

# Molecular Dynamics

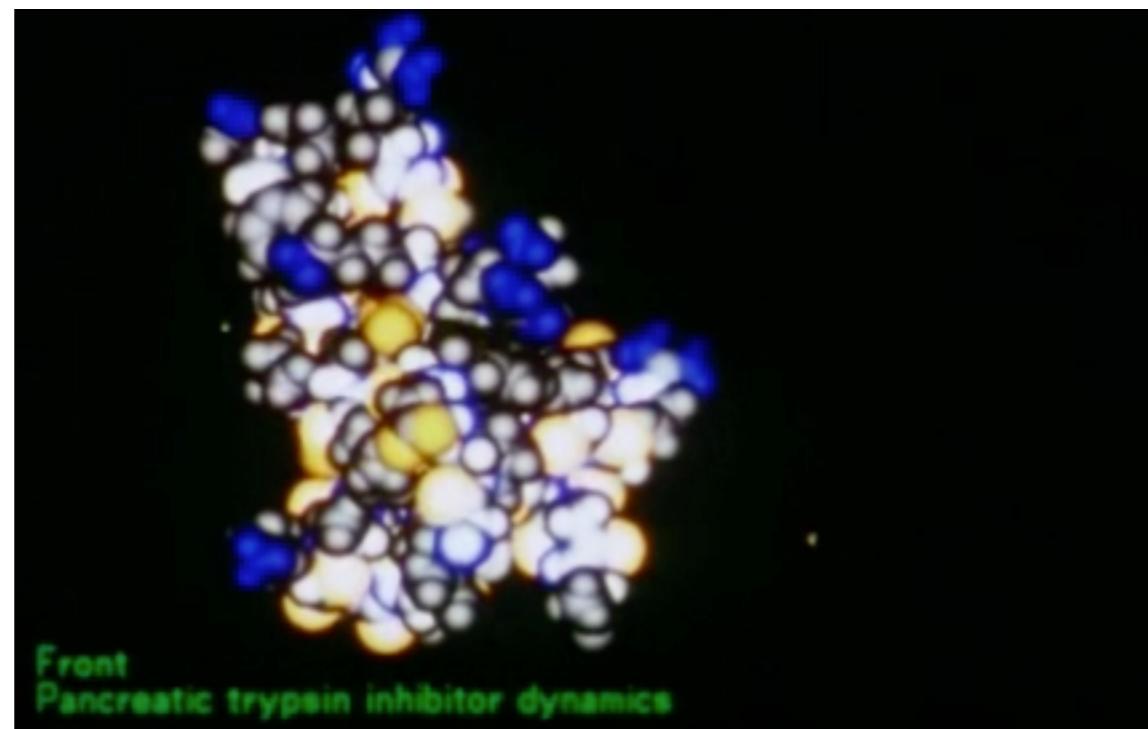
# Day 2

Ben Leimkuhler

**More on Splitting  
Accuracy in MD Simulation  
Stability Threshold  
Multiple Timestepping  
Constraints**

Peking 2018

First movie of a solvated protein (BPTI), 1979

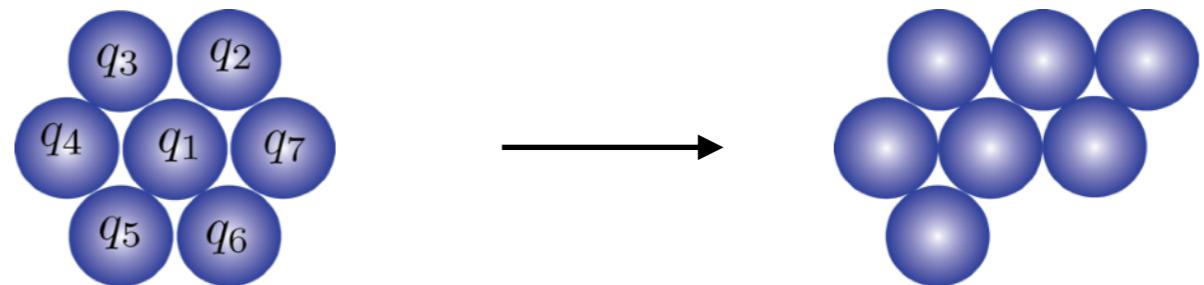


by Nobel Prize winner **Michael Levitt**

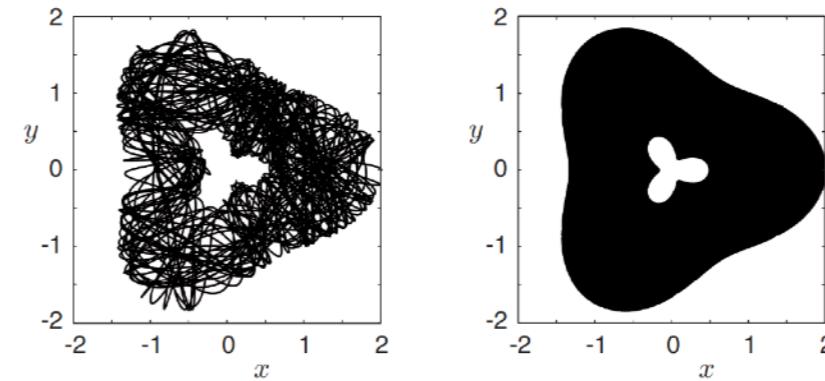
~100ps (100 picoseconds)

# How long do we need to simulate for?

**Rare events:**



**Slow Convergence:**



**May need many many timesteps**

**Hence, we require stable integration methods**

# Why are symplectic methods useful?

Symplectic numerical methods are  
**near-identity** symplectic maps

It turns out that all **near-identity symplectic maps** are  
“close to” Hamiltonian flow maps\*

So we can interpret our numerical solution as the **exact  
solution of a Hamiltonian system** — just a slightly different  
Hamiltonian system than we started with.

\*Meiss: <http://amath.colorado.edu/pub/dynamics/papers/SympMaps.pdf>

# Harmonic Oscillator

Ex:  $H = \frac{p^2 + \omega^2 q^2}{2}, \quad J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

$$H_1(q, p) = p^2/2$$

$$H_2(q, p) = \omega^2 q^2/2$$

$$y' = J \nabla H_1 \rightarrow \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} p \\ 0 \end{bmatrix}$$

$$y' = J \nabla H_2 \rightarrow \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 \\ -\omega^2 q \end{bmatrix}$$

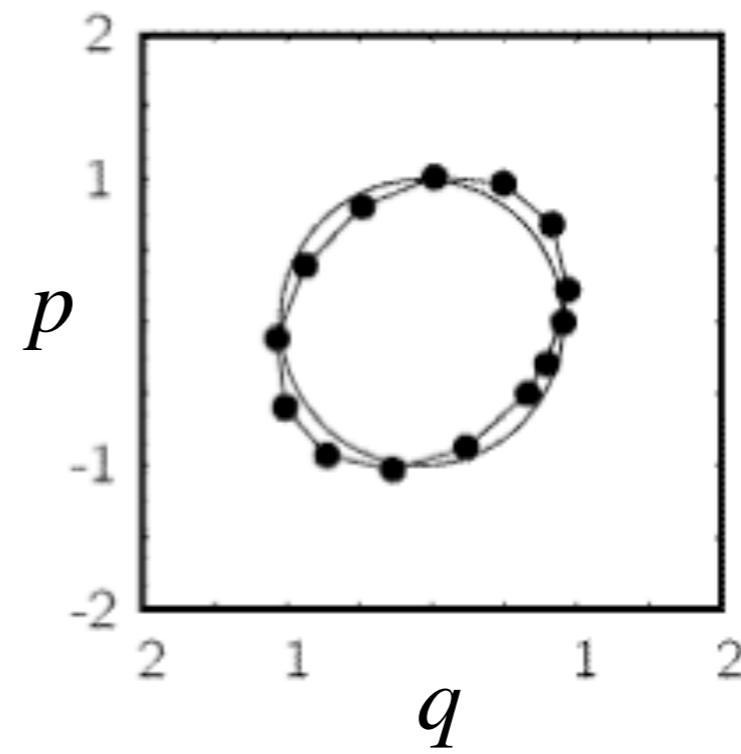
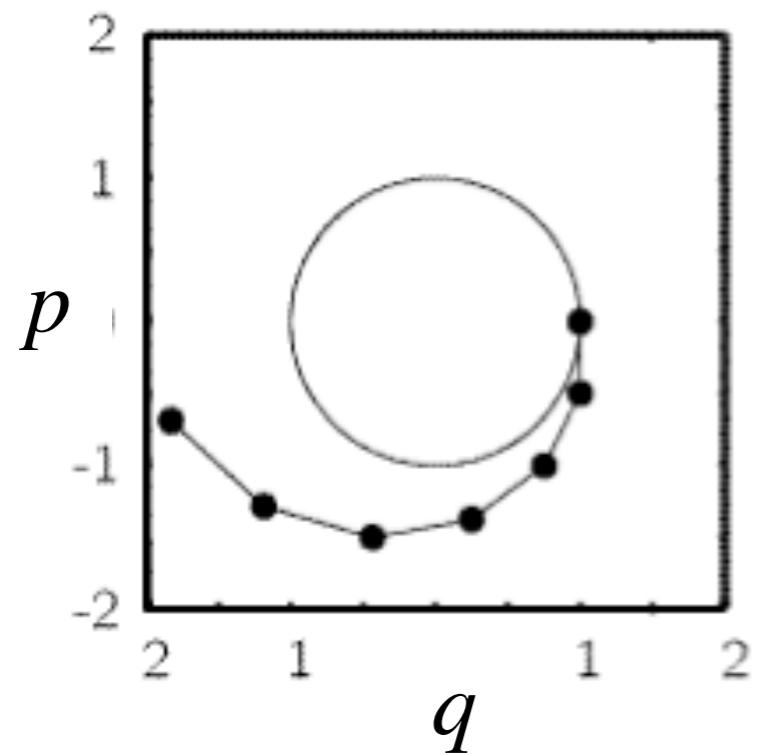
Composition method:

$$q_{n+1} = q_n + h p_{n+1} \quad \text{“Symplectic Euler”}$$

$$p_{n+1} = p_n - h \omega^2 q_n \quad \text{“Euler-B”}$$

# Stability

$$H(q, p) = p^2/2 + \omega^2 q^2/2$$



Euler's method

$$q_{n+1} = q_n + h p_n$$

$$p_{n+1} = p_n - h \omega^2 q_n$$

Symplectic Euler

$$q_{n+1} = q_n + h p_{n+1}$$

$$p_{n+1} = p_n - h \omega^2 q_n$$

# Symplectic Euler

The method can be written in this form:

$$\begin{bmatrix} q_{n+1} \\ p_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - h^2\omega^2 & h \\ -h\omega^2 & 1 \end{bmatrix} \begin{bmatrix} q_n \\ p_n \end{bmatrix} = DRD^{-1} \begin{bmatrix} q_n \\ p_n \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & \omega \end{bmatrix}; \quad R = \begin{bmatrix} 1 - h^2\omega^2 & h\omega \\ -h\omega & 1 \end{bmatrix};$$

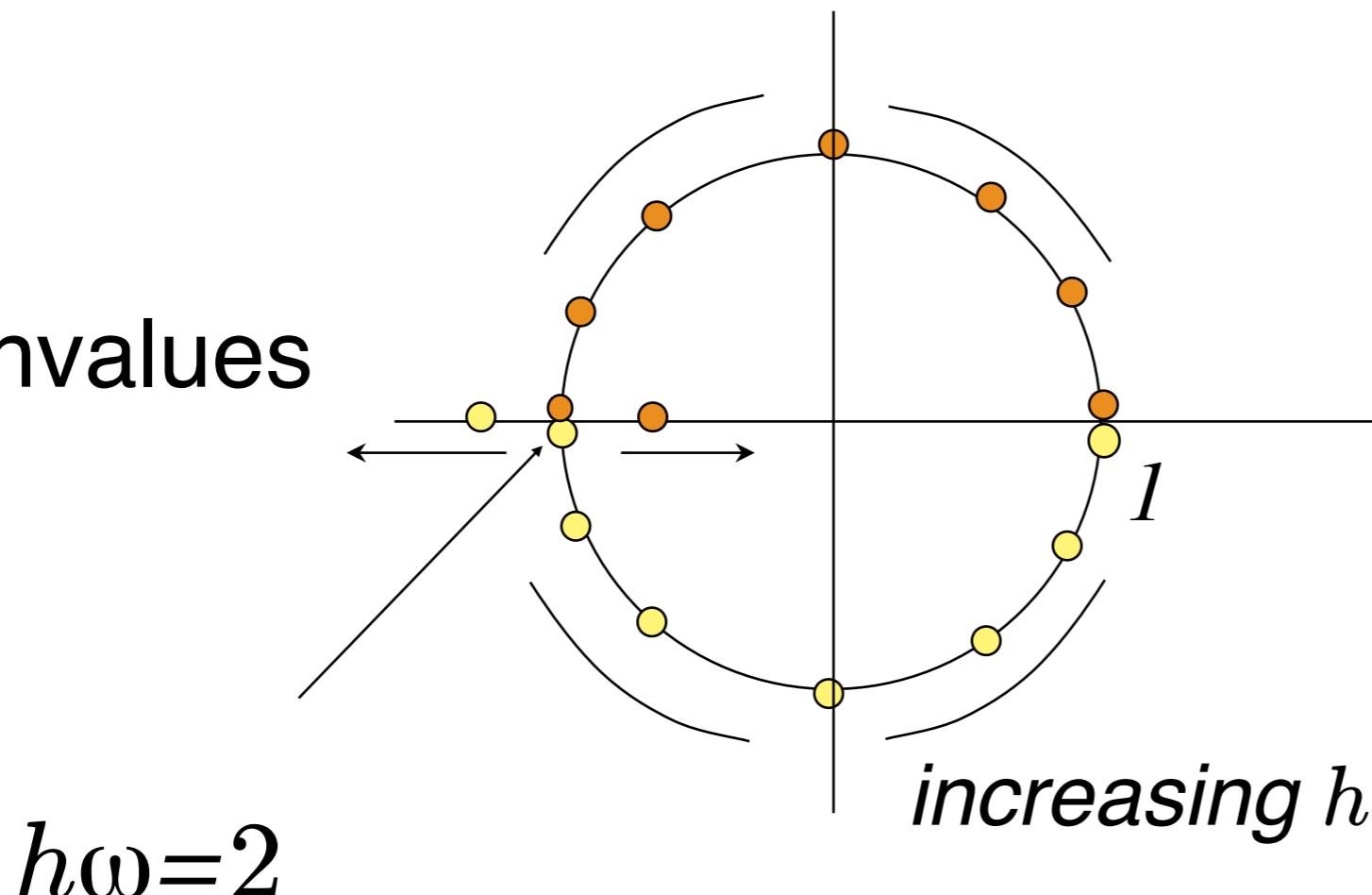
## Eigenvalues:

$$\mu_{1,2} = 1 - h^2\omega^2/2 \pm \sqrt{\Delta}/2$$

$$\Delta = h^4\omega^4 - 4h^2\omega^2,$$

$$h\omega \leq 2$$

Eigenvalues



$$h\omega=2$$

For  $h\omega < 2$ , after a  $O(h)$  rotation of coordinates,  $(q,p) \rightarrow (Q,P)$   
we find that the numerical method is equivalent  
to the solution of a harmonic oscillator with Hamiltonian

$$\tilde{H}(Q, P) = Q^2/2 + \Omega(h)^2 P^2/2$$

$$\Omega(h) = \omega + O(h)$$

# Theorem about Symplectic Integrators

**[Benettin and Giorgilli]** Given a symplectic numerical method  $\Psi_h$ , there is a modified Hamiltonian whose flow map  $\tilde{\Phi}_h$  is exponentially close.

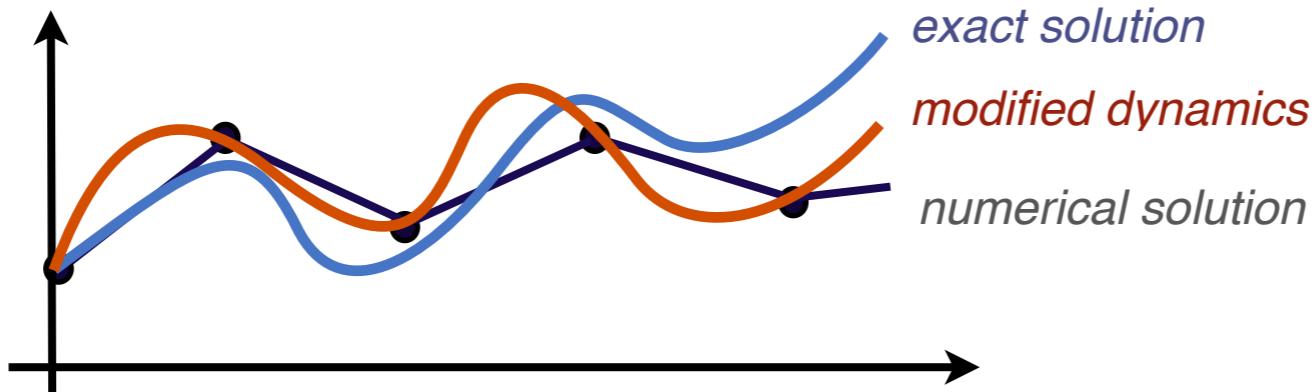
[Benettin & Giorgilli 1994] On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms, J Stat Phys, 74, 1117-1143, 1994.

**Implies existence of an expansion of the Hamiltonian for the numerical method in the general case – not just for linear systems.**

How do we find this expansion? **The Modified Equations!**

# Modified Equations

Warming and Hyett (1974): construct a dynamical system whose flow interpolates a discrete (numerical) trajectory   *exact solution*



# Geometric Integrators

$$\begin{array}{ccc}
 \text{ode} & & \text{modified dynamics} \\
 \dot{x} = f & \approx & \dot{x} = \hat{f}_h \\
 \downarrow & & \uparrow \\
 \text{flow map} & & \text{numerical method} \\
 x(t) = \Phi_t(x(0)) & \approx & x_{n+1} = \hat{\Phi}_h(x_n) \\
 & & IP / V / TR / S
 \end{array}$$

# Lie Derivative

$$\frac{d}{dt} \begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} \nabla_p H \\ -\nabla_q H \end{bmatrix} \quad \dot{z} = f(z)$$

$$g : \mathbb{R}^{2d} \rightarrow \mathbb{R} \quad \mathcal{L}_f g = f \cdot \nabla g$$

$$g(z(t)) = g(z(0)) + t \mathcal{L}_f g|_{z=z(0)} + \frac{t^2}{2} \mathcal{L}_f^2 g|_{z=z(0)} + \dots$$

$$\Phi_t g := g \circ \Phi_t = e^{t\mathcal{L}_f} g$$

Transportation of functions

$$\dot{z} = f_1(z) + f_2(z)$$

**Numerical Method**       $\Psi_h = e^{h\mathcal{L}_{f_1}} e^{h\mathcal{L}_{f_2}}$

**Flow Map**       $\Phi_h = e^{h\mathcal{L}_{f_1+f_2}} = e^{h(\mathcal{L}_{f_1} + \mathcal{L}_{f_2})}$

$$\Phi_h = 1 + h(\mathcal{L}_{f_1} + \mathcal{L}_{f_2}) + \frac{h^2}{2}(\mathcal{L}_{f_1} + \mathcal{L}_{f_2})^2 + \dots$$

$$\Psi_h = (1 + h\mathcal{L}_{f_1} + \frac{h^2}{2}\mathcal{L}_{f_1}^2 + \dots)(1 + h\mathcal{L}_{f_2} + \frac{h^2}{2}\mathcal{L}_{f_2}^2 + \dots)$$

$$\Psi_h = \Phi_h + \frac{h^2}{2}[\mathcal{L}_{f_1}, \mathcal{L}_{f_2}] + O(h^3)$$

local error

# The Modified Equations

$$\Psi_h = e^{h\mathcal{L}_{\tilde{f}}}$$

$$\tilde{f} = f + hf^{(1)} + h^2 f^{(2)} + \dots$$

$$e^{h\mathcal{L}_{\tilde{f}}} = 1 + h\mathcal{L}_{\tilde{f}} + \frac{h^2}{2}\mathcal{L}_{\tilde{f}}^2 + \dots$$

$$= 1 + h(\mathcal{L}_f + h\mathcal{L}_{f^{(1)}} + \dots) + \frac{h^2}{2}\mathcal{L}_f^2 + O(h^3)$$

$$\mathcal{L}_{f^{(1)}} + \frac{1}{2}\mathcal{L}_f^2 = \mathcal{L}_{f_1}\mathcal{L}_{f_2} + \mathcal{L}_{f_1}^2 + \mathcal{L}_{f_2}^2$$

$$\mathcal{L}_{f^{(1)}} = \frac{1}{2}[\mathcal{L}_{f_1}, \mathcal{L}_{f_2}] \longrightarrow f^{(1)} = \frac{1}{2}[f_1, f_2]$$

Lie bracket  
of vector  
fields

similarly, we may construct expressions for  $f^{(2)}, f^{(3)}, \dots$

# Baker-Campbell-Hausdorff Expansion

$$\log(\exp X \exp Y) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \sum_{\substack{r_1+s_1>0 \\ \vdots \\ r_n+s_n>0}} \frac{[X^{r_1} Y^{s_1} X^{r_2} Y^{s_2} \dots X^{r_n} Y^{s_n}]}{\sum_{i=1}^n (r_i + s_i) \cdot \prod_{i=1}^n r_i! s_i!}$$

$$[X^{r_1} Y^{s_1} \dots X^{r_n} Y^{s_n}] = \underbrace{[X, [X, \dots [X,}_{r_1} \underbrace{[Y, [Y, \dots [Y,}_{s_1} \dots \underbrace{[X, [X, \dots [X,}_{r_n} \underbrace{[Y, [Y, \dots Y]] \dots]}_{s_n}]$$

$$\begin{aligned} Z(X, Y) &= \log(\exp X \exp Y) \\ &= X + Y + \frac{1}{2}[X, Y] + \frac{1}{12} ([X, [X, Y]] + [Y, [Y, X]]) \\ &\quad - \frac{1}{24} [Y, [X, [X, Y]]] \\ &\quad - \frac{1}{720} ([Y, [Y, [Y, [Y, X]]]] + [X, [X, [X, [X, Y]]]]) \\ &\quad + \frac{1}{360} ([X, [Y, [Y, [Y, X]]]] + [Y, [X, [X, [X, Y]]]]) \\ &\quad + \frac{1}{120} ([Y, [X, [Y, [X, Y]]]] + [X, [Y, [X, [Y, X]]]]) + \dots \end{aligned}$$

# Hamiltonian Splitting

Splitting methods applied to vector fields correspond to modified differential equations

**When the system involved is Hamiltonian and the splitting respects this structure, then the numerical methods correspond to Hamiltonian vector fields.**

The terms of the general splitting method are expressed as Lie brackets of vector fields

**The terms of the energy function of a Hamiltonian splitting method are expressed as “Poisson brackets” of the Hamiltonian terms involved.**

# Hamiltonian Splitting

$$f = f_1 + f_2 = J \nabla H_1 + J \nabla H_2$$

$$\Psi_h = e^{h\mathcal{L}_{f_1}} e^{h\mathcal{L}_{f_2}}$$

$$\Psi_h = \exp(h\mathcal{L}_{J \nabla \tilde{H}})$$

$$\tilde{H} = H + \frac{h}{2} \{H_1, H_2\} + O(h^2)$$

$$\{H_1, H_2\} = \nabla H_1^T J \nabla H_2$$

*Poisson  
bracket*

# Splitting Tricks and Techniques

- You can have **more than two terms**

$$H = H_1 + H_2 + H_3$$

$$\dot{y} = J\nabla H_1$$

step h

$$\dot{y} = J\nabla H_2$$

step h

$$\dot{y} = J\nabla H_3$$

step h

- You can also choose to **integrate a piece twice** (with half the timestep) (symmetric splitting)

$$H = H_1 + H_2$$

$$\dot{y} = J\nabla H_1$$

step h/2

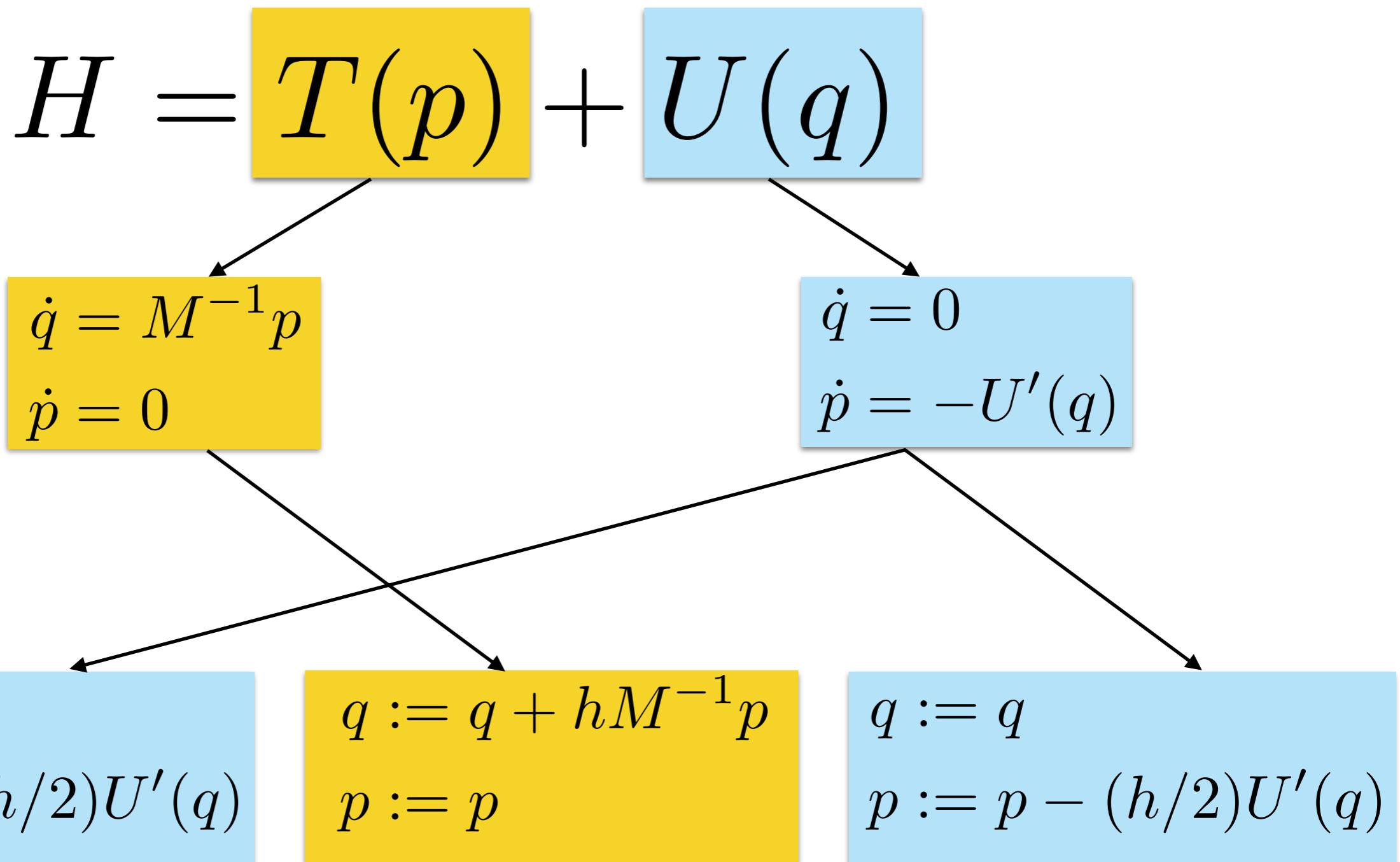
$$\dot{y} = J\nabla H_2$$

step h

$$\dot{y} = J\nabla H_1$$

step h/2

# Leapfrog/Verlet by Hamiltonian splitting



Second order approximation  
***Symplectic, Symmetric***

***One Force Evaluation per Timestep***

# **Symmetric Methods**

# Adjoint Method

Let  $\Psi_h$  be any consistent method

The adjoint method is defined by:

$$\Psi_h^* = \Psi_{-h}^{-1}$$

If the original methods is  $p$ th order accurate,  $p \geq 1$ , then so is its adjoint.

A **symmetric method** is one that is self-adjoint in this sense

# Symmetric Composition

Symmetric methods can be constructed from nonsymmetric ones by composition.

Suppose  $\Psi_h$  is any method of order  $p$ .

Then  $\Psi_h^*$  is also a  $p$ th order method, hence

$$\tilde{\Psi}_h := \Psi_{h/2}^* \Psi_{h/2}$$

is at least a  $p$ th order self-adjoint method.

Symmetric methods are always **even order**.

## ***Example:***

### *Verlet from Symplectic Euler*

$$H(q, p) = p^2/2 + U(q)$$

#### Symplectic Euler

$$q_{n+1} = q_n + hp_{n+1}$$

$$p_{n+1} = p_n - hU'(q_n)$$

$$\text{SE}_h(\cdot)$$

#### Verlet

$$q_{n+1} = q_n + hp_{n+1/2}$$

$$p_{n+1/2} = p_n - (h/2)U'(q_n)$$

$$p_{n+1} = p_{n+1/2} - (h/2)U'(q_{n+1})$$

$$\text{V}_h(\cdot)$$

$$\text{V}_h \equiv \text{SE}_{h/2}^* \circ \text{SE}_{h/2}$$

Symplectic Euler is 1st order

Verlet is symmetric hence (at least) 2nd order

# Symmetric BCH Expansion

$$\exp\left(\frac{t}{2}X\right)\exp(tY)\exp\left(\frac{t}{2}X\right) = \exp(t(X+Y) + t^3\tilde{Z}_{[3]} + \dots)$$

$$\tilde{Z}_{[3]} = \frac{1}{12}[Y,[Y,X]] - \frac{1}{24}[X,[X,Y]].$$

---

**splitting:**  $\dot{z} = f_1(z) + f_2(z)$

$$\begin{aligned} \exp\left(\frac{h}{2}\mathcal{L}_{f_1}\right)\exp(h\mathcal{L}_{f_2})\exp\left(\frac{h}{2}\mathcal{L}_{f_1}\right) \\ = \exp\left(h(\mathcal{L}_f + Wh^2 + O(h^4))\right) \end{aligned}$$

$$W = \frac{1}{12}[\mathcal{L}_{f_2}, [\mathcal{L}_{f_2}, \mathcal{L}_{f_1}]] - \frac{1}{24}[\mathcal{L}_{f_1}, [\mathcal{L}_{f_1}, \mathcal{L}_{f_2}]]$$

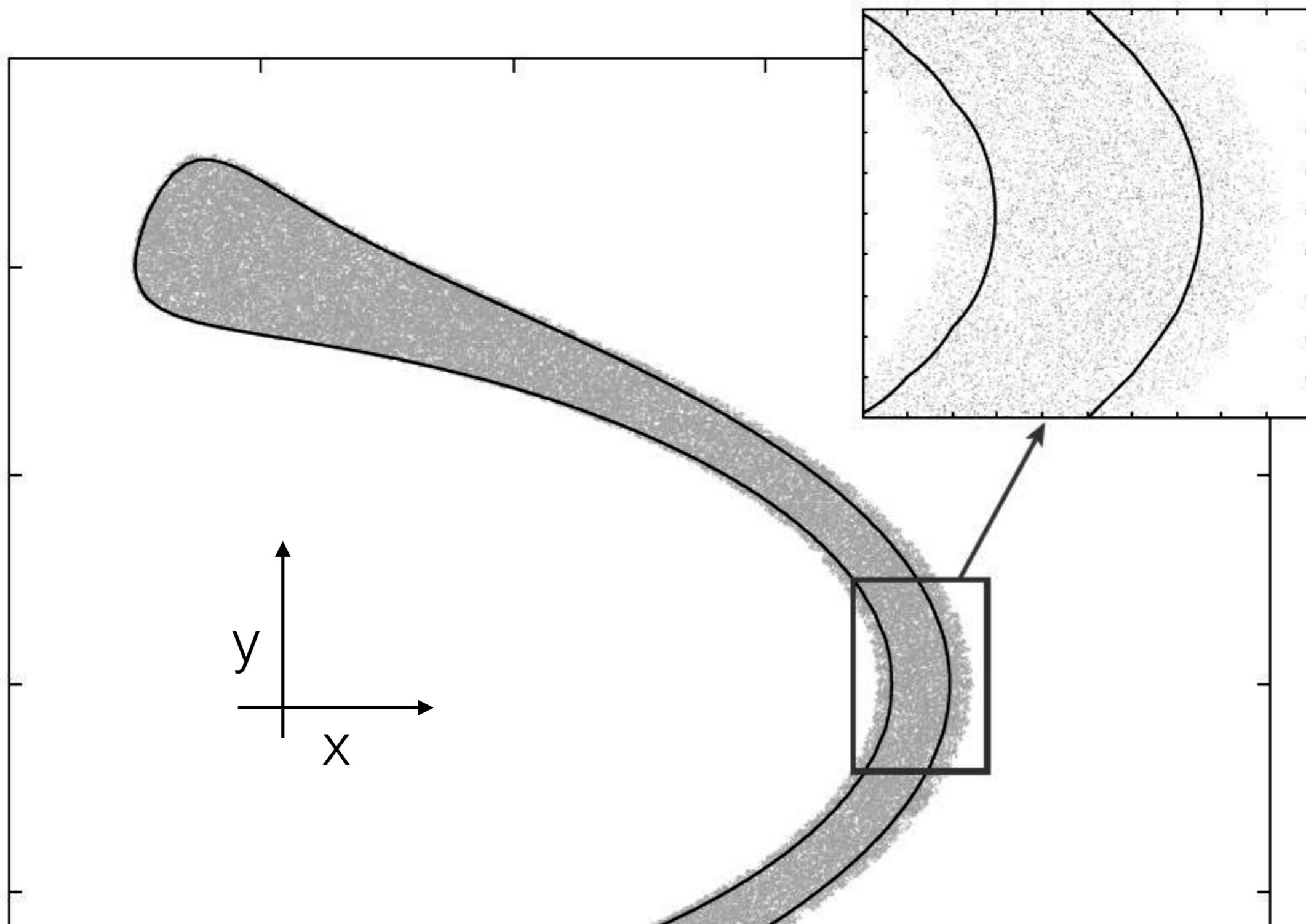
## Verlet/Leapfrog method (written compactly)

$$\begin{aligned} e^{hH} &\approx e^{(h/2)\mathcal{L}_{J\nabla_z U}} e^{h\mathcal{L}_{J\nabla_z T}} e^{(h/2)\mathcal{L}_{J\nabla_z U}} \\ &= e^{h\tilde{H}_h} \end{aligned}$$

$$\begin{aligned} \tilde{H}_h = T + U + \frac{h^2}{12} &\left( \{T, \{T, U\}\} - \frac{1}{2} \{U, \{U, T\}\} \right) \\ + \frac{h^4}{120} &\left( -\frac{1}{6} \{T, \{T, \{T, \{T, U\}\}\}\} + \frac{1}{3} \{U, \{T, \{T, \{T, U\}\}\}\} \right. \\ &\quad \left. - \frac{1}{4} \{U, \{U, \{T, \{T, U\}\}\}\} + \{T, \{T, \{U, \{U, T\}\}\}\} \right) + \mathcal{O}(h^6) \end{aligned}$$

***Using Verlet is just like solving  
a modified Hamiltonian system***

# Trimer – Verlet discretization



**Can we improve the accuracy?**

# **Higher Order Methods**

# Do we need symplectic methods?

Maybe by using a higher accuracy method we don't have to worry about the symplectic property.

Most of the world is very happy with the 4th order Runge-Kutta method, explicit, defined for  $\dot{x} = f(x)$  by:

$$k_1 = f(x_n)$$

$$k_2 = f(x_n + (h/2)k_1)$$

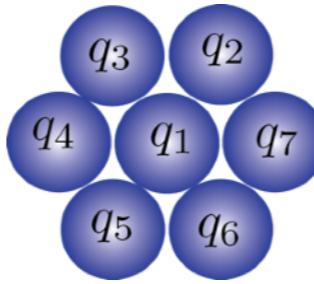
$$k_3 = f(x_n + (h/2)k_2)$$

$$k_4 = f(x_n + hk_3)$$

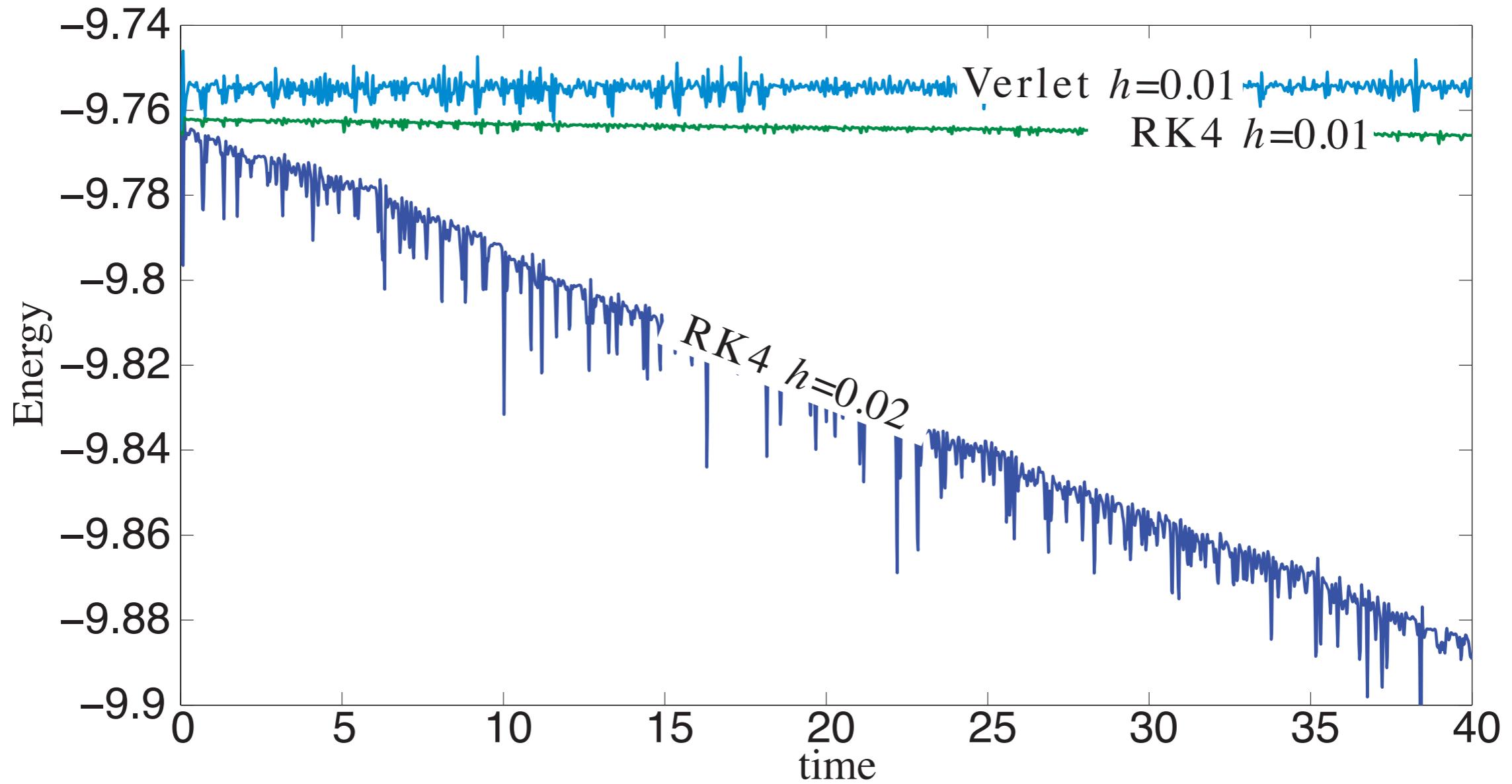
$$x_{n+1} = x_n + (h/6)(k_1 + 2k_2 + 2k_3 + k_4)$$

***Four Force Evaluations  
per Timestep***

LJ-7



*RK4 is four times as expensive at Verlet!*



***What we really need is symplectic, higher order methods!***

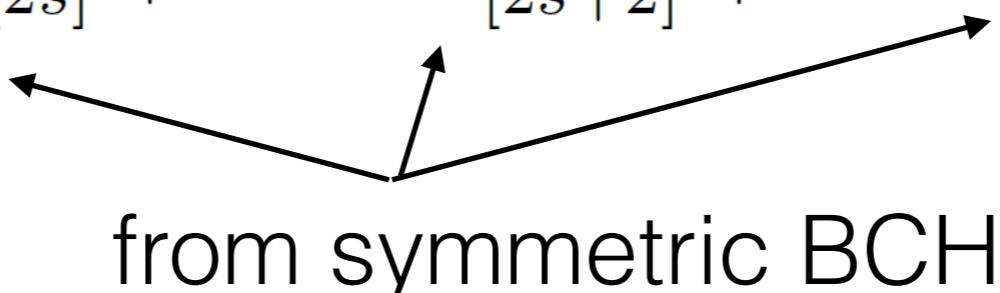
# Yoshida Higher Order Trick

Suppose we have an even order symplectic method

$$\Psi_h = e^{h\hat{\mathcal{L}}_h}$$

where

$$\hat{\mathcal{L}}_h \stackrel{\text{def}}{=} \mathcal{L}_{\tilde{H}_h} = \mathcal{L}_H + h^{2s} \tilde{\mathcal{L}}_{[2s]} + h^{2s+2} \tilde{\mathcal{L}}_{[2s+2]} + h^{2s+4} \tilde{\mathcal{L}}_{[2s+4]} + \mathcal{O}(h^{2s+6})$$



Consider the symplectic method defined by:

$$\exp\left(\tau_0 h \hat{\mathcal{L}}_{\tau_0 h}\right) \exp\left(\tau_1 h \hat{\mathcal{L}}_{\tau_1 h}\right) \exp\left(\tau_0 h \hat{\mathcal{L}}_{\tau_0 h}\right)$$
$$2\tau_0 + \tau_1 = 1$$

Using Symmetric BCH, we find:

$$\exp\left(\tau_0 h \hat{\mathcal{L}}_{\tau_0 h}\right) \exp\left(\tau_1 h \hat{\mathcal{L}}_{\tau_1 h}\right) \exp\left(\tau_0 h \hat{\mathcal{L}}_{\tau_0 h}\right) = e^{h Z_h}$$

$$Z_h = 2\tau_0 \hat{\mathcal{L}}_{\tau_0 h} + \tau_1 \hat{\mathcal{L}}_{\tau_1 h} + \mathcal{O}(h^{2s+2}),$$

$$= (2\tau_0 + \tau_1) \mathcal{L}_H + (2\tau_0^{2s+1} + \tau_1^{2s+1}) h^{2s} \tilde{\mathcal{L}}_{[2s]} + \mathcal{O}(h^{2s+2})$$

= 1  $\uparrow$

**How can we make this term zero?**

$$\tau_0 = \frac{1}{2 - \kappa}, \quad \tau_1 = -\frac{\kappa}{2 - \kappa}, \quad \kappa^{2s+1} = 2.$$

Example ( $s=1$ ):  $2\tau_0^3 + \tau_1^3 = 0$        $\tau_0 = \frac{1}{2 - \sqrt[3]{2}}$ ,       $\tau_1 = -\frac{\sqrt[3]{2}}{2 - \sqrt[3]{2}}$

$$2\tau_0 + \tau_1 = 1$$

# 4th Order Yoshida Method

$$\tau_0 = \frac{1}{2 - \sqrt[3]{2}}, \quad \tau_1 = -\frac{\sqrt[3]{2}}{2 - \sqrt[3]{2}}$$

```
function (Q,P)= verlet(q,p,h)
P = p + 0.5*h*F(q);
Q = q + h*P;
P = P +0.5*h*F(Q);
```

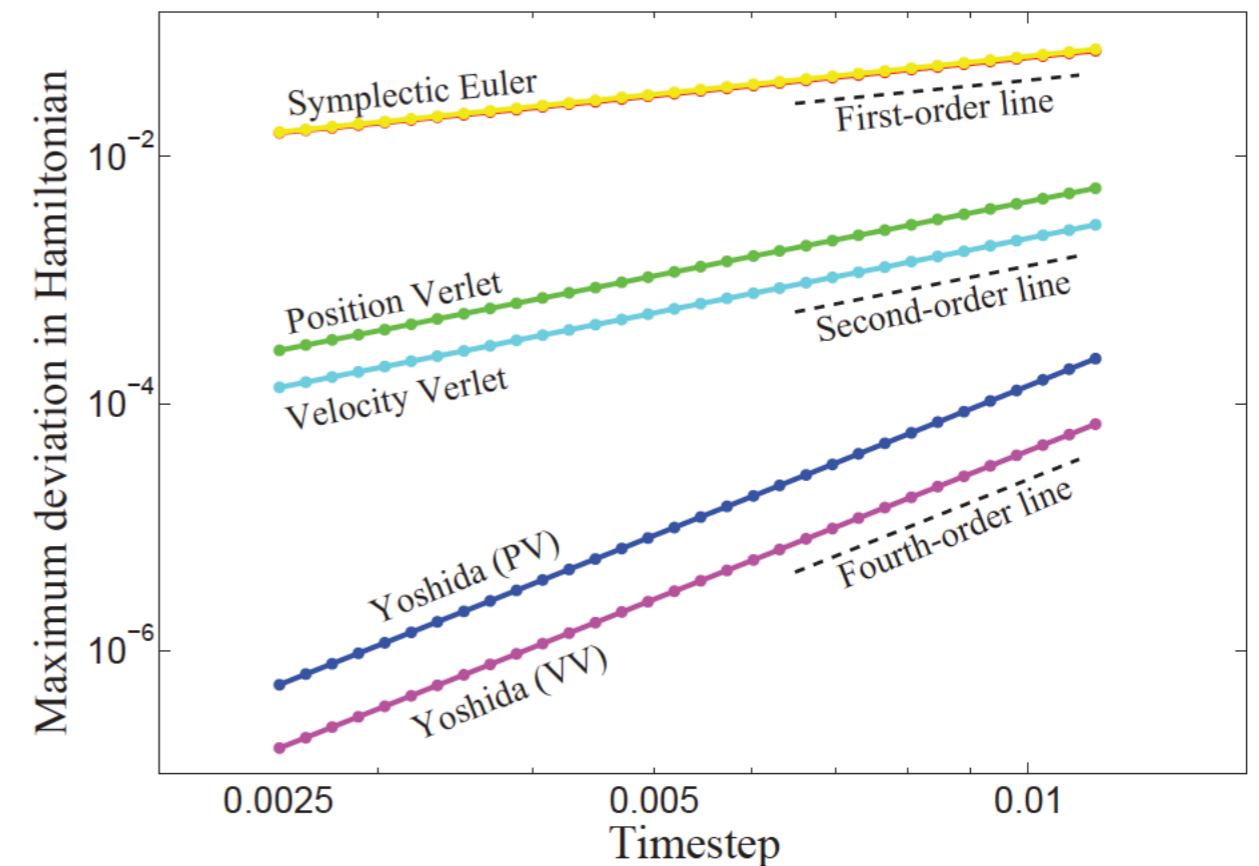
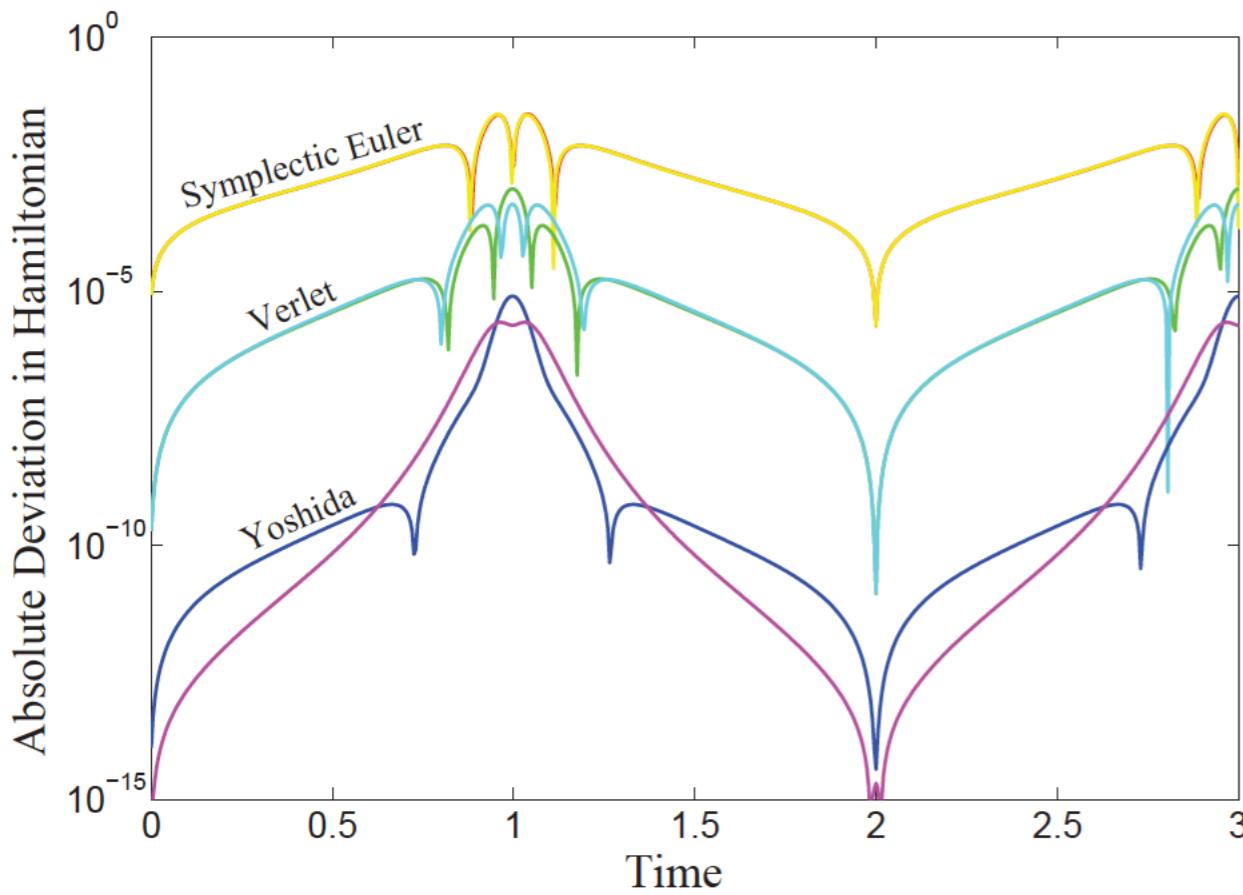
```
function (Q,P)= Yoshida4(q,p,h)
tau0 =1.3512; tau1= -1.7024
(Q,P) = verlet(q,p,h*tau0);
(Q,P) = verlet(Q,P,h*tau1);
(Q,P) = verlet(Q,P,h*tau0);
```

# Comparisons

## Symplectic Euler

## Verlet

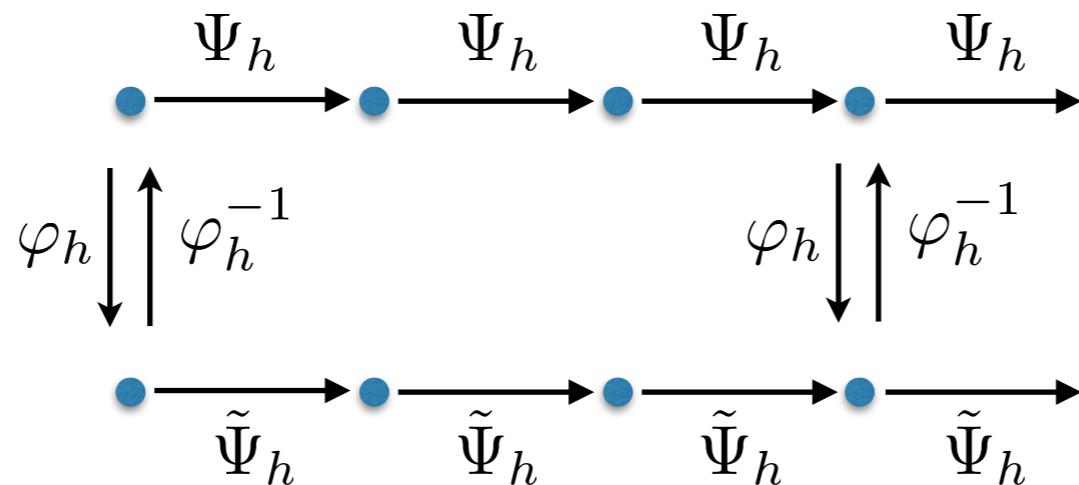
## Yoshida-4 (Forest&Ruth)



# Conjugate Methods

$$\Psi_h = \varphi_h^{-1} \tilde{\Psi}_h \varphi_h$$

$$\begin{aligned}\Psi_h^n &= \Psi_h \circ \Psi_h \circ \dots \circ \Psi_h(z_0) \\ &= (\varphi_h^{-1} \tilde{\Psi}_h \varphi_h)^n \\ &= \varphi_h^{-1} \tilde{\Psi}_h^n \varphi_h\end{aligned}$$



- A sort of method equivalency
- Sometimes can be used to design higher order schemes

# Rowlands' Method

$$\hat{P} = p - (h/2) \nabla \tilde{U}(q),$$

$$Q = q + hM^{-1}\hat{P},$$

$$P = \hat{P} - (h/2) \nabla \tilde{U}(Q)$$

$$\tilde{U}(q) = U(q) - \frac{h^2}{24} \nabla U(q)^T M^{-1} \nabla U(q)$$

- a. Rowlands' Method is a **symplectic method, why?**
- b. Rowlands' Method is **conjugate to** a 4th order method

Lopez-Marcos, Sanz-Serna and Skeel, Explicit Symplectic Integrators  
Using Hessian-Vector Products, SIAM J. Sci Comput., 1997

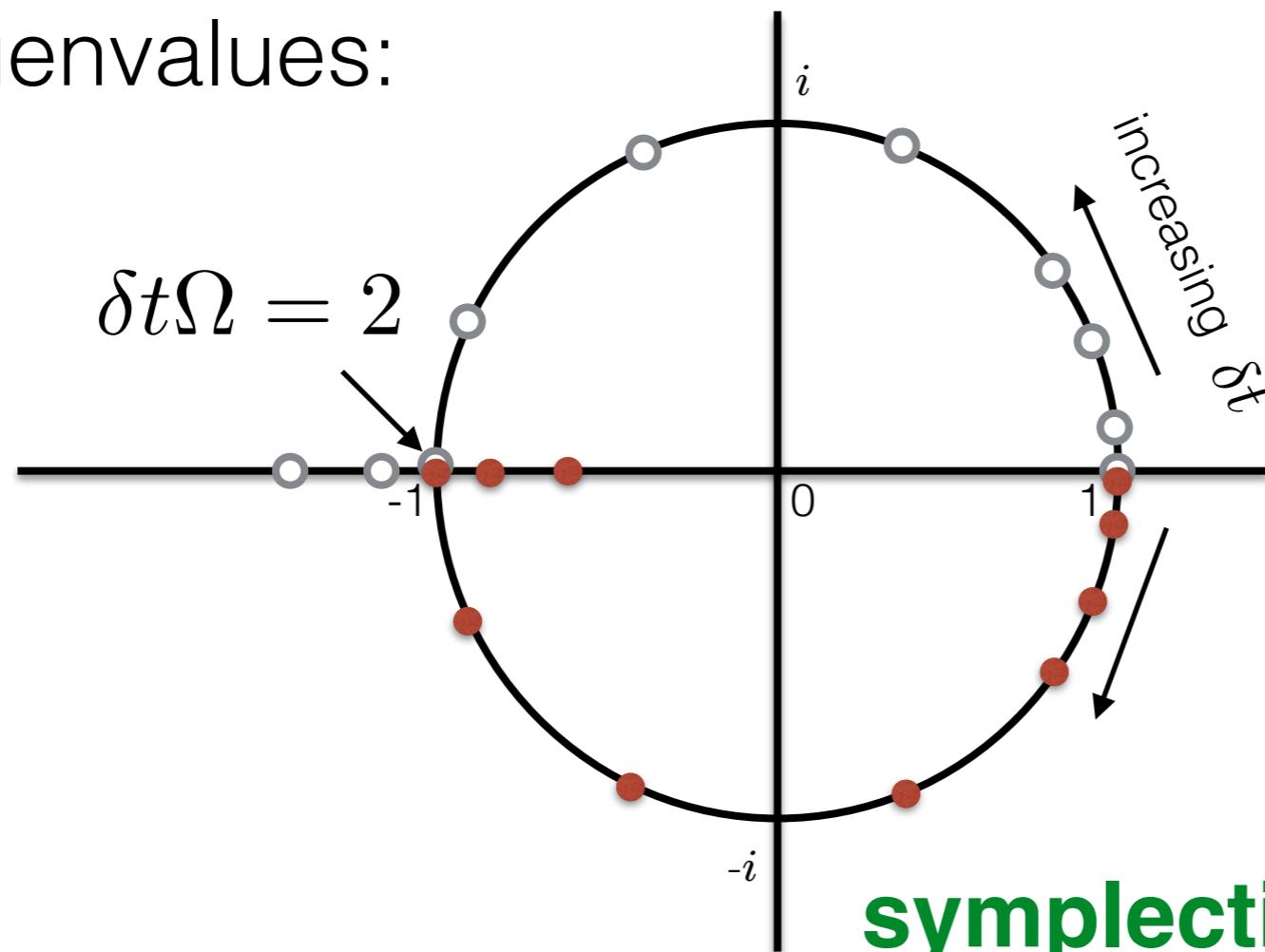
# **The problem of the timestep**

# Harmonic Oscillator

$$\dot{q} = p$$

$$\dot{p} = -\Omega^2 q$$

eigenvalues:



# leapfrog/Verlet

$$q_{n+1} = q_n + \delta t p_{n+1/2}$$

$$p_{n+1/2} = p_n - \frac{\delta t}{2} \Omega^2 q_n$$

$$p_{n+1} = p_{n+1/2} - \frac{\delta t}{2} \Omega^2 q_{n+1}$$

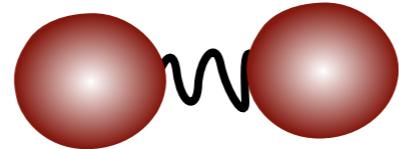
stable for  
 $\delta t \Omega \leq 2$

**symplectic** for all  $\delta t$   
but only **useful** for small  $\delta t$

# Stepsize Restriction

$$\text{H.O. } H = \frac{p^2}{2} + \frac{\Omega^2 q^2}{2}$$

think bonds



stability threshold

$$\delta t < 2/\Omega$$

in molecular dynamics  
bonds to H atoms have  
high frequencies

$$\delta t < 3\text{fs} = 3 \times 10^{-15}\text{s}$$

# The Timestep Problem in MD

Find timestepping methods that allow

$$\delta t \gg \delta t_{\text{Verlet}}$$

Three possibilities:

Use an **implicit method**

Eliminate the motion of fast components  
**constraints**

Isolate the stiff terms for efficient treatment  
**multiple timestepping**

# Implicit Methods

e.g. implicit midpoint (unconditionally stable)

$$\dot{z} = f(z) \quad z_{n+1} = z_n + h \frac{f(z_n) + f(z_{n+1})}{2}$$

Requires solution of a nonlinear system of equations by some iterative procedure, e.g. fixed point iteration,

$$z_{n+1}^{(k+1)} = z_n + \frac{h}{2}(f(z_n) + f(z_{n+1}^{(k)}))$$

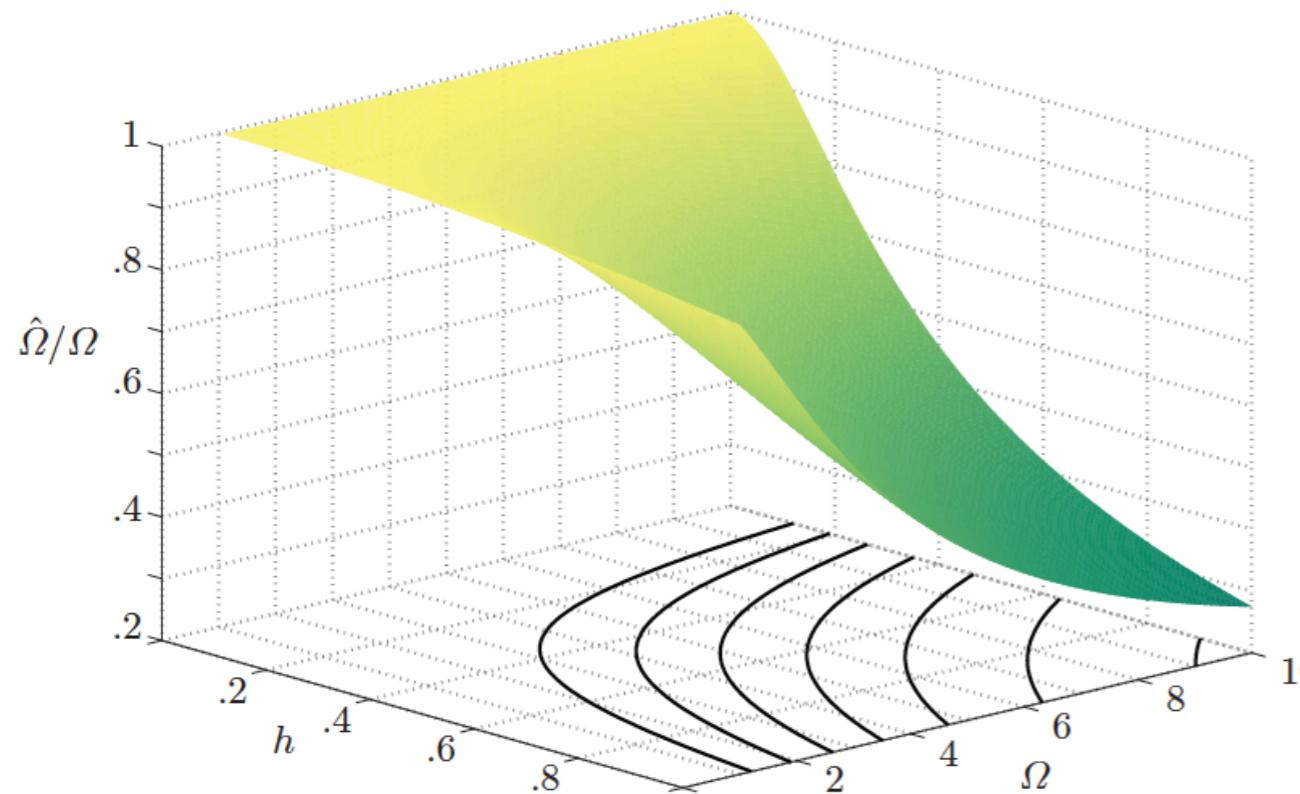
- need to **iterate to convergence**
- requires **multiple  $f$ -evaluations**

# Implicit Methods

Even if stable, a method can alter model frequencies

$$\dot{z} = f(z) \quad z_{n+1} = z_n + h \frac{f(z_n) + f(z_{n+1})}{2}$$

$$\dot{z} = i\Omega z \quad \hat{\Omega} = \frac{1}{ih} \ln \frac{1 + ih\Omega/2}{1 - ih\Omega/2}$$



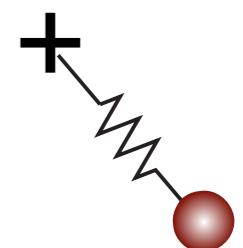
At large stepsize, high frequencies become low frequencies!

Leads to nonphysical resonances of dynamical modes, e.g. **bond stretch** directly impacting a **dihedral bend**...

# Stiff Spring Oscillator

energy:

$$H = \frac{|p|^2}{2m} + \frac{k(|q| - L)^2}{2} = E$$



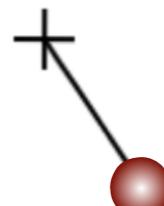
equations of motion:

$$\frac{dq}{dt} = p/m$$

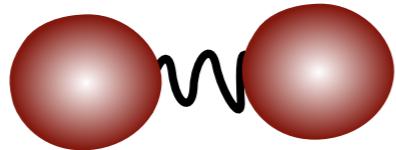
$$\frac{dp}{dt} = -k \left(1 - \frac{L}{|q|}\right) q$$

If energy is fixed, then in the limit of large k  
we must have  $|q| \sim L$

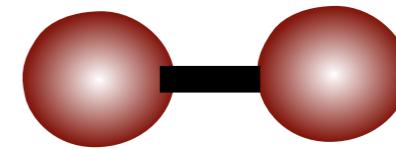
Constrained dynamics:  $|q| = L$



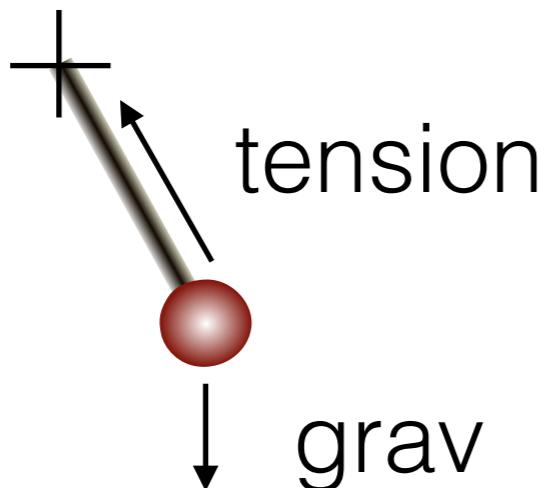
# Pendulum



stiff harmonic  
spring with rest length



rigid rod  
(holonomic constraint)



$$\dot{q} = m^{-1} p$$

$$\dot{p} = - \begin{bmatrix} 0 \\ g \end{bmatrix} - q\lambda$$

$$0 = q \cdot q - 1$$

# Symplectic Property

For the pendulum...

$$\begin{aligned}\dot{q} &= p/m \\ \dot{p} &= \lambda q \\ 0 &= \frac{1}{2}(\|q\|^2 - L^2)\end{aligned}$$

$$\begin{aligned}\dot{dq} &= dp/m \\ \dot{dp} &= \lambda dq + q d\lambda \\ 0 &= q \cdot dq\end{aligned}$$

$$\begin{aligned}\frac{d}{dt} dq \wedge dp &= dq \wedge \dot{dp} + \dot{dq} \wedge dp \\ &= dq \wedge (\lambda dq) + dq \wedge (qd\lambda) + (dp/m) \wedge dp \\ &= (q \cdot dq) \wedge d\lambda \\ &= 0\end{aligned}$$

# Constrained Hamiltonian Systems

$$\frac{d}{dt}q = \mathbf{M}^{-1}p$$

$$\frac{d}{dt}p = F - \sum_{i=1}^m \lambda_i \nabla g_i(q)$$

$$0 = g_j(q), \quad j = 1, 2, \dots, m$$

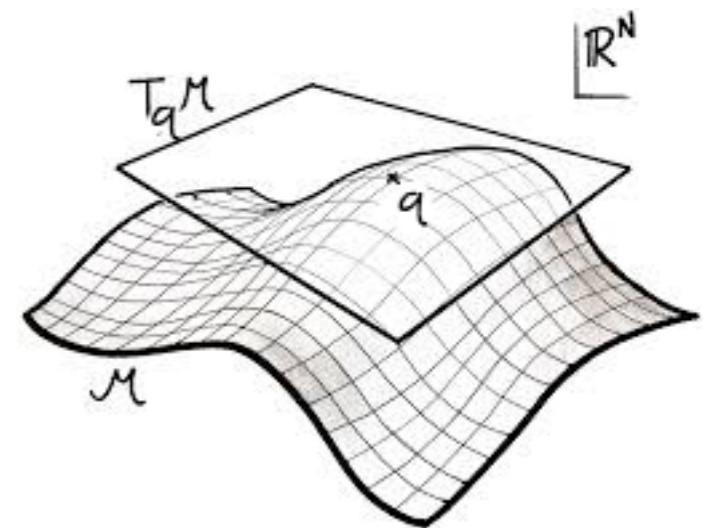
$$0 = \nabla g_j(q)^T \mathbf{M}^{-1} p, \quad j = 1, 2, \dots, m$$

**“hidden constraints”**

**“cotangent bundle”**

$$T^*\mathcal{M} = \{(q, p) \mid g_j(q) = 0, \nabla g_j(q)^T \mathbf{M}^{-1} p = 0, j = 1, 2, \dots, m\}$$

**symplectic**: dynamics preserves  $[dq \wedge dp]_{T^*\mathcal{M}}$



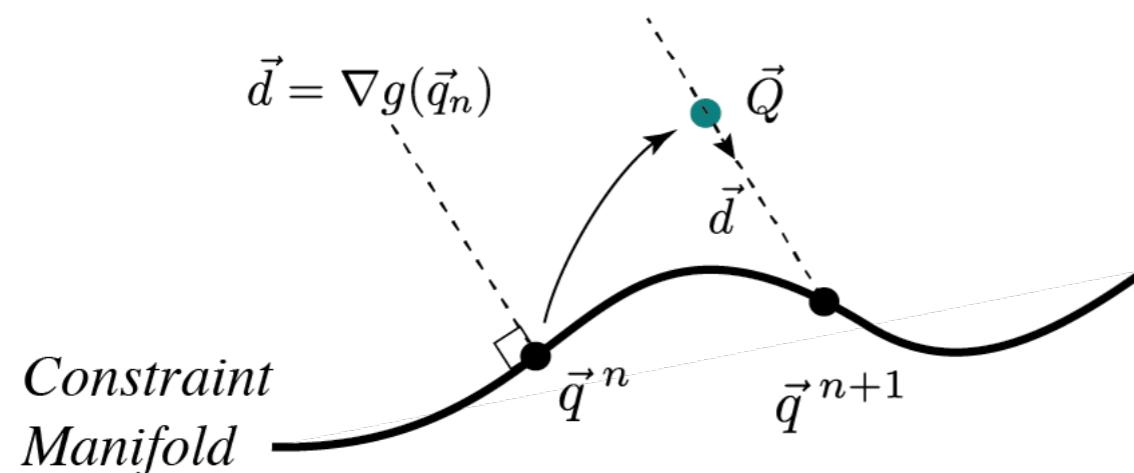
# SHAKE

[Berendsen, Ciccotti, Ryckaert 1977]

# RATTLE

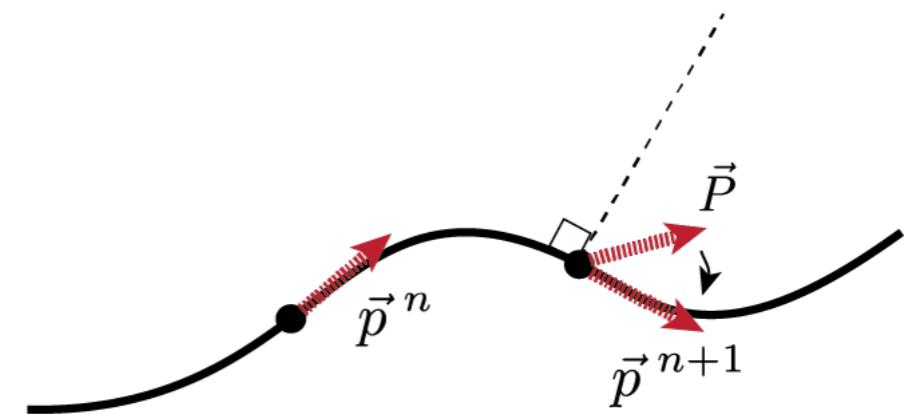
[Andersen 1983]

“SHAKE” Projection



$$\mathcal{M} = \{\vec{q} \mid g(\vec{q}) = 0\}$$

“RATTLE” Projection



L. & Skeel, J.Comput. Phys, 1994

**SHAKE** and **RATTLE** are **symplectic** methods  
and are actually the same method (**conjugate** methods).

# Constraint preserving methods

$$H(q, p) = |p|^2/2 + U(q)$$

$$\begin{aligned}\dot{q} &= p \\ \dot{p} &= -\nabla_q U(q) - \lambda \nabla g \\ 0 &= g(q) \\ 0 &= \nabla g(q) \cdot p\end{aligned}$$

$$(q, p) \mapsto (Q, P)$$

approximating a  
step in time of size  $h$

$$Q := q + h\tilde{P}$$

$$\tilde{P} := p - h\nabla U(q) - \lambda \nabla g(q)$$

$$0 = g(Q)$$

$$P := \tilde{P} - \mu \nabla g(Q)$$

$$0 = \nabla g(Q) \cdot P$$

first order

$$Q := q + (h/2)\tilde{P}$$

$$\tilde{P} := p - (h/2)\nabla U(q) - \lambda \nabla g(q)$$

$$0 = g(Q)$$

$$P := \tilde{P} - (h/2)\nabla U(Q) - \mu \nabla g(Q)$$

$$0 = \nabla g(Q) \cdot P$$

2nd order “RATTLE”

# Symplectic 2-form

**bilinear form**  $\mathbb{R}^{2n} \times \mathbb{R}^{2n} \rightarrow \mathbb{R}$

$$d\mathbf{q} \wedge d\mathbf{p}(\xi, \eta) = \sum_{i=1}^n dq_i \wedge dp_i(\xi, \eta) = \xi^T \mathbf{J} \eta$$

$$dq_i \wedge dp_i(\xi, \eta) = \xi_i \eta_{n+i} - \eta_i \xi_{n+i}$$

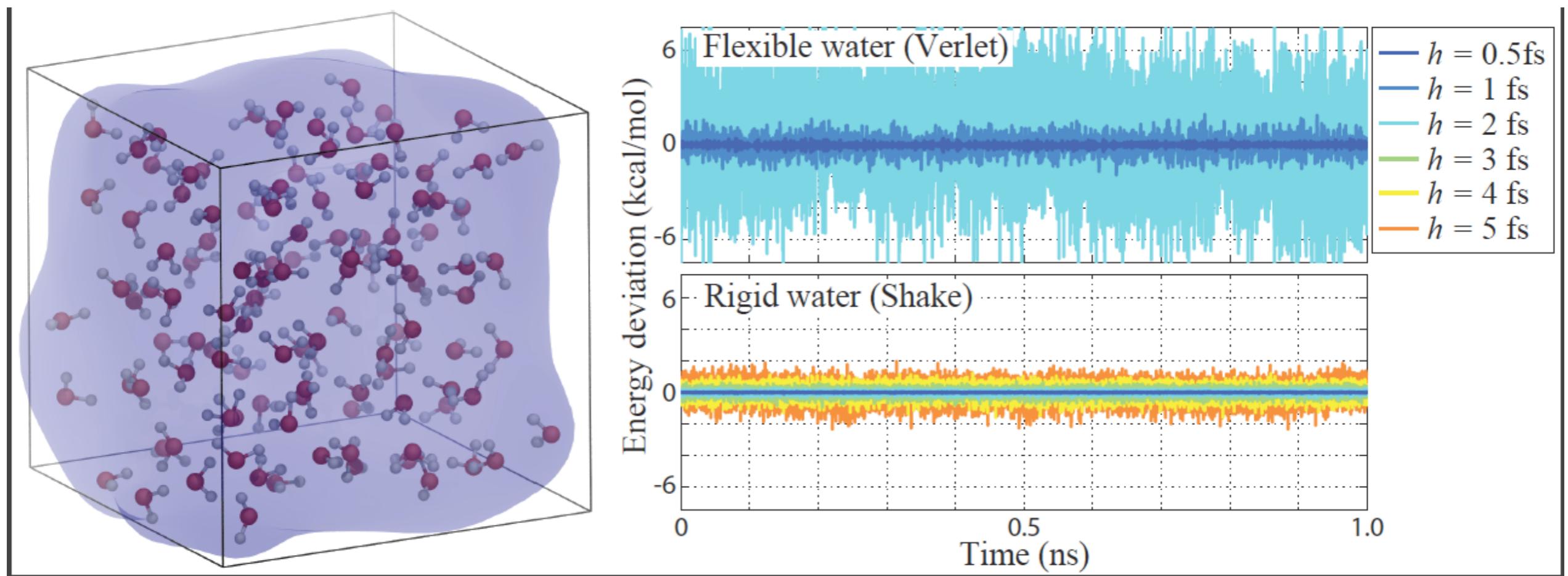
## properties

$$\mathbf{f} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^k \quad df = \mathbf{f}'(\mathbf{z})d\mathbf{z}$$

$$\begin{aligned} df \wedge (\mathbf{A}dg) &= -(\mathbf{A}dg) \wedge df \\ &= -dg \wedge (\mathbf{A}^T df) \end{aligned}$$

$$\mathbf{A} = \mathbf{A}^T \Rightarrow df \wedge (\mathbf{A}df) = 0$$

# Microcanonical Sampling



**Verlet** limited to about  
**2.7fs with stability**  
**1fs with 1 kcal/mol energy accuracy**

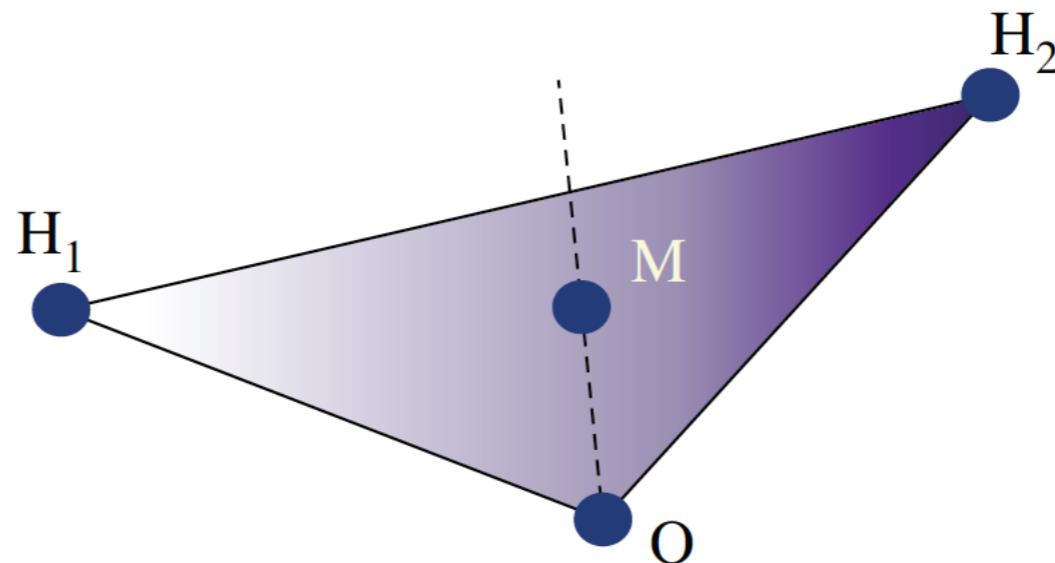
**SHAKE (RATTLE)** OK up to as much as  
**3-4fs with energy accuracy**

# Rigid Bodies

A lot of bio-molecular dynamics is just water

## TIP4P:

4 charge sites  
planar rigid body



For independent planar rigid bodies, the  
SHAKE/RATTLE equations  
can be solved analytically... [SETTLE Method]

Many advances specialized to rigid body systems,  
see especially

Dullweber, L. & McLachlan, J. Chem. Phys., 1997

Davidchack, Oulridge, Tretyakov, J. Chem Pys, 2014

# **Multiple timestepping**

# Multiple Timestepping (RESPA)

*Tuckerman, Martyna, Berne '90*

*also Grubmüller, Heller, Windemuth, Schulten '91*

Some forces are **fast-changing** (e.g. short-ranged)  
but **cheap** to compute

Some forces are **slow-changing** (e.g. long-ranged)  
and **costly** to compute

$$\begin{aligned} & \exp\left(\frac{\delta t}{2}\mathcal{L}_{U^{\text{slow}}}\right) \times \\ & \left[ \exp\left(\frac{\delta t}{2r}\mathcal{L}_{U^{\text{fast}}}\right) \exp\left(\frac{\delta t}{r}\mathcal{L}_K\right) \exp\left(\frac{\delta t}{2r}\mathcal{L}_{U^{\text{fast}}}\right) \right]^r \\ & \times \exp\left(\frac{\delta t}{2}\mathcal{L}_{U^{\text{slow}}}\right) \end{aligned}$$

Many fast cheap evaluations  
Few slow costly ones

# Linear Model Problem

model problem

$$H = \frac{p^2}{2} + \frac{(1 + \Omega^2)q^2}{2} \quad \Omega \gg 1$$

**idealized** *multiple timestepping (fast solve is exact)*

$$H(q, p) = p^2/2 + U_{\text{S}}(q) + U_{\text{F}}(q)$$

$$U_{\text{S}}(q) = q^2/2$$

$$U_{\text{F}}(q) = \Omega^2 q^2/2$$

Kick with  $U_{\text{S}}$

$$\delta t/2$$

Solve  $H_{\text{F}} = p^2/2 + U_{\text{F}}(q)$

$$\delta t$$

Kick with  $U_{\text{S}}$

$$\delta t/2$$

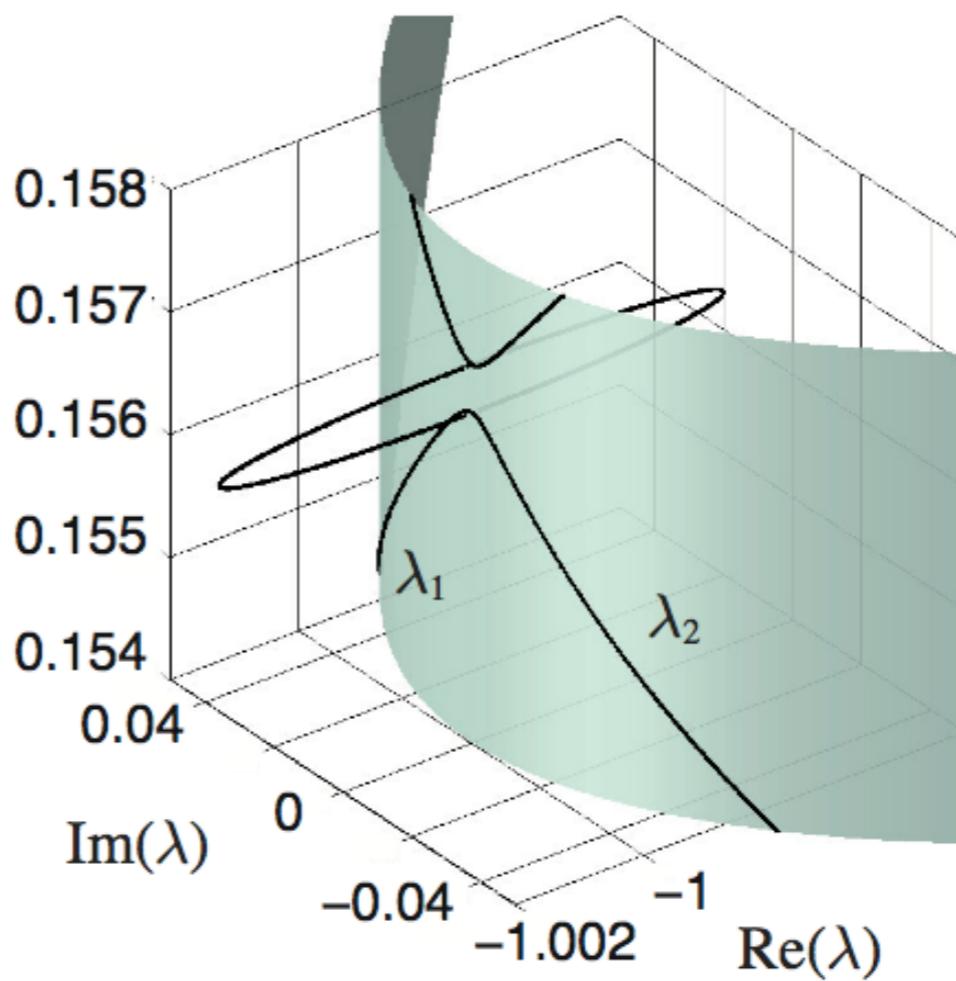
# Resonance

**Eigenvalues of timestep map:**

$$\lambda_1 \lambda_2 = 1; \quad \lambda_1 + \lambda_2 = 2 \cos(\delta t \Omega) - \frac{\delta}{\Omega} \sin(\delta t \Omega)$$
$$\varepsilon \propto \delta t - \pi/\Omega < 0 \quad \lambda = -1 \pm (|\varepsilon|/2)^{1/2} + O(\varepsilon)$$

**instabilities...**

$\delta t$



# Resonance

For the harmonic model, Verlet introduces a stability restriction

$$\delta t < 2/\Omega$$

Multiple timestepping, in the idealized form described here has a stability restriction of about

$$\delta t < \pi/\Omega$$

i.e., not very dramatic improvement. In practice, we are limited to around **3.5fs**

# Mollified Impulse Method

*Garcia-Archilla, Sanz-Serna and Skeel 1998*

To stabilize multiple timestepping, one idea is to average out over an associated fast dynamics to ``mollify'' the impulse in RESPA

starting from e.g  $(q, 0)$ , solve

$$H^{\text{fast}}(\mathbf{q}, \mathbf{p}) = \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}/2 + U_F(\mathbf{q})$$

to produce a sequence  $\theta_0 = q, \theta_1, \theta_2, \dots, \theta_K$   
then define:

$$\mathcal{A}(\mathbf{q}) = \frac{1}{K+1} \sum_{i=0}^K \phi_i \theta_i(\mathbf{q})$$

Advance the step using a *mollified impulse*:

$$\tilde{U}(q) := U_s(\mathcal{A}(q))$$

# Pushing the limits

**SHAKE-MOLLY** [*Izaguirre, Reich, Skeel 1999*]

constrains the slow force evaluations to lie exactly at the bond stretch minima (multiple timestepping + constraints).

With the very best **deterministic** schemes, however, we find for a biological molecule:

$$\delta t \lesssim 5\text{fs}$$

Bob Skeel's license plate ca. 2000: **FEMTO5**

# Stochastic Methods

Introducing random perturbations might seem to complicate the numerical integrator.

In fact, if done correctly, it is possible to gain in two ways:

- 1) a significant **stability improvement** is possible
- 2) substantially **higher accuracy** is possible by shifting emphasis to the **invariant distribution**

These benefits carry over to **multiple timestepping** and **constraints**, and **the combination can be even more powerful than expected!!**

# Next Up

**Wednesday** - Stochastic differential equations -  
Brownian dynamics • Euler-Maruyama • SDE accuracy •  
Leimkuhler-Matthews • Langevin dynamics • splitting • BAOAB and ABOBA

**Thursday** - push up stepsize in SDE numerics • multiple  
timestepping for SDEs • Constraints • g-BAOAB • isokinetic  
multiple timestepping

**Friday** - thermostats (deterministic and stochastic) • ergodic properties  
• applications - nonequilibrium and in data science