P. Koumoutsakos, P. Hadjidoukas
ETH Zentrum, HG D1.1/G3
CH-8092 Zürich

Set 9 - CUDA and N-body problem

Issued: 30 April 2018
Hand in: 14 May 2018

Question 1: N-body

In this exercise you will have to extend the lecture materials to create a simple N-body solver. The program should compute the movement of N bodies in a periodic domain interacting with Lennard-Jones force.

The system consists of N particles with coordinates $\boldsymbol{r}_i$ in a periodic domain $\Omega = [0, l] \times [0, l] \times [0, l]$. The Lennard-Jones (LJ) potential with parameters ε, σ is defined as follows:

$$U^{LJ}(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{1}$$

a) Compute the force exerted on a particle with coordinate $\boldsymbol{r}_1$ due to the presence of another particle at $\boldsymbol{r}_2$. Assume the energy of the system follows LJ law such that $U(\boldsymbol{r}_1, \boldsymbol{r}_2) = U^{LJ}(|\boldsymbol{r}_1 - \boldsymbol{r}_2|)$.
Hint: $\boldsymbol{F} = -\nabla U$

b) Starting with the provided code, write the functor for LJ forces that can be used in one of the kernels computing forces.

Periodicity of the domain means that a particle i interacts with all the periodic copies of a particle j with coordinates $\boldsymbol{r}_j + \boldsymbol{a}_x l + \boldsymbol{a}_y m + \boldsymbol{a}_z n$, where $\boldsymbol{a}_x = (l, 0, 0)$, $\boldsymbol{a}_y = (0, l, 0)$, $\boldsymbol{a}_z = (0, 0, l)$ and l, m, n are whole numbers defining one of the infinitely many periodic copies of Ω. So the force on particle i due to the presence of particle j is computed as follows:

$$F_{ij} = \sum_{l,m,n \in \mathbb{Z}} F(\boldsymbol{r}_i - \boldsymbol{r}_j - \boldsymbol{a}_x l - \boldsymbol{a}_y m - \boldsymbol{a}_z n) \tag{2}$$

In general, computing all the interactions with the periodic images is a very difficult task, but for the given problem we can make the problem much easier. The key observation here is that LJ potential decays very quickly with r, and so does the force due to it. Therefore we can assume that the force is zero for $r > 4\sigma$ and if $l > 8\sigma$ **only one** of the periodic images in the series **??** contributes towards the total force. That image is actually the closest to the i-th particle along all the three axes and can therefore be very easily determined.

c) Extend one of the three kernels provided in class (any) to compute the forces in the periodic system.

The evolution of the particle system in time follows the second Newtons's law $\ddot{\boldsymbol{r}} = \boldsymbol{F}/m$, where the mass of every particles in our system is assumed to be 1.

We will use a well-known Velocity-Verlet integration scheme to update the positions of the particles with discretely changing time:

$$\begin{aligned} \boldsymbol{r}^{n+1} &= \boldsymbol{r}^n + \boldsymbol{v}^n \Delta t + \frac{1}{2}\boldsymbol{a}^n \Delta t^2 \\ \boldsymbol{v}^{n+1} &= \boldsymbol{v}^n + \frac{1}{2}\left(\boldsymbol{a}^n + \boldsymbol{a}^{n+1}\right)\Delta t \end{aligned} \tag{3}$$

where $\boldsymbol{v}$ is the particle velocity, $\boldsymbol{a}$ is its acceleration and superscripts mean timestep index: $\boldsymbol{r}^n = \boldsymbol{r}(n\,\Delta t)$

d) Write two kernels implementing two equations of **??**.

e) Write a function that will initialize the particles on a regular 3D lattice making sure that periodic copies of the particles don't overlap. The initial velocities and forces of the particles should be set to zero. Initialization should be done on the CPU before the data is transferred on the GPU.

Now you are ready to put the parts together and to setup a complete N-body simulation.

f) Write a program to perform an N-body simulation. Employ the following outline:

    function $\text{NBODY}(N,\ l,\ \Delta t,\ T)$
        $\boldsymbol{r}, \boldsymbol{u}, \boldsymbol{f} \leftarrow \text{INITIALIZE}(N,\ l)$
        $t \leftarrow 0$
        while $t < T$ **do**
            $\boldsymbol{r} \leftarrow \text{INTEGRATEVV\_STEP1}(N,\ l,\ \boldsymbol{r},\ \boldsymbol{v},\ \Delta t)$
            $\boldsymbol{f}^* \leftarrow \text{COMPUTEFORCES}(N,\ l,\ \boldsymbol{r})$
            $\boldsymbol{v} \leftarrow \text{INTEGRATEVV\_STEP2}(N,\ \boldsymbol{v},\ \boldsymbol{f},\ \boldsymbol{f}^*,\ \Delta t)$
            $\boldsymbol{f} \leftarrow \boldsymbol{f}^*$
            $t \leftarrow t + \Delta t$
        end while
    end function

Use the following parameters: $\varepsilon = 0.1$, $\sigma = 0.5$, $\delta t = 10^{-4}$, $T = 1$ and:

- $l = 10,\ N = 10000,$
- $l = 20,\ N = 50000.$