Name: Yubo Ouyang

Periodic Boundary Conditions

Introduction

Question: What is the output from the following commands?

```
box_length = 1.0
pset = ParticleSet(1,1)
pset.change_pos(0,np.array([0.501,0.499,0.501]))
print( pos_in_box(pset.pos(0),box_length) )
pset.change_pos(0,np.array([1.000, -.501, 4.501]))
print( pos_in_box(pset.pos(0),box_length) )
```

Answer:

[-0.499 0.499 -0.499]
[0. 0.499 -0.499]

Minimum Image Convention

Question: What is the output for the following commands?

```
box_length = 1.0
pset = ParticleSet(2,1)
pset.change_pos(0,np.array([ 0.499, 0.000, 0.000 ]) )
pset.change_pos(1,np.array([ -.499, 0.000, 0.000 ]) )
print( distance(0,1,pset,box_length) )

pset.change_pos(0,np.array([ 0.001, 0.000, 0.000 ]) )
pset.change_pos(1,np.array([ -.001, 0.000, 0.000 ]) )
print( distance(0,1,pset,box_length) )
```

Answer:

0.002
0.002

Integrator

Question: Prove that the Verlet algorithm is time-reversal invariant. That is, show that if we use r(t+h) and r(t) as inputs we get r(t-h) (up to round-off errors).

Answer:

The Verlet algorithm is: $r(t+\Delta t) = 2r(t) - r(t-\Delta t) + a(t)\Delta t^2$

When we use $r(t+h)$ and $r(t)$ as inputs, $\Delta t = -h$, and we get:

$r(t-h) = 2r(t) - r(t+h) + a(t)h^2$ Thus, Verlet algorithm is time-reversal invariant.

Question: State two possible problems with finite-precision arithmetic about the Verlet time-stepping algorithm.
Answer: (1) The trajectory is not very accurate for long time steps (2) The kinetic energy, related to velocity, cannot be calculated very precisely

Question: The Velocity Verlet integrator is algebraically equivalent to Verlet but avoids taking differences between large numbers. Prove that velocity Verlet is equivalent to Verlet. Which is preferable?
Answer:
To prove velocity Verlet is equivalent to Verlet, we need to replace the velocity in the equation. Using the position function of velocity Verlet, we can get:

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

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

By subtraction, we get:

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

The velocity equation is:

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

Substitution of the velocity equation yields:

$r(t+2\Delta t) = 2r(t+\Delta t) - r(t) + a(t+\Delta t)\Delta t^2$, which is exactly the Verlet integrator.

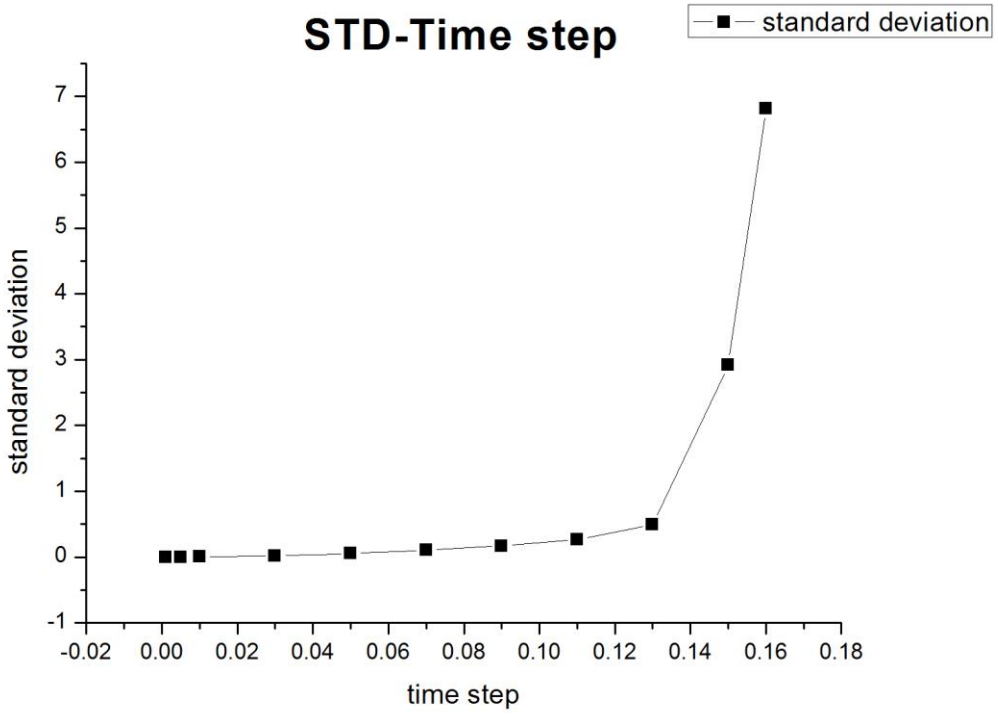
Velocity Verlet is preferable, because it provides better estimate of velocity.

Time Steps
Question: Make a graph of the standard deviation of the total energy versus time step size. In other words, you are trying to see how the energy fluctuates with respect to your choice of time step. For this you need to make several simulations using as many as 10 different time steps.
Answer: Using the analyze_trace.py in HW1, standard deviation can be obtained for different time steps:

Time step	Standard deviation
0.001	0.0001
0.005	0.0011
0.01	0.0028
0.03	0.0192
0.05	0.0532
0.07	0.1054
0.09	0.1680
0.11	0.2664
0.13	0.4960

0.15	2.9169
0.16	6.8152

As can be seen from the graph, STD increases when time steps become larger. Around time step = 0.15, there is a sharp increase in STD.



Question: What happens if you pick a very large time step? Why?
Answer: When I pick a very large time step, the console shows:

```
G:\UIUC\course\MSE 485\HW2\mdsim.py:146: RuntimeWarning: divide by zero encountered in double_scalars
  force += (-24 / dist ** 2) * (2 * (1 / (dist ** 12)) - 1 / (dist ** 6)) * disp
G:\UIUC\course\MSE 485\HW2\mdsim.py:146: RuntimeWarning: invalid value encountered in double_scalars
  force += (-24 / dist ** 2) * (2 * (1 / (dist ** 12)) - 1 / (dist ** 6)) * disp
```

which means that in the force calculation, the distance between two atoms are very small. In other words, atoms become very close.
The reason is that verlet algorithm estimates positions and velocities with errors, which will become larger when time steps become larger. As a result, the whole system will be unstable and energy conservation will be invalid.

Question: From your graph, determine the largest time step one can use and still maintain energy conservation (a stable mean and fluctuations within 5% of the mean).
Answer: The stable mean is around -257.5, so fluctuations (standard deviation) should be less than 13.
By checking the graph, the largest time step can be determined about 0.15. Thus, we should probably work with a time step about 0.015 or so.