Molecular Dynamics in Dyalog APL

Richard Park

$$
\begin{aligned}
p_{i,n+1/2} &= p_{i,n} - \frac{h}{2} s_n V_n^{1/3} \nabla_i U(V^{1/3}\mathbf{q}) \\
\pi_{v,n+1/2} &= \pi_{v,n} + \frac{h}{2} s_n [\mathcal{P}(\mathbf{q}_n, \mathbf{p}_{n+1/2}, V_n, s_n) - P_{ext}] \\
\pi_{s,n+1/2} &= \pi_{s,n} + \frac{h}{2} \left(\sum_{i=1}^{N} \frac{p_{i,n+1/2}^2}{m_i V_n^{2/3} s_n^2} - g k_B T \right) \\
&\quad - \frac{h}{2} \Delta\mathcal{H}(\mathbf{q}_n, \mathbf{p}_{n+1/2}, V_n, \pi_{v,n+1/2}, s_n, \pi_{s,n+1/2}) \\
s_{n+1} &= s_n + \frac{h}{2}(s_n + s_{n+1}) \frac{\pi_{s,n+1/2}}{Q_s} \\
V_{n+1} &= V_n + \frac{h}{2}(s_n + s_{n+1}) \frac{\pi_{v,n+1/2}}{Q_v} \\
q_{i,n+1} &= q_{i,n} + \frac{h}{2} \left(\frac{1}{s_n V_n^{2/3}} + \frac{1}{s_{n+1} V_{n+1}^{2/3}} \right) \frac{p_{i,n+1/2}}{m_i} \\
\pi_{s,n+1} &= \pi_{s,n+1/2} + \frac{h}{2} \left(\sum_{i=1}^{N} \frac{p_{i,n+1/2}^2}{m_i V_{n+1}^{2/3} s_{n+1}^2} - g k_B T \right) \\
&\quad - \frac{h}{2} \Delta\mathcal{H}(\mathbf{q}_{n+1}, \mathbf{p}_{n+1/2}, V_{n+1}, \pi_{v,n+1/2}, s_{n+1}, \pi_{s,n+1/2}) \\
\pi_{v,n+1} &= \pi_{v,n+1/2} + \frac{h}{2} s_{n+1} [\mathcal{P}(\mathbf{q}_{n+1}, \mathbf{p}_{n+1/2}, V_{n+1}, s_{n+1}) - P_{ext}] \\
p_{i,n+1} &= p_{i,n+1/2} + \frac{h}{2} s_{n+1} V_{n+1}^{1/3} \nabla_i U(V_{n+1}^{1/3}\mathbf{q_{n+1}})
\end{aligned}
$$

What is MD?

Modelling atomic and molecular systems using classical (Newtonian) mechanics

$$\vec{F} = m\vec{a}$$

$$F = -\frac{dV(x)}{dx}$$

Numerical integration

- Insoluble ODEs
- Step from state to state using numerical approximations
- Velocity Verlet

More details

- Webinar

  https://dyalog.tv/Webinar/?v=GDp89NbSW5Q

- GitHub

  https://github.com/rikedyp/APLPhys

APLPhys: Writing an MD engine

-]LINK
- Maths
- Arrays
- Operators

Text files

Functions and operators

]LINK.Create APLPhys “\path\to\APLPhys”

Text files

LAMMPS: *.in

HOOMD: *python

APLPhys: .aplphys

Maths

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

```
LJ ← {4×-/ω∘.*¯12 ¯6}
```

Maths

$$\mathbf{a}_{\mathbf{b}} = (\mathbf{a} \cdot \hat{\mathbf{b}})\hat{\mathbf{b}}$$

```
Mag←{0.5*⍨+/⍵*2}      ⍝ Magnitude of vector in Euclidean space

Norm←÷∘Mag⍨           ⍝ Normalised vector

InDirOf←(⊢×+.×)∘Norm  ⍝ Component of vector ⍺ in direction of ⍵
```

Arrays

Position vector (x, y, z)
  Similar for velocity, acceleration (forces)

Matrix of position vectors

```
0.1315377881   0.7556053222   0.4586501319
0.5327672374   0.2189591863   0.04704461621
0.6788647169   0.6792964058   0.9346928959
0.3835020775   0.5194163721   0.8309653461
0.03457211053  0.05346163504  0.5297001933
```

Arrays

```
∘.-⍨↓pos
```

```
-⍤1⍤1 99⍨pos
```

Operators

```
next & previous:
pos vel acc ene_pot_avg ene_kin_avg temp press
```

```
next ← ComputeForces RunStyle previous
```

Operators

```
next & previous:
pos vel acc ene_pot_avg ene_kin_avg temp press
```

```
next ← ComputeForces RunStyle ⍣nsteps⊢ previous
```

APLPhys: Visualising MD

- MiServer
- DUI

- HTML5 Canvas

APLPhys: Visualising MD

- MiServer
- DUI
- Babylon JS 

Future

- Clean up this mess
- Speed: co-dfns?
- Scalability: Isolates?
- GUI
  - Download rendered videos
  - Load pre-calculated simulations
  - Graphs e.g. temperature, kinetic and potential energy
- + much more….

APLPhys

Github.com/rikedyp/APLPhys