Practise 2

I did not get correct results. I failed to find the errors on time. If you have some insight on them, please tell me and I'll send a proper report.

System functions:

```python
def euler(part, vel1, dt, L, cutoff, m):
    forces, pot = find_force_LJ0(part, cutoff, L)
    part2 = part + vel1 * dt + 0.5 * forces * dt ** 2 / m
    vel2 = vel1 + forces * dt / m
    kin = kinetic_energy(vel2)
    return part2, vel2, pot, kin


def vel_verlet(part1, vel1, dt, L, cutoff, m):
    forces, pot = find_force_LJ0(part1, cutoff, L)
    part2 = part1 + vel1 * dt + 0.5 * forces * dt ** 2 / m
    forces2, pot = find_force_LJ0(part2, cutoff, L)
    vel2 = vel1 + (forces + forces2) * 0.5 * dt
    kin = kinetic_energy(vel2)
    return part2, vel2, pot, kin


def verlet(part1, part2, dt, L, cutoff, m):
    forces, pot = find_force_LJ0(part1, cutoff, L)
    part3 = 2.0 * part2 - part1 + forces * dt ** 2.0 / m
    vel = (part3 - part1) / (2.0 * dt)
    kin = kinetic_energy(vel)
    return part2, part3, pot, kin
```

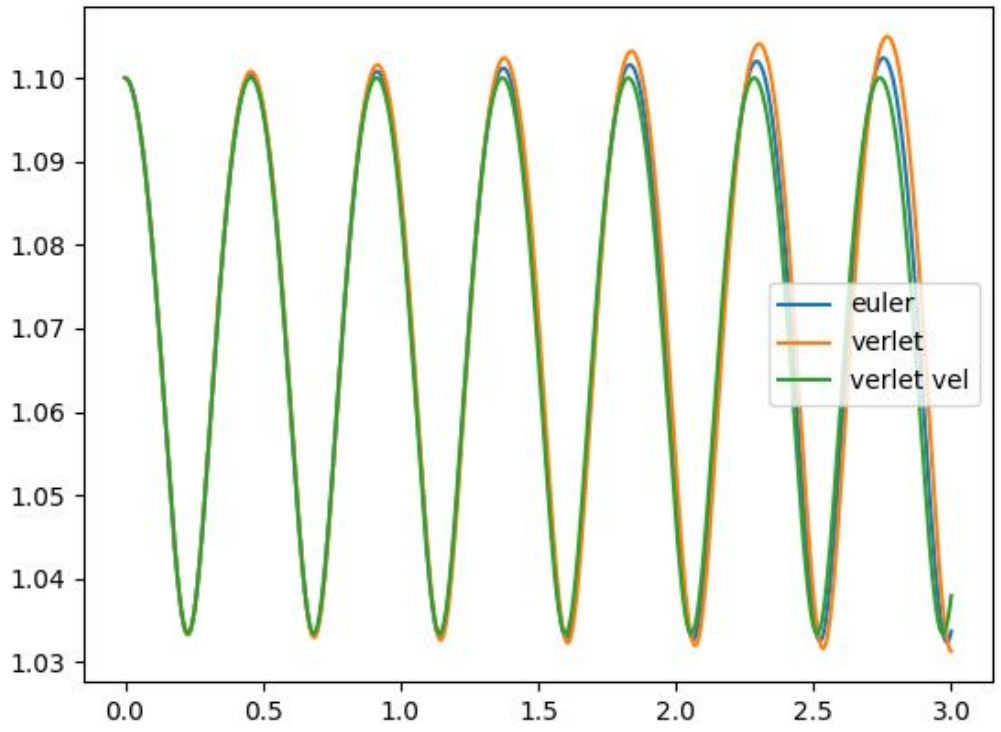
Write to file function

```python
def write_file(particles, filetype):
    f = open(filetype, "a")
    f.write(str(len(particles)) + "\n\n")
    for particle in particles:
        string = ""
        for j in particle:
            string += str(j) + " "
        f.write("C " + string + "\n")
    f.close
```

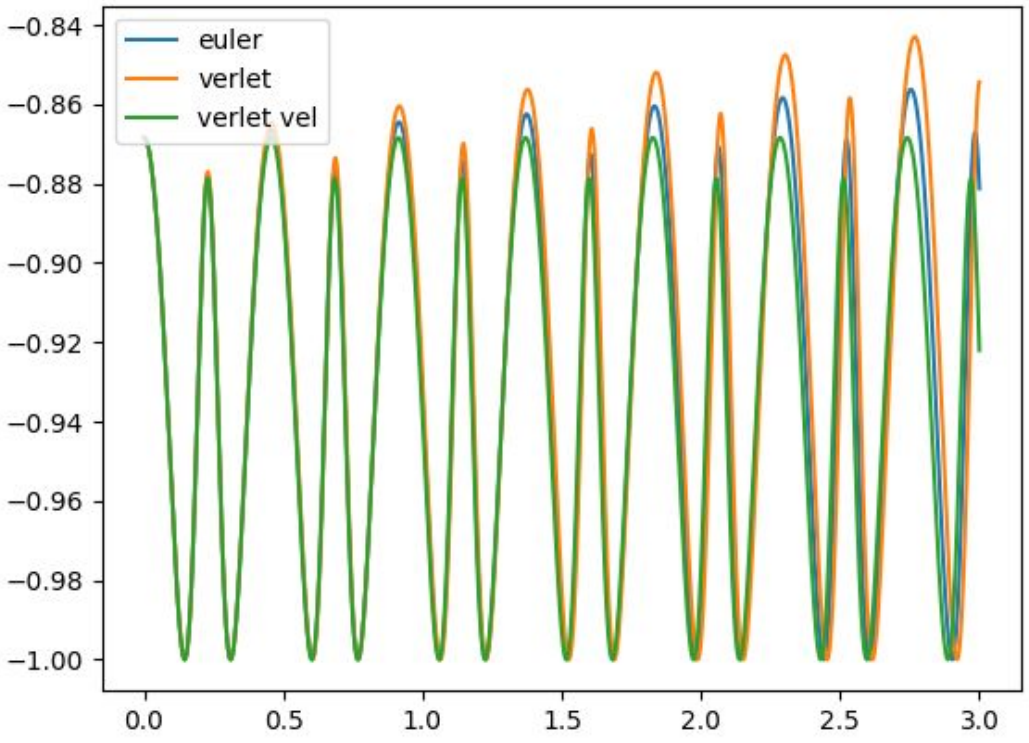
Main loop

```python
import numpy as np
from funcions.write_xyz import write_file
from funcions.system_functions import find_force_LJ0, vel_verlet, verlet, euler
import matplotlib.pyplot as plt

part1 = np.array([[0, 1.1, 0], [0, 0, 0]])
part2 = np.copy(part1)
L = 10
dt = 0.0003
iteracions = 10000
m = 1

pose, posv, posvv = [], [], []
pote, potv, potvv = [], [], []
cine, cinv, cinvv = [], [], []
enee, enev, enevv = [], [], []

t = np.arange(0, iteracions) * dt
vel = np.zeros(np.shape(part1))
for i in range(iteracions):
    # write_file(part1, "teest")
    part1, vel, pot, kin = euler(part1, vel, dt, L, L / 2, m)
    pose.append(abs(part1[0][1] - part1[1][1]))
    enee.append(pot + kin)
    pote.append(pot)
    cine.append(kin)

part1 = np.array([[0, 1.1, 0], [0, 0, 0]])
vel = np.zeros(np.shape(part1))
for i in range(iteracions):
    part1, vel, pot, kin = vel_verlet(part1, vel, dt, L, L / 2, m)
    posvv.append(abs(part1[0][1] - part1[1][1]))
    enevv.append(pot + kin)
    potvv.append(pot)
    cinvv.append(kin)

part1 = np.array([[0, 1.1, 0], [0, 0, 0]])
part2 = np.array([[0, 1.1, 0], [0, 0, 0]])
for i in range(iteracions):
    part1, part2, pot, kin = verlet(part1, part2, dt, L, L / 2, m)
    posv.append(abs(part1[0][1] - part1[1][1]))
    enev.append(pot + kin)
    potv.append(pot)
    cinv.append(kin)

tote = [pose, pote, cine, enee]
totv = [posv, potv, cinv, enev]
totvv = [posvv, potvv, cinvv, enevv]

plot = 0  # 0 for distances, 1, 2, or 3 for potential, kinetic and total energy

plt.plot(t, pose, label="euler")
plt.plot(t, totv[plot], label="verlet")
plt.plot(t, totvv[plot], label="verlet vel")
plt.legend()
plt.show()
```

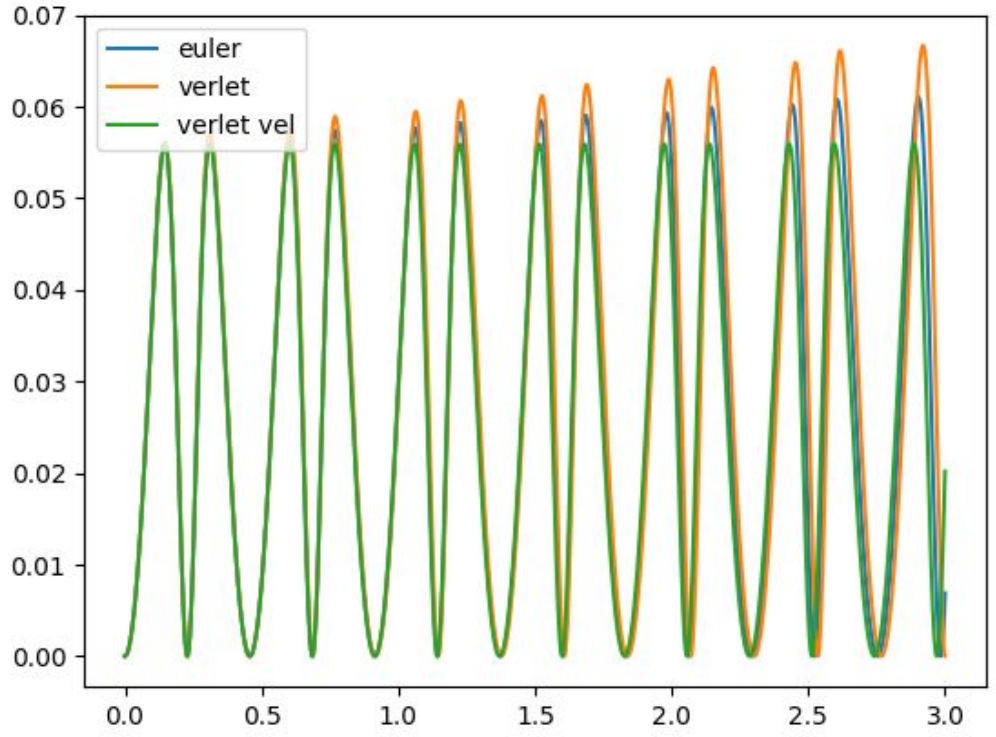
Plots:
Distances



Potential energy



Kinetic energy



Total energy