

– the total energy of the system is decreasing

State of the system – example II

alanine dipeptide in aqueous solution:

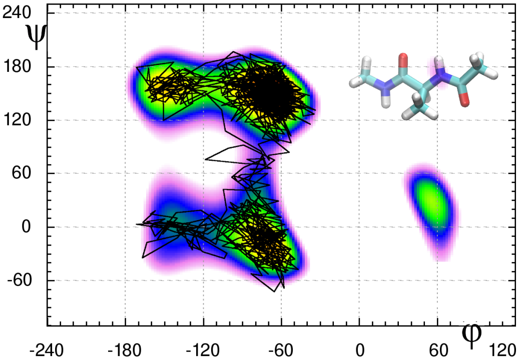
config. space of dihedral angles φ, ψ (Ramachandran plot)



State of the system – example II

alanine dipeptide in aqueous solution:

config. space of dihedral angles φ, ψ (Ramachandran plot)



Thermodynamic properties

MD simulation generates a trajectory in phase space
→ snapshots $\{\vec{r}_i(t_k), \vec{p}_i(t_k)\}$ in time instants $t_k \; (k = 1, \dots, M)$

Generally – obtain the average value of a property of interest
over all observed structures

For energy: evaluate E_k in every snapshot
and calculate the average:

$$\langle E \rangle_t = \frac{1}{M} \sum_{k=1}^{M} E_k$$

Thermodynamic properties

Still, there are issues:

- Do we have enough snapshots = all relevant conformations?
  – convergence of the simulation
- How do we consider experimental conditions – temperature?
- Suppose we know the structure of the reactant.
  How do we get the structure of the product?
  or even the whole reaction path?
- Does the average of energy provide useful information?
  What about free energies / entropy?

Thermodynamic properties

Characteristics of (bio)molecular simulations:

- it is easy to derive the total energy – force field
- not so easy to make proper use of the energy function
  to get the thermodynamic properties right
- it is all about thermodynamics
  in possible contrast to quantum chemistry

Thermodynamic properties

time average for energy and other properties of interest:

$$\langle A \rangle_t = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} A(t)\, \mathrm{d}t$$

experimental sample – huge number of molecules,
all relevant conformations of molecule/solvent are present
– thermodynamic ensemble

How many molecules in the ensemble are found in $\{\vec{r}_i, \vec{p}_i\}$?
$\rightarrow$ phase-space density (per volume unit) $\rho(\vec{r}, \vec{p})$
$\rightarrow$ ensemble average can be calculated:

$$\langle A \rangle_e = \frac{\int A \cdot \rho(\vec{r}, \vec{p})\, \mathrm{d}\vec{r}\, \mathrm{d}\vec{p}}{\int \rho(\vec{r}, \vec{p})\, \mathrm{d}\vec{r}\, \mathrm{d}\vec{p}}$$

Thermodynamic properties

experiment – ensemble average is always measured
simulation – a single molecule – time average available

simulation – system is considered ergodic
– passes through all points of phase space
constituting the real ensemble,
provided the simulation is long enough
– implies:

$$\langle A \rangle_t = \langle A \rangle_e$$

proper sampling × danger of undersampling / lack of convergence

Déjà vu – energy

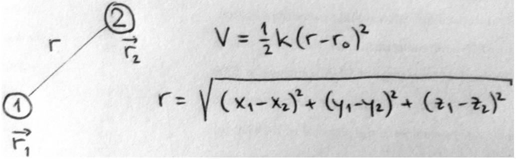
$$E(R^N) =$$

$$\begin{aligned} &= \frac{1}{2}\sum_i k_i(r_i - r_i^0)^2 + \frac{1}{2}\sum_j k_j^\vartheta(\vartheta_j - \vartheta_j^0)^2 + \frac{1}{2}\sum_n V_n \cdot \cos\left[n\omega - \gamma_n\right] \\ &+ \sum_i^N \sum_{j=i+1}^N \left\{4\varepsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6\right) + \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{r_{ij}}\right\} \end{aligned}$$

Déjà vu – forces

$$V = V(r_{ij}) + V(r_{ik}) + V(r_{il}) + \dots$$

$$F_i^x = -\frac{\partial V}{\partial x_i} = -\frac{\partial V(r_{ij})}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial x_i} - \frac{\partial V(r_{ik})}{\partial r_{ik}}\frac{\partial r_{ik}}{\partial x_i} - \frac{\partial V(r_{il})}{\partial r_{il}}\frac{\partial r_{il}}{\partial x_i} - \dots$$



$$V = \frac{1}{2}k(r-r_0)^2$$

$$r = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$

$$\vec{F}_1 = -k(r_{12}-r_0)\cdot\frac{\vec{r}_{12}}{r_{12}}$$

Equations of motion

total energy – Hamilton function (Hamiltonian):

$$H = T + V = \frac{1}{2}\frac{p^2}{m} + \frac{1}{2}kr^2$$

equations of motion in Hamilton's formalism:

$$\dot{r}_i = \frac{\partial H}{\partial p_i} \qquad \dot{p}_i = -\frac{\partial H}{\partial r_i}$$

leading to ordinary differential eqn (ODE) of 2nd order

$$\begin{aligned} \dot{r} &= \frac{\partial H}{\partial p} = \frac{p}{m} \rightarrow p = m\dot{r} \rightarrow \dot{p} = m \cdot \ddot{r} \\ \dot{p} &= -\frac{\partial H}{\partial r} = -\frac{\partial V}{\partial r} = F \end{aligned}$$

$$m \cdot \ddot{r} = F$$

Equations of motion

example – harmonic oscillator: $H = \frac{1}{2}\frac{p^2}{m} + \frac{1}{2}kr^2$

$$\begin{aligned}\dot{r} &= \frac{\partial H}{\partial p} = \frac{p}{m} \\ \dot{p} &= -\frac{\partial H}{\partial r} = F = -k \cdot r\end{aligned}$$

equation of motion: $m \cdot \ddot{r} = -k \cdot r$

- MD – similar concept: consider x, y, z of atoms instead of r, and take forces from the 'long equation'
- Hamilton / Lagrange formalisms are more general
  - other coordinates than x, y, z of atoms may be used
  - internal coordinates ...

MD: very complex expression for the force
→ no analytical solution → numerical solution is necessary

(Too) simple numerical solution

$$\ddot{r} = f(r, t)$$

common trick – Taylor expansion ($\Delta t = t - t_0$):

$$r(t) = r(t_0) + \dot{r}(t_0) \cdot \Delta t + \frac{1}{2}\ddot{r}(t_0) \cdot \Delta t^2 + \ldots$$

Euler method – 1st-order approximation:

$$r(t) \approx r(t_0) + \dot{r}(t_0) \cdot \Delta t$$

Numerical integration starts at time t_0 – we make a step Δt:

$$\begin{aligned} a(t_0) &= -\frac{F}{m} \\ r(t_0 + \Delta t) &= r(t_0) + v(t_0) \cdot \Delta t \\ v(t_0 + \Delta t) &= v(t_0) + a(t_0) \cdot \Delta t \end{aligned}$$

Verlet – normal form

Euler method – too large numerical error $\mathcal{O}(\Delta t^2)$
more accurate integration is needed

Verlet method:
Taylor expansion up to 2nd order,
derivation from two virtual steps, forwards and backwards:

$$\begin{aligned} r(t+\Delta t) &= r(t) + \dot{r}(t)\cdot\Delta t + \tfrac{1}{2}\ddot{r}(t)\cdot\Delta t^2 \\ r(t-\Delta t) &= r(t) - \dot{r}(t)\cdot\Delta t + \tfrac{1}{2}\ddot{r}(t)\cdot\Delta t^2 \end{aligned}$$

add both equations – eliminate the velocity $\dot{r}$:

$$\begin{aligned} r(t+\Delta t) &= 2\cdot r(t) - r(t-\Delta t) + \ddot{r}(t)\cdot\Delta t^2 \\ \ddot{r}(t) &= a(t) = \frac{F(t)}{m} = -\frac{1}{m}\frac{\partial V}{\partial r}(t) \end{aligned}$$

Verlet – normal form

$$r(t+\Delta t) \;=\; 2 \cdot r(t) - r(t-\Delta t) + \ddot{r}(t) \cdot \Delta t^2$$

$r(t-\Delta t)$? information equivalent to velocity,
so that initial conditions may be converted:

$$r(t_0 - \Delta t) = r(t_0) - v(t_0) \cdot \Delta t$$

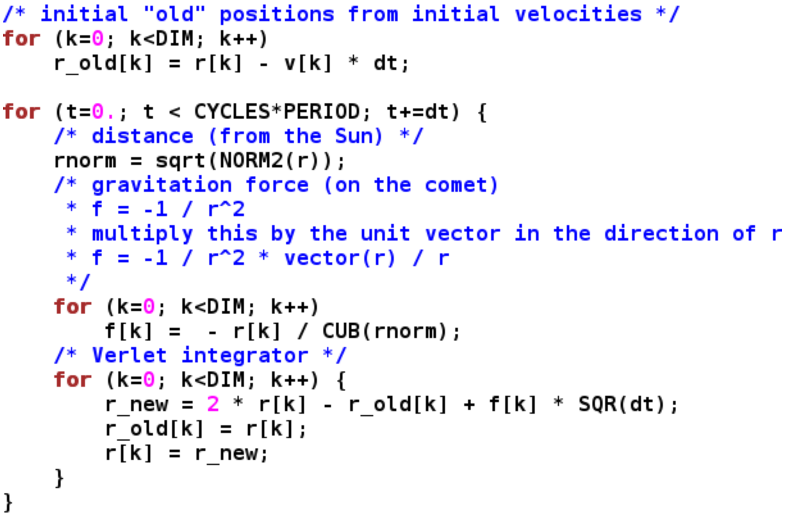
velocities – not in there explicitly, but may be obtained:

$$\dot{r}(t) = v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2 \cdot \Delta t}$$

(Verlet normal form)

Verlet – normal form

program for ‘astronomic’ simulations: $(\vec{F} = -1/r^2 \cdot \vec{r}/r)$



```
/* initial "old" positions from initial velocities */
for (k=0; k<DIM; k++)
    r_old[k] = r[k] - v[k] * dt;

for (t=0.; t < CYCLES*PERIOD; t+=dt) {
    /* distance (from the Sun) */
    rnorm = sqrt(NORM2(r));
    /* gravitation force (on the comet)
     * f = -1 / r^2
     * multiply this by the unit vector in the direction of r
     * f = -1 / r^2 * vector(r) / r
     */
    for (k=0; k<DIM; k++)
        f[k] =  - r[k] / CUB(rnorm);
    /* Verlet integrator */
    for (k=0; k<DIM; k++) {
        r_new = 2 * r[k] - r_old[k] + f[k] * SQR(dt);
        r_old[k] = r[k];
        r[k] = r_new;
    }
}
```

Velocity Verlet

another, equivalent formulation

- positions calculated first using velocities

$$r(t+\Delta t) = r(t) + v(t)\cdot\Delta t + \tfrac{1}{2}a(t)\cdot\Delta t^2$$

- forces (→ accelerations) calculated in new positions,
  and new velocities obtained as

$$v(t+\Delta t) = v(t) + \tfrac{1}{2}\big(a(t) + a(t+\Delta t)\big)\cdot\Delta t$$

- next calculation of positions $r\dots$

MD is started with r_0 and v_0.
Then, in every step, $r(t+\Delta t)$ is calculated first
so that $a(t+\Delta t)$ can be updated, to get $v(t+\Delta t)$

Velocity Verlet

VV – better numerical precision than normal Verlet

numerical problem of normal Verlet:

- adding a small but important term $\ddot{r}(t_0)\Delta t^2$ to
  a large term calculated as difference: $2r(t) - r(t - \Delta t)$
- large relative uncertainty

desirable – use an algorithm that

- is mathematically equivalent, and
- does not involve potentially problematic calculations

Leap-frog

yet another equivalent formulation, similar to VV:

r and v are calculated in an alternating fashion:
$r(t)$, $v(t+\frac{1}{2}\Delta t)$, $r(t+\Delta t)$, $v(t+\frac{3}{2}\Delta t)$, $r(t+2\Delta t)$ ...

- velocities at $t+\frac{1}{2}\Delta t$ are obtained first:

$$v(t+\tfrac{1}{2}\Delta t) = v(t-\tfrac{1}{2}\Delta t) + a(t)\cdot\Delta t$$

- then, positions are updated at $t+\Delta t$:

$$r(t+\Delta t) = r(t) + v(t+\tfrac{1}{2}\Delta t)\cdot\Delta t$$

So, accelerations have to be calculated at t, $t+\Delta t$, $t+2\Delta t$...
from forces, and positions are needed to compute forces
(note: $a(t)$ can only be calculated whenever $r(t)$ are known)

Initial conditions

To start the MD
– the positions r_0 and the velocities v_0 have to be specified
First step – calculations of forces at r_0 to get accelerations a_0
Then – the integrator may provide r (and v) at time $t_0 + \Delta t$

To obtain a trajectory over a time interval T,
we perform M steps
– we have to evaluate the forces on all atoms $M = T/\Delta t$ times

Computational cost of the calculation of forces
determines how many steps we can afford to make

Δt – crucial parameter

Numerical issue:

- we neglect contributions in Δt^3 and higher orders
  $\rightarrow$ error per step in the order of Δt^3 ($\mathcal{O}(\Delta t^3)$)
- keep the step short → make the error small
- disadvantage: we may need too many steps to simulate certain time T
- trade-off: Δt too long → too large error
  dynamics may deviate, momentum may not be conserved. . .

Δt – crucial parameter

Chemical issue:

- fastest motion – hydrogen atoms, period around 10 fs
- rule of thumb – stable integration with $\Delta t \leq$ fastest period (much more relaxed than in 'astronomic' simulations ☺)
- practically, Δt of 1 fs is used (2 fs with special treatment)

→ 1M calculations of forces needes for a trajectory of 1 ns
large systems – multi-ns simulations routinely, μs possible

Δt – astronomic test

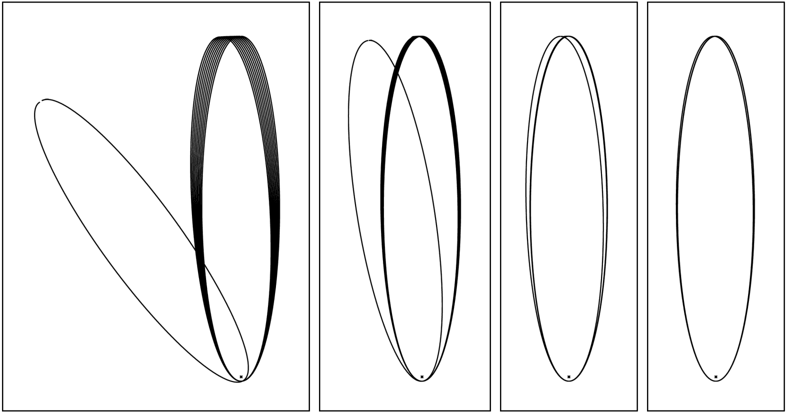
static heavy object (star) + moving light object (comet)
+ gravity ($F = -1/r^2$) → dynamics with Verlet integrator

gravity – inverse-square law much like Coulomb between atoms
exact trajectory – periodic along an ellipse with star in 1 focus

simulation – four different values of the time step:
1×10^{-5}, 2×10^{-5}, 5×10^{-5} and 10×10^{-5} of the orbital period
→ 100k, 50k, 20k and 10k steps per period – mmore than in MD

Δt – astronomic test

first 10 orbits are shown, and then the 100th shown again



credit for the idea: Jiří Kolafa, Prague

Δt – astronomic test

shortest step – reasonable trajectory, small deviation
2nd-longest step – error becomes evident
longest step – large deviation

important – trajectory is precessing (ellipse is 'rotating'),
  but it remains elliptic whatever the time step is
total energy and orbital period remain constant, also
  – consequence of reversibility of Verlet:
  if we reverse the course of time ($\Delta t \to -\Delta t$),
    we will simulate towards initial conditions of the trajectory

generally – energy in Verlet fluctuates (with longer step),
  but it does not drift

Verlet or something better?

Verlet – very approximative yet still routinely used for MD
why? – because it is efficient – why?

- forces on atoms (→ accelerations) calculated only 1× per step
- no higher derivatives of positions are involved

more accurate methods to integrate ODEs are available, and
are used in some applications, if improved accuracy is required

straightforward – involve extra terms from Taylor expansion
– hardly ever done, there are other ways to improve accuracy. . .

Gear integration: predictor–corrector

- provides solution correct to an order of choice
- new positions are calculated (predicted) from Taylor expansion
  using a certain number of previous steps
- then, forces ($\rightarrow$ accelerations)
  are calculated in the predicted positions
- accelerations used to make correction of positions
- additional computational effort, decreased efficiency
- accuracy may be improved significantly, longer step possible
- still, only 1 calculation of forces per step

Gear integration: predictor–corrector

nth-order Gear integrator:
coords of all atoms $\vec{r}$ and their derivatives up to the order of $n-1$:

$$R = \begin{pmatrix} \vec{r} \\ \dot{\vec{r}} \cdot \Delta t \\ \ddot{\vec{r}} \cdot \frac{1}{2}\Delta t^2 \\ \dddot{\vec{r}} \cdot \frac{1}{6}\Delta t^3 \end{pmatrix}$$

for the 4th-order method

initialization: $\vec{r}$ and $\dot{\vec{r}}$ from init. conditions, $\ddot{\vec{r}}$ calculated from forces
– 1 calculation of forces required at start
higher derivatives may be set to zero

Gear: 1: prediction

MD step at time t starts with prediction
of coordinates+derivatives at time $t + \Delta t$:

$$R_\mathrm{p}(t + \Delta t) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot R(t)$$

- the matrix contains binomial coefficients
- the calculation passes a polynomial of order $n - 1$
  through the previous n points of the trajectory
  (at t, $t - \Delta t, \ldots t - (n-1)\Delta t$)
  and generates a point on this polynomial after Δt
- prediction may be good for continuous force functions
- no calculation of force up to this point!

Gear: 2: error

next, we calculate the error of the prediction:
we obtain the force at the predicted position,
and compare it with the force predicted in step 1 → error

$$E = \frac{1}{2}\left(\frac{\vec{f}(\vec{r}_{\mathrm{p}})}{m} - \ddot{\vec{r}}_{\mathrm{p}}\right)\Delta t^2$$

E – vector with as many components as the vector of coordinates (every coordinate with its derivatives has ‘its own’ error)

Gear: 3: correction

Finally, using the error E, we calculate the corrected coordinates and derivatives as

$$R(t+\Delta t) = R_{\mathrm{p}}(t+\Delta t) + E \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

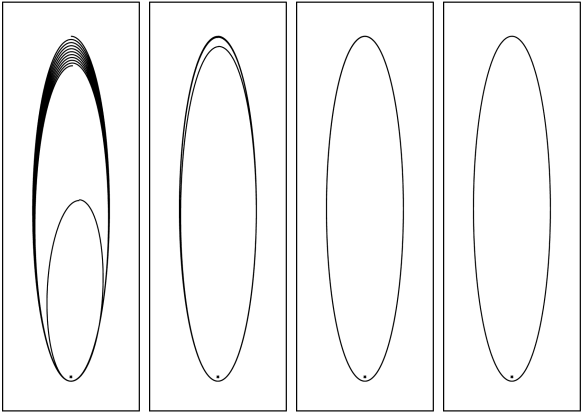
coefficients $a_0,\ a_1 \ldots a_{n-1}$

- estimated to prevent the accumulation of integration errors
- may be looked up in tables
- for 4th-order method for 2nd-order ODE and forces not depending on velocities:

$a_0 = \frac{1}{6}$, $a_1 = \frac{5}{6}$, $a_2 = 1$ and $a_3 = \frac{1}{3}$

Gear – astronomic test

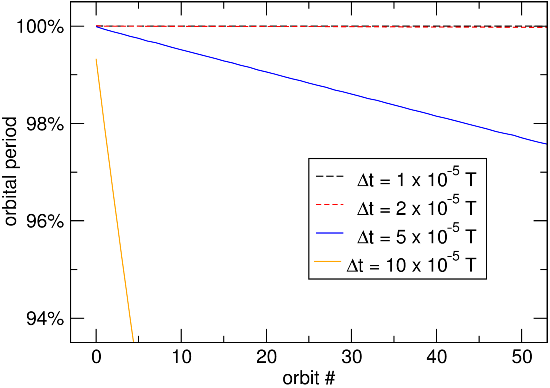
$\Delta t = 10 \times 10^{-5}$, 5×10^{-5}, 2×10^{-5} and 1×10^{-5} of correct period



Gear – astronomic test

- Gear may provide more accurate trajectories than Verlet,
  with the same Δt
- perfect trajectories with the two shortest time steps
  (Verlet showed deviations even with the shortest step)
- incorrect behavior with the second-largest step,
  just wrong with the longest one
- different character of deviation than with Verlet:
  the elliptic trajectory of the comet is getting ‘shorter’,
  rather than precessing
- important: the orbital period is becoming shorter,
  and total energy is decreasing

Gear – astronomic test



general observation:

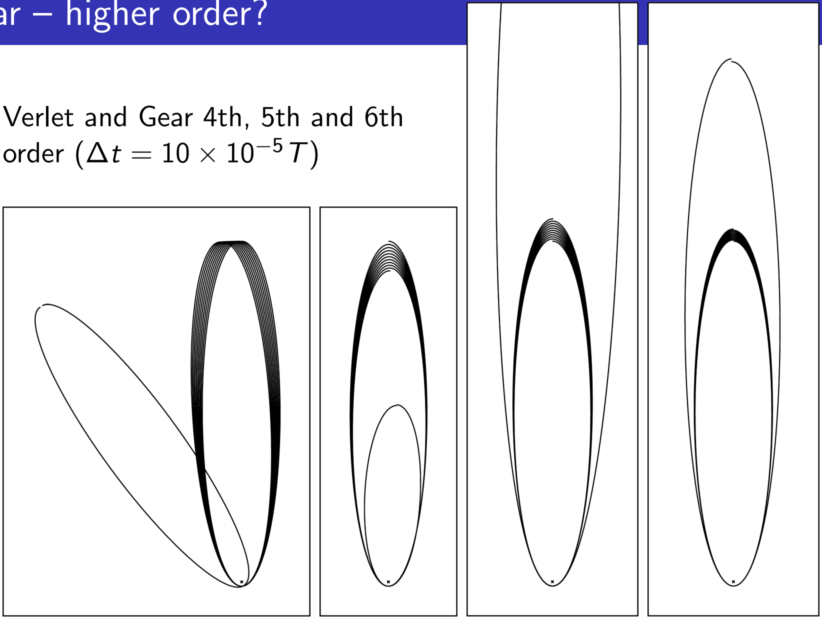
energy will decrease or increase (drift) in the simulation

this may be negligible with longer step / higher-order Gear

Gear: not reversible, does not conserve energy.

Gear – higher order?

Verlet and Gear 4th, 5th and 6th order ($\Delta t = 10 \times 10^{-5} T$)



Gear – higher order?

note: Verlet corresponds to 3th-order Gear formally
higher derivatives in calculation improve the results only slightly
– the drift of energy is slower but still unsatisfactory

general observations:

- when making the step shorter,
  results of higher-order methods will improve faster
- when making the time step longer,
  higher-order methods are more prone to fail completely
  while lower-order methods are more robust
- higher-order integrators are a good choice
  if accurate trajectories are desired
- lower-order or Verlet integration is sufficient for applications
  with ‘weaker’ requirements – typically, MD

Runge–Kutta integration

Runge–Kutta methods – numerical integrators of 1st-order ODEs
classical 4th-order method RK4:

- 4 calculations of the derivative in every step
- points at which the derivative is calculated
  - chosen depending on the previous calculations,
  - the first is done at the start of the integration step

$$\begin{aligned} g_0 &= \dot{r}\left(r(t)\right) \\ g_1 &= \dot{r}\left(r(t) + \tfrac{1}{2} g_0 \Delta t\right) \\ g_2 &= \dot{r}\left(r(t) + \tfrac{1}{2} g_1 \Delta t\right) \\ g_3 &= \dot{r}\left(r(t) + g_2 \Delta t\right) \end{aligned}$$

We calculate the value of the function at time $t + \Delta t$
using a weighted average of the obtained derivative values:

$$r_{n+1} = r_n + \tfrac{1}{6}\left(g_0 + 2g_1 + 2g_2 + g_3\right) \cdot \Delta t$$

the RK4 method

derivative of y is calculated at points m_0, m_1, m_2 and m_3
calculated derivatives $g_0, \ldots g_3$ are shown as arrows

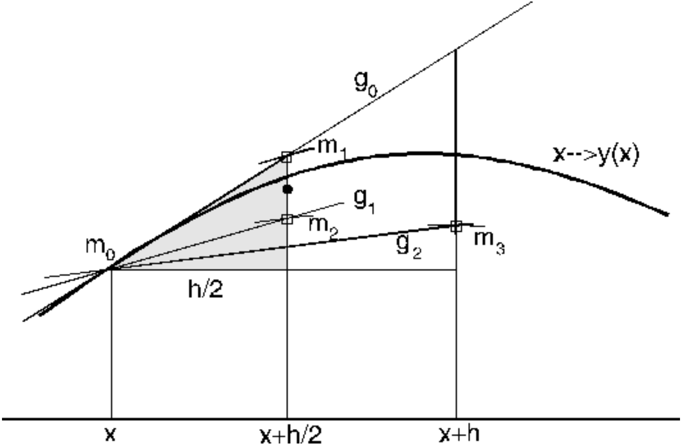


image downloaded from `www.hsg-kl.de`

the RK4 method

- actually predictor–corrector with 4 predictions per step
- 4 calculations of the derivative needed per step
- error per step reduced to $\mathcal{O}(\Delta t^5)$ – 4th-order method
- solves 1st-order ODEs much like the Euler method does
- to solve Newton eqns of motion (2nd-order ODEs)
  - eqns are converted to system of two 1st-order eqns
  - positions and velocities of atoms are propagated:

$$\begin{aligned} \dot{\vec{r}} &= \vec{v} \\ \dot{\vec{v}} &= \frac{\vec{f}}{m} \end{aligned}$$

the RK4 method

Can RK4 be used directly somewhere in computational chemistry?
Yes! Let us propagates a time-dependent Schrödinger equation
– 1st-order ODE for the wave function Ψ of the system:

$$\frac{\partial \Psi}{\partial t} = -\frac{i}{\hbar}\hat{H}\Psi$$

- we express Ψ as linear combination of suitable basis functions:
  $\Psi = \sum_m c_m \varphi_m$
- Hamiltonian is a matrix of elements between basis functions:
  $H_{mn} = \left\langle \varphi_m \middle| \hat{H} \middle| \varphi_n \right\rangle$
- we calculate the derivative with matrix multiplication as $H \cdot \Psi$

Constraints & restraints

Constraint

- condition that the system is required to meet
- example: a bond has length of d exactly: $|\vec{r}_{12}|^2 = d^2$
- the associated mode of motion does not contain any energy

Restraint

- additional energy contribution in the force field
- example: using NMR-estimated distance of atoms j and k,
  $V_{\text{rest}} = \frac{1}{2} k_{\text{rest}} (r_{jk} - r_{\text{NMR}})^2$
- imposes an energy penalty on any deviation,
  but still r_{jk} is allowed to deviate from r_{NMR}
- the affected mode still contributes $\frac{1}{2}kT$ to kinetic energy

Constraint dynamics

Dynamics of large flexible (bio) molecules
   – complex combination of different motions
High-frequency modes of motions – bond stretch / angle bend
   – rather uninteresting, no need for exact description
Lower frequency modes – dihedrals and larger
   – conformational changes, important, must be treated properly

Time step – directed by the highest-frequency modes involved
Idea – keep the bond lengths (or, additionally, angles) fixed,
   and leave other modes of motion untouched
   – introduce constraints

Constraint dynamics

Formally:

Introduce additional (artificial) forces $\vec{G}$ on atoms,
which keep the bond lengths and, optionally, angles fixed:

$$m_i\ddot{\vec{r}}_i = \vec{F}_i + \vec{G}_i$$

Technique:

1. integrate eqns of motion for one step with 'normal' forces $\vec{F}$,
   without considering $\vec{G}$ for now
2. determine the forces $\vec{G}$ required to satisfy constraints
3. correct the new atom positions

The math is somewhat complex. . .

Constraint dynamics – details 1

Example: 3-atomic molecule, bonds 1–2 and 2–3 fixed, angle is free

Eqns of motion:

$$\begin{aligned} m_1 \ddot{\vec{r}}_1 &= \vec{F}_1 + \vec{G}_1 \\ m_2 \ddot{\vec{r}}_2 &= \vec{F}_2 + \vec{G}_2 \\ m_3 \ddot{\vec{r}}_3 &= \vec{F}_3 + \vec{G}_3 \end{aligned}$$

Constraints to be fulfilled:

$$\begin{aligned} \delta_{12} &= r_{12}^2 - d_{12}^2 = 0 \\ \delta_{23} &= r_{23}^2 - d_{23}^2 = 0 \end{aligned}$$

Constraint dynamics – details 2

Lagrangian mechanics provides the constraint forces, generally:

$$\vec{G}_a = \frac{1}{2}\lambda_{12}\,\nabla_a\delta_{12} + \frac{1}{2}\lambda_{23}\,\nabla_a\delta_{23}$$

with so-far undetermined Lagrange multipliers λ

conditions for $\vec{G}_a$:

- must be directed along bonds (to only affect the bond length)
- must obey Newton's 3rd law

$$\begin{aligned}\vec{G}_1 &= \lambda_{12}\vec{r}_{12}\\ \vec{G}_2 &= -\lambda_{12}\vec{r}_{12} + \lambda_{23}\vec{r}_{23}\\ \vec{G}_3 &= -\lambda_{23}\vec{r}_{23}\end{aligned}$$

Constraint dynamics – details 3

Modified eqn for the Verlet integrator:

$$\vec{r}_i(t+\Delta t) = \vec{r}_i{}'(t+\Delta t) + \Delta t^2/m_i \cdot \vec{G}_i$$

Insert the previously obtained constraint forces

$$\begin{aligned}
\vec{r}_1(t+\Delta t) &= \vec{r}_1{}'(t+\Delta t) + \Delta t^2/m_1 \cdot \lambda_{12}\vec{r}_{12} \\
\vec{r}_2(t+\Delta t) &= \vec{r}_2{}'(t+\Delta t) + \Delta t^2/m_2 \cdot (-\lambda_{12}\vec{r}_{12} + \lambda_{23}\vec{r}_{23}) \\
\vec{r}_3(t+\Delta t) &= \vec{r}_3{}'(t+\Delta t) + \Delta t^2/m_3 \cdot (-\lambda_{23}\vec{r}_{23})
\end{aligned}$$

Subtract eqns I–II and II–III to obtain the lengths to be fixed
→ obtain 2 conditions, from which
2 unknowns λ_{12} and λ_{23} can be determined

Constraint dynamics – details 4

$$\begin{aligned}
\vec{r}_{12}(t+\Delta t) &= \vec{r}_{12}{}'(t+\Delta t) + \\
&+ \Delta t^2 (m_1^{-1} + m_2^{-1}) \cdot \lambda_{12} \vec{r}_{12} - \Delta t^2 m_2^{-1} \cdot \lambda_{23} \vec{r}_{23} \\
\vec{r}_{23}(t+\Delta t) &= \vec{r}_{23}{}'(t+\Delta t) - \\
&- \Delta t^2 m_2^{-1} \cdot \lambda_{12} \vec{r}_{12} + \Delta t^2 (m_2^{-1} + m_3^{-1}) \cdot \lambda_{23} \vec{r}_{23}
\end{aligned}$$

- take square modulus of both sides of eqns ($|\vec{r}_{12}|^2, \ldots$)
- apply constraints, $|\vec{r}_{12}|^2 = d_{12}^2, \ldots$
- obtain a set of quadratic eqns for λ_{12} and λ_{23}
- solve, perhaps in a linearized form and iteratively
- obtain the final new coordinates from (previous slide)

$$\vec{r}_1(t+\Delta t) = \vec{r}_1{}'(t+\Delta t) + \Delta t^2 / m_1 \cdot \lambda_{12} \vec{r}_{12}$$

SHAKE

Large (bio)molecule – large number of constraints n_c
Set of eqns – solution requires inversion of an $n_c \times n_c$ matrix
– possibly time-consuming

SHAKE – an alternative algorithm:

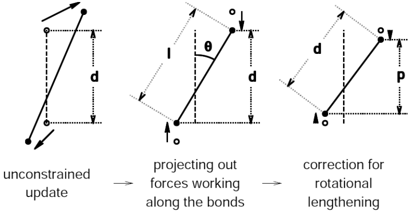
- process the constraints one by one
- satisfying one constraint may violate another
  → iterative procedure necessary
- run until all constraints are met within a preset tolerance
- angle constraints – re-formulate as bond constraints (rigid Δ)

Similar algorithms exist for other integrators,
e.g. RATTLE for velocity Verlet, to treat velocities

LINCS

yet another constraint algorithm

- resets bond lengths after an unconstrained integration step
- non-iterative – always 2 steps:

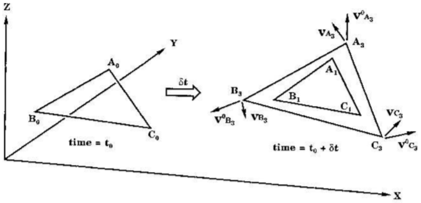


- no expensive matrix operations
- faster and more stable than SHAKE
- available for bond constraints and isolated angle constraints

SETTLE

specialized algorithm for rigid triangles – H_2O

- 3 bond constraints for a molecule with 3 atoms
- analytical, non-iterative solution of SHAKE+RATTLE
- fulfills constraints exactly (→ no tolerance values needed)
- faster than SHAKE → useful for molecules in aqueous solution



Constraint dynamics

Condition

- no coupling between the constrained and unconstrained modes of motion

Usual choices

- bonds with hydrogen
  - Δt may be increased from 1 to 2 fs
- all bonds
- all bonds + all angles
  - may look absurd, but is often a good idea for proteins

Restrained molecular dynamics

Additional contributions in the eqn for total (potential) energy

- ‘penalty’ for deviation from a desired value of a coordinate
- generates additional force
- still, the coordinate *may* deviate from the reference value, and fluctuate

position restraints, angle restraints, distance restraints, orientation restraints and dihedral restraints

Position restraints

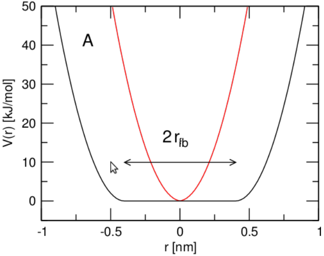
distance of an atom from a fixed reference position $\vec{R}_i$:

$$V_{\text{posres}} = \frac{1}{2} k_{\text{posres}} \left| \vec{r}_i - \vec{R}_i \right|^2$$

- to restrain e.g. the protein during equilibration
  while the solvent is free to move
  - prevent any unwanted drastic rearrangements
- to restrain the surroundings of a region of interest
  whenever there is not enough info on the surroundings
  - the region of interest is simulated without restrains

Flat-bottomed position restraints

- no energy penalty up to a certain distance r_{fb} from the reference position
- restraints the atom to a volume rather than to a point

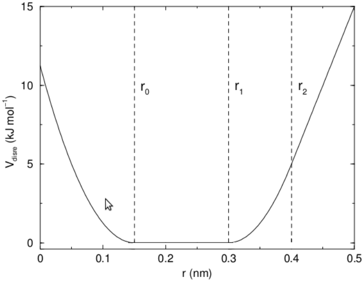


$$r = \left| \vec{r}_i - \vec{R}_i \right|$$

$$V_{\text{fb}} = \begin{cases} \frac{1}{2} k_{\text{fbpr}} (r - r_{\text{fb}})^2 & \text{if } r > r_{\text{fb}} \\ 0 & \text{if } r < r_{\text{fb}} \end{cases}$$

Distance restraints

- penalty according to the distance between two atoms
- often – impose experimental restraints on molecular motion
  e.g. from NMR or diffraction experiments
- MD – tool for structure refinement using NMR data
- optionally time- or ensemble-averaging



$$r = |\vec{r}_i - \vec{r}_j|$$

$$V_{\mathrm{dr}} = \begin{cases} \frac{1}{2}k_{\mathrm{dr}}(r_0 - r)^2 & \text{if } r < r_0 \\ 0 & \text{if } r_0 < r < r_1 \\ \frac{1}{2}k_{\mathrm{dr}}(r - r_1)^2 & \text{if } r_1 < r < r_2 \\ \frac{1}{2}k_{\mathrm{dr}}r(r_2 - r_1) + c & \text{if } r > r_2 \end{cases}$$

Restraints – further ideas

- angle restraints – angle between two bonds
- dihedral restraints
- orientation restraints – angle of two vectors
- time averaging for distance restraints
  – so that fluctuations are not damped
- averaging over multiple pairs of atoms
  – due to the nature of NMR data