Molecular Dynamics in Dyalog APL

Richard Park
\[ p_{i,n+1/2} = p_{i,n} - \frac{h}{2} s_n V_n^{1/3} \nabla_i U(V_n^{1/3} q) \]

\[ \pi_{v,n+1/2} = \pi_{v,n} + \frac{h}{2} s_n \left[ \mathcal{P}(q_n, p_{n+1/2}, V_n, s_n) - P_{ext} \right] \]

\[ \pi_{s,n+1/2} = \pi_{s,n} + \frac{h}{2} \left( \sum_{i=1}^N \frac{p_{i,n+1/2}^2}{m_i V_n^{2/3} s_n^2} - g k_B T \right) \]

\[ -\frac{h}{2} \Delta \mathcal{H}(q_n, p_{n+1/2}, V_n, \pi_{v,n+1/2}, s_n, \pi_{s,n+1/2}) \]

\[ s_{n+1} = s_n + \frac{h}{2} (s_n + s_{n+1}) \frac{\pi_{s,n+1/2}}{Q_s} \]

\[ V_{n+1} = V_n + \frac{h}{2} (s_n + s_{n+1}) \frac{\pi_{v,n+1/2}}{Q_v} \]

\[ q_{i,n+1} = q_{i,n} + \frac{h}{2} \left( \frac{1}{s_n V_n^{2/3}} + \frac{1}{s_{n+1} V_{n+1}^{2/3}} \right) \frac{p_{i,n+1/2}}{m_i} \]

\[ \pi_{s,n+1} = \pi_{s,n+1/2} + \frac{h}{2} \left( \sum_{i=1}^N \frac{p_{i,n+1/2}^2}{m_i V_{n+1}^{2/3} s_{n+1}^2} - g k_B T \right) \]

\[ -\frac{h}{2} \Delta \mathcal{H}(q_{n+1}, p_{n+1/2}, V_{n+1}, \pi_{v,n+1/2}, s_{n+1}, \pi_{s,n+1/2}) \]

\[ \pi_{v,n+1} = \pi_{v,n+1/2} + \frac{h}{2} s_{n+1} \left[ \mathcal{P}(q_{n+1}, p_{n+1/2}, V_{n+1}, s_{n+1}) - P_{ext} \right] \]

\[ p_{i,n+1} = p_{i,n+1/2} + \frac{h}{2} s_{n+1} V_{n+1}^{1/3} \nabla_i U(V_{n+1}^{1/3} q_{n+1}) \]
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

]LI NK.Create APLPhys "\path\to\APLPhys"
Text files

LAMMPS: *.in
HOOMD: *python

APLPhys: .aplphys
Maths

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

\[ \text{LJ} \leftarrow \{ 4\times-/\omega . \times^-12 \ \ -6 \} \]
Maths

\[ \mathbf{a} \cdot \mathbf{b} = (\mathbf{a} \cdot \hat{\mathbf{b}}) \hat{\mathbf{b}} \]

Mag←{0.5*网购+/\omega*2}  A Magnitude of vector in Euclidean space

Norm←÷Mag  A Normalised vector

InDirOf←(⊢×+.x)*Norm  A Component of vector \( \alpha \) in direction of \( \omega \)
Arrays

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

Matrix of position vectors

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1315377881</td>
<td>0.7556053222</td>
<td>0.4586501319</td>
</tr>
<tr>
<td>0.5327672374</td>
<td>0.2189591863</td>
<td>0.04704461621</td>
</tr>
<tr>
<td>0.6788647169</td>
<td>0.6792964058</td>
<td>0.9346928959</td>
</tr>
<tr>
<td>0.3835020775</td>
<td>0.5194163721</td>
<td>0.8309653461</td>
</tr>
<tr>
<td>0.03457211053</td>
<td>0.05346163504</td>
<td>0.5297001933</td>
</tr>
</tbody>
</table>
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