Study of phase transitions in argon using molecular dynamics.

The task is to transfer the force calculations from the CPU to the GPU in a program simulating phase transitions of argon.

In the first part of the task, we will write a C/C++ program to simulate a cluster of argon atoms using the molecular dynamics method.

In the second part, transfer the algorithm to the GPU using C for CUDA.

In the third part, we will use the program to perform an interesting simulation of a physical phenomenon using the scalar version on the CPU and the GPU version and draw conclusions.

LENNARD-JONES

The interaction between argon atoms can be described by the following Lennard-Jones potential:

$$V_{ij} = \epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right)$$

The parameters

σ_{ij} is the equilibrium distance between argon atoms

ε_{ij} the interaction energy at the equilibrium distance

Since all atoms in the system are identical, σij and εij are the same for all pairs of atoms.

rij being the distance between atoms i and j.

σij = σ = 0.369 nm

εij = ε = 1.19 kJ/mol

TOTAL POTENTIAL ENERGY OF THE SYSTEM

The total potential energy of an LJ-system is:

$$V = \frac{1}{2}\sum_{i\neq j} V_{ij} = \frac{1}{2}\sum_{i\neq j}\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - 2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right)$$

TOTAL KINETIC ENERGY OF THE SYSTEM

The total kinetic energy of an LJ-system is:

$$T = \frac{1}{2}\sum_{i}\frac{mv_i^2}{2}$$

TOTAL ENERGY OF THE SYSTEM

The total energy is given by:

E = T + V

FORCE VECTORS

In an MD simulation, we need to compute the forces acting on each particle. The force vector acting on the i-th atom can be obtained from the formula?

$$F_x^i = -\frac{\partial V}{\partial x_i},\ F_y^i = -\frac{\partial V}{\partial y_i},\ F_z^i = -\frac{\partial V}{\partial z_i}$$

So,

$$F_x^i = -\frac{\partial V}{\partial x_i} = -\sum_j \frac{\partial}{\partial x_i} V_{ij} = -\sum \epsilon \left(\frac{\partial}{\partial x_i} \frac{\sigma^{12}}{r_{ij}^{12}} - 2 \frac{\partial}{\partial x_i} \frac{\sigma^6}{r_{ij}^6} \right)$$

$$F_x^i = \sum_j \epsilon \left(12 \frac{\sigma^{12}}{r_{ij}^{12}} - 12 \frac{\sigma^6}{r_{ij}^6} \right) \frac{1}{r_{ij}} \frac{\partial}{\partial x_i} r_{ij}$$

$$F_x^i = 12 \sum_j \epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) \frac{1}{r_{ij}} \frac{\partial}{\partial x_i} \sqrt{(x^i - x^j)^2 + (y^i - y^j)^2 + (z^i - z^j)^2}$$

$$F_x^i = 12 \sum_j \epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) \frac{1}{r_{ij}} \frac{1}{2} \frac{2((x^i - x^j)}{\sqrt{(x^i - x^j)^2 + (y^i - y^j)^2 + (z^i - z^j)^2}}$$

$$F_x^i = 12 \sum_j \epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) \frac{1}{r_{ij}} \frac{(x^i - x^j)}{r_{ij}}$$

$$F_x^i = 12 \sum_j \epsilon \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) \frac{(x^i - x^j)}{r_{ij}^2}$$

Similarly, we calculate the coordinates F^{i}y, and F^{i}z.

To integrate the equations of motion, the leap-frog method is used.

SIMULATION GRID

As the model system for CPU, a system of 864 argon atoms arranged in a 6x6x6 grid of elementary cells should be considered.

For GPU, a system of 4000 argon atoms arranged in a regular grid of 10x10x10 elementary cells should be considered. Each elementary cell contains four argon atoms.

Coordinates of atoms in the elementary cell:

(0.5a, 0.5a, 0.5a), (0.5a, 1.5a, 1.5a), (1.5a, 0.5a, 1.5a), (1.5a, 1.5a, 0.5a)
where the length unit a is determined by the equation:

σ =√2*a

and the elementary cell is a cube with a side length of 2*a.

In the program, the size of the grid (number of elementary cells) can be set as a constant defined in the header file or as a parameter to allow scaling of the system size.

NOTE
After constructing the system, it should be moved in such a way that the center of mass of the system is located at the point (0.0, 0.0, 0.0).

VELOCITY INITIALIZATION

Initial velocities are assigned randomly using a uniform distribution. The distribution will have a mean of zero. The variance of the distribution will be such that the average kinetic energy per degree of freedom is kB*T, where kB is the Boltzmann constant and T is the temperature.

Under equilibrium conditions, the average kinetic energy per degree of freedom is (kB*T)/2. However, our system in the starting position is at the potential energy minimum. As a result, about half of the kinetic energy will convert into potential energy during the simulation, the excess potential energy above the minimum to be specific.

Thus, the average kinetic energy should be: (m*v^2)/2 = kB*T

$$\frac{m\bar{v}^2}{2} = k_BT$$

Therefore, v^2 = (2*kB*T)/2

$$\bar{v}^2 = \frac{2k_BT}{2}$$

Considering that the variance of the uniform distribution on the interval (-vm, vm) is $\frac{4v_m^2}{12}$ (4*vm^2)/12,   we obtain the condition:

$$\frac{4v_m^2}{12} = \frac{2k_BT}{m}$$

(4*vm^2)/12 = (2*kB*T)/m
Thus,

$$v_m^2 = \sqrt{\frac{24k_BT}{4m}} = \sqrt{\frac{6k_BT}{m}}$$

vm^2 = sqrt((24*kB*T)/(4*m)) = sqrt((6*kB*T)/m)

Therefore, we assign initial velocities to the atoms from the interval

$$\left(-\sqrt{\frac{6kT}{m}},\sqrt{\frac{6kT}{m}}\right).$$

(-sqrt((6*k*T)/m), +sqrt((6*k*T)/m))

NOTE

After assigning all velocities, the velocity of the center of mass should be set to zero! This is done by calculating the average velocity of the atoms for each coordinate and subtracting that value from the velocity of each atom.

TRUNCATION IN A BALLOON.

For avoiding a situation where argon atoms in the gas phase disperse infinitely, we enclose the system in a "balloon" by adding an additional term to the potential energy in the following form:

```
\[ V_i = \begin{cases}
    \frac{1}{B} (r_i - r_B)^2 & \text{for } r_i > r_B \\
    0 & \text{for } r_i \leq r_B
  \end{cases}
\]
```

which contributes the following force:

```
\[ F_i = \begin{cases}
    -B (r_i - r_B) x_i & \text{for } r_i > r_B \\
    0 & \text{for } r_i \leq r_B
  \end{cases}
\]
```

To prevent the argon atoms in the gas phase from running off to infinity, we enclose the system in a "balloon" by adding an additional term to the potential energy:

$$Vi = \begin{cases} \frac{1}{B} \times (r_i - r_B)^2 & \text{for } r_i > r_B \\ 0 & \text{for } r_i \leq r_B \end{cases}$$

which contributes the following term to the force:

$$Fi = \begin{cases} -B \times (r_i - r_B) \times x_i & \text{for } r_i > r_B \\ 0 & \text{for } r_i \leq r_B \end{cases}$$

SIMULATION UNITS

In our simulations, we utilize a convenient system of units for atomistic simulations, namely nanometers (nm) for length, picoseconds (ps) for time, and atomic mass units (a.u.m.) for mass.

This system allows for straightforward calculations and meaningful results. The natural unit of energy in this system is kilojoules per mole (kJ/mol).

In this system, the numerical value of the Boltzmann constant (k) is 8.31 × 10^-3 when expressed in atomic units of energy.

To establish the relationships and conversions between these units, let's consider the following conversions:

- 103 kilograms (kg) is equal to 10^3 grams (g), which corresponds to 10^18 square nanometers (nm^2).

- The product of grams (g) and square nanometers (nm^2) results in 10^2 kilojoules (kJ) and 10^24 square picoseconds (ps^2).

- One mole is equivalent to the atomic mass (ajm) multiplied by the length in nanometers (nm) and the energy in atomic units (ajE).

The Boltzmann constant (kB) has a value of 8.31 J/(mol * K) or, equivalently, 8.31 × 10^-3 kJ/(mol * K).

$$\frac{kJ}{mol} = \frac{10^3 kg \frac{m^2}{s^2}}{mol} = \frac{10^3 \cdot 10^3 g \frac{10^{18} nm^2}{10^{24} ps^2}}{mol} = \frac{\frac{g}{mol} nm^2}{ps^2} = \frac{ajm \cdot nm^2}{ps^2} = ajE$$

$$k_B = \frac{R}{N_A} = \frac{8.31 \frac{J}{K}}{mol} = 8.31 * 10^{-3} \frac{kJ}{mol \cdot K} = 8.31 \cdot 10^{-3} \frac{ajE}{K}$$

Lastly, the atomic mass of argon is 39.95 atomic mass units (ajm).

By employing this consistent system of units, we can perform calculations and obtain meaningful results.

PART I - STABILITY ANALYSIS OF THE ALGORITHM.

A necessary condition for algorithm stability is the stability of the total energy. If the energy is not conserved, the system would either cool down to absolute zero or heat up to unphysical temperatures. In either case, it would not be consistent with reality (the law of energy conservation applies).

The absolute deviation of the final total energy from the initial energy, as well as the standard deviation of the total, kinetic, and potential energies, should be determined for the previously mentioned time step values.

Note: Due to starting from a position close to the minimum and the phenomenon of energy equipartition in the simulation, a decrease in kinetic energy (temperature) should occur in the initial period of the simulation.

Simplified simulation algorithm with constant energy:

```
BEGIN
   Arrange the atoms among the nodes of the grid.
   Give the atoms initial velocities.
   Calculate the forces acting on atoms.
   for (i=0;i<n;i++)
        AlgorithmLeapFrog
   Calculate the mean, variance and standard deviation for total energy,
   kinetic energy, and potential energy.
END
```

Optimization

1. Both algorithmic and compilation option optimizations should be performed.
2. For optimization of the **scalar variant** of the program, the following optimizations should be considered in the following areas:
   a) loop order
   b) data organization
   c) utilization of Newton's third law of motion
   d) neighbor lists.
3. The baseline performance of the program should be measured as the execution time of 10,000 simulation steps for a cutoff radius R = 1.3 nm.
4. Attention should be given to the possibility of using vectorization options (SIMD instructions).

Cutoff radius

The influence of introducing a cutoff radius on the speed and accuracy of calculations should be examined. To do this, correctness tests should be conducted for the following cutoff radius values:

R = 1nm, 1.3nm, 1.5nm, 2nm, no cutoff.

PART II - PORTING THE ALGORITHM TO THE GPU.

Stage 1. Porting the algorithm in the simplest version.

Stage 2. Introducing a cutoff radius of R = 1.3 nm.

Introduce a condition that if the distance between atoms is greater than the cutoff radius, the interaction is not calculated.

PART III - PHASE TRANSITION ANALYSIS IN ARGON.

Conduct a simulation with constant energy to verify the stability of the algorithm. In the second part, simulate the temperature increase from 20 K to 120 K and observe phase transitions.

Energy supply to the system can be achieved by scaling velocities. The total energy change when heating from 20 K to 120 K should be N*2 * 3/2 * k*(120K-20K). The factor 2 accounts for energy equipartition (part of the supplied kinetic energy is converted into an increase in potential energy). Therefore, the total energy change DE should be 3NkDT.

$$E_k = \frac{1}{2} m v^2$$

$$E_k + \Delta E_a = \frac{1}{2} m v'^2$$

$$v' = (1 + \alpha) v$$

$$\Delta E_a = \frac{1}{2} m \left(v'^2 - v^2\right) = \frac{1}{2} m \left(v^2 + 2\alpha v^2 + \alpha^2 v^2 - v^2\right)$$

$$\frac{3\mathrm{k}\Delta\mathrm{T}}{n} \cong m \alpha v^2$$

$$\alpha = \frac{3\mathrm{k}\Delta\mathrm{T}}{n \times m \times v^2}$$

If equal amounts of energy are supplied to each atom during the entire simulation, the energy change per atom per step will be DEa = 3kDT/n, where n is the number of steps.

For simplicity, we will use a single coefficient for each step calculated from the root-mean-square velocity (or current temperature):

$$\alpha = \frac{3\mathrm{k}\Delta\mathrm{T}}{n \times m \times \bar{v}^2} = \frac{3\mathrm{k}\Delta\mathrm{T}m}{n \times m \times 3kT} = \frac{\Delta\mathrm{T}}{\mathrm{nT}}$$

Simplified simulation algorithm with simulated heating:

```
BEGIN
   Arrange the atoms among the nodes of the grid.
   Give the atoms initial velocities.
   Calculate the forces acting on atoms.
      for (i=0;i<n;i++)
      {
           Propagate the LeapFrog algorithm
           Scale the speeds
           Complete the appropriate values for sums, counters, etc.
           Save the potential, kinetic and total energy values to a file.
           Save the expected and actual temperature values to a file.
           If appropriate, save the atom coordinates to a file.
       }
END
```

A simulation of 20,000 steps should be conducted with a integration time step of 0.01 ps. Then, using any suitable tools (gnuplot, ggplot, R, Python Excel), a graphical analysis of the results should be performed.

Note - for ambitious individuals, try to identify phase transitions using a temperature plot. Other parameters describing the system that should change during phase transitions can also be investigated.

FUNCTIONAL REQUIREMENTS

The program configuration can be done in header files (during compilation)

The configuration sets the following parameters:
- Level of diagnostic log output (DEBUG) with at least two values.
- Number of simulation steps in the "thermalization" phase - a phase with constant energy (Nterm).
- Optionally, the number of simulation steps in the heating phase (Ngrz).
- Optionally, the number of simulation steps in the cooling phase (Nch).
- Radius of the balloon containing the system (a negative value indicates no balloon is used).
- Energy output frequency (ENERGY), a value of zero means no output.
- Coordinate output frequency (TRAJECTORY), a value of zero means no output.

Cutoff radius for interactions, if the value is zero, interactions are not truncated. (Note - for performance reasons, it is better to implement two separate functions for calculating interactions - one with cut-off and another without cut-off).

Diagnostic output is written to the standard output file “argon.out”, indicating what is happening in the program. The output should have varying levels of precision depending on the DEBUG flag setting.

Energy is written to the file “argon_energy.csv” in each step, broken down into:
Total energy
Potential energy
Potential energy - repulsive term (r^-12)
Potential energy - attractive term (r^-6)
Potential energy - term associated with the balloon
Kinetic energy

System temperature:
Coordinates of atoms are written to the file “argon_trajectory.xyz” every N steps (in plain text format).
The goal is to perform simulations with variable temperature (heating and cooling by scaling velocities).

Validity condition of the program:
Stability of the total energy of the system for 10,000 simulation steps using a time step of t = 0.001 ps.

The stability of the algorithm should also be examined for the following values of the time step: t = 0.002; 0.005, 0.010, 0.020, 0.050 ps.

Tests are conducted at an initial temperature of 70K.