

HW5: N-body Simulation (CUDA)

Due: Mon, 2020/6/1 23:59

- [HW5: N-body Simulation \(CUDA\)](#)
- [Introduction](#)
- [Problem Description](#)
 - [Idea](#)
 - [The Crisis](#)
 - [Counter-Attack](#)
 - [Questions](#)
- [Formulas & Parameters](#)
 - [Gravity simulation](#)
 - [Parameters](#)
- [I/O](#)
 - [Compilation](#)
 - [Command Line Format](#)
 - [Input Format](#)
 - [Output Format](#)
- [Report](#)
- [Submission](#)
- [Grading](#)
- [Appendix](#)
 - [Resources](#)
 - [Judge](#)
 - [Judge Verdict Table](#)

Introduction

- Simulation of a dynamical system of particles
- Under the influence of gravity
- The dataset is an imitation of our Milky Way galaxy

In this homework, you are asked to use CUDA with 2 GPUs to perform the simulation.

Problem Description

This part is the same as the slides. You can skip to [Formula & Parameters](#).

Idea

- N-body is used to predict the motion of celestial bodies. We break continuous time into discrete time steps. The time difference of each time step is given by Δt .

- At each time step:
 - Calculate the accelerations applied to each body.
 - Update the velocities using the accelerations of each body.
 - Update the positions using the velocities of each body.

The Crisis

In the future, human beings have developed a technology to create artificial gravity. Space travel is as normal as driving a car. One day, Planetary Defense Agency discovered that terrorists intend to use this technology and disguise gravity-generation devices into seemingly harmless civilian spacecrafts. They want to use these gravity devices to induce multiple asteroids to hit the colonized planet with high population. If their plan works out, the planet will face massive destruction. Your mission is to determine whether the terrorist's plan will work and how to prevent it from attack.

Gravity devices (type=device) have a special property: their masses will fluctuate. See below for detailed formula.

Counter-Attack

Planetary Defense Agency has to act quickly because the asteroids are so fast that it's infeasible to deflect their direction once the asteroids get near to the planet. The best solution they can do is to destroy such gravity devices with space missiles in order to eliminate the gravitational pull.

The missile carries some explosives and fuels just enough to reach the targeted gravity device. Launching the missile has a cost positively correlated to the time that the missile will travel.

For the ease of simulation, we will assume:

- The missile is sent from the planet
- The missile can guide itself to hit the gravity device with minimal distance, i.e., it travels in a straight line at a constant speed.
- The missile has zero mass and is unaffected by gravity
- A gravity device's mass becomes zero after it is destroyed
- The missile reaches the device when the travel distance is greater than the distance between the planet and the targeted device

Questions

Given an input, you are asked to simulate a given amount of time steps and output the answers to the following questions:

1. If there were no gravity devices, what is the minimal distance between the planet and the asteroid?
2. At what time step will the asteroid hit the planet?
3. Can the collision be prevented if a missile is launched to destroy one of the gravity devices? If so, determine the gravity device to destroy which saves the planet and with the lowest cost.

Formulas & Parameters

Gravity simulation

1. Calculate the force vector on body i caused by body j :

$$\begin{aligned}\mathbf{f}_{ij} &= \frac{Gm_i m_j}{\|\mathbf{q}_j - \mathbf{q}_i\|^2} \cdot \frac{(\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_j - \mathbf{q}_i\|} \\ &= \frac{Gm_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_j - \mathbf{q}_i\|^3}\end{aligned}$$

2. Calculate the total force vector on body i caused by all of the other $(N - 1)$ bodies:

$$\begin{aligned}\mathbf{F}_i &= \sum_{j \neq i} \mathbf{f}_{ij} \\ &= \sum_{j \neq i} \frac{Gm_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_j - \mathbf{q}_i\|^3}\end{aligned}$$

3. To avoid the gravitational force growing indefinitely in situations when two bodies come too close, we add a softening factor ε :

$$\mathbf{F}_i \approx \sum_{j \neq i} \frac{Gm_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{(\|\mathbf{q}_j - \mathbf{q}_i\|^2 + \varepsilon^2)^{\frac{3}{2}}}$$

4. We need acceleration to update positions and velocities ($\mathbf{F}_i = m_i \mathbf{a}_i$)

$$\mathbf{a}_i \approx \sum_{j \neq i} \frac{Gm_j (\mathbf{q}_j - \mathbf{q}_i)}{(\|\mathbf{q}_j - \mathbf{q}_i\|^2 + \varepsilon^2)^{\frac{3}{2}}}$$

5. Update velocities by: $\mathbf{v}_i(t) = \mathbf{v}_i(t - \Delta t) + \mathbf{a}_i(t) \cdot \Delta t$

6. Update positions by: $\mathbf{q}_i(t) = \mathbf{q}_i(t - \Delta t) + \mathbf{v}_i(t) \cdot \Delta t$

Parameters

- Simulation time steps: 200000
- Time difference between each time step (s): $\Delta t = 60$
- Softening factor (m): $\varepsilon = 10^{-3}$
- Gravity constant ($\text{m}^3 \text{kg}^{-1} \text{s}^{-2}$): $G = 6.674 \times 10^{-11}$
- Gravity devices' mass (kg): $m(t) = m(0) + 0.5 \cdot m(0) \cdot \left| \sin \frac{t}{6000} \right|$
- The asteroid hits the planet if their distance at a time step is below (m): 10^7
- Missile traveling speed (m/s): 10^6
- Missile cost: $10^5 + 10^3 t$, where t is the time the missile reach the asteroid.

All the units given above are SI units, so you don't need to do conversion. (If you find otherwise, it's a bug! Please let the TAs know)

I/O

Compilation

We will compile your program with a command equivalent to:

```
ninja hw5
```

Command Line Format

Your program will be tested with a command equivalent to:

```
srun -ppp --gres=gpu:2 ./hw5 input-file output-file
```

Input Format

The input is a text file looking like this:

```
N planet-id asteroid-id
qx0 qy0 qz0 vx0 vy0 vz0 m0 type0
qx1 qy1 qz1 vx1 vy1 vz1 m1 type1
qx2 qy2 qz2 vx2 vy2 vz2 m2 type2
...
```

The first line input file contains three integers, which are:

1. N , the number of celestial bodies
2. ID_{planet} , the (0-indexed) ID of the planet
3. ID_{asteroid} , the ID of the asteroid

Then there are N lines following, each line contains 8 values:

$$\mathbf{q}_{xi}(0), \mathbf{q}_{yi}(0), \mathbf{q}_{zi}(0), \mathbf{v}_{xi}(0), \mathbf{v}_{yi}(0), \mathbf{v}_{zi}(0), m_i(0), \text{type}_i$$

1. $\mathbf{q}_{xi}(0), \mathbf{q}_{yi}(0), \mathbf{q}_{zi}(0)$: the initial position of the i -th body, double
2. $\mathbf{v}_{xi}(0), \mathbf{v}_{yi}(0), \mathbf{v}_{zi}(0)$: the initial velocity of the i -th body, double
3. $m_i(0)$: the initial mass of the body, double
4. type_i : the type of the body, string

Output Format

The output is a text file looking like this:

```
min-dist
hit-time-step
gravity-device-id missile-cost
```

- The first line is the minimal distance between the asteroid and the planet if there were no gravity devices. This value is considered correct if the relative error is within 10^{-8} .
- The second line is the time step that the asteroid hit the planet. If the asteroid did not hit the planet, output `-2`. This value is considered correct if the absolute error is within ± 1 .
- The third line contains two numbers. `gravity-device-id` is the ID of the gravity device that we want to destroy, `missile-cost` is the cost of the missile. If there is no need to destroy the gravity devices, output `-1 0`. If destroying one gravity device couldn't prevent the asteroid hitting the planet, output `-1 0` as well. `missile-cost` is considered correct if the relative error is within 10^{-8} .

Report

Answer the following questions, in either English or Traditional Chinese.

1. What is your parallelism strategy?
2. If there are 4 GPUs instead of 2, what would you do to maximize the performance?
3. If there are 5 gravity devices, is it necessary to simulate 5 n-body simulations independently for this problem?
4. (Optional) Any suggestions or feedback for the homework are welcome.

Submission

Upload these files to iLMS:

- `hw5.cu` – the source code of your implementation.
- `build.ninja` – optional. Submit this file if you want to change the build command. If you didn't submit this file, `/home/ipc20/ta/hw5/build.ninja` will be used.
- `report.pdf` – your report.

Please follow the naming listed above carefully. Failing to adhere to the names above will result to points deduction. Here are a few bad examples: `hw5.CU`, `HW5.cu`, `hw5.pdf`, `report.docx`, `report.pages`.

Grading

1. (30%) Correctness. 10% for each question.
2. (50%) Performance. Based on the total time you solve all the test cases.
3. (20%) Report.

Appendix

Please note that this spec, the sample test cases and programs might contain bugs. If you spotted one and are unsure about it, please ask on iLMS [討論區](#)!

Resources

Refer to `/home/ipc20/ta/hw5/` on hades for sample test cases, source codes and tools.

Judge

The `hw5-judge` command can be used to automatically judge your code against all sample test cases, it also submits your execution time to the scoreboard so you can compare your performance with others.

The scoreboard is [here](#).

To use it, run `hw5-judge` in the directory that contains your code `hw5.cu`. It will automatically search for `build.ninja` and use it to compile your code, or fallback to the TA provided `/home/ipc20/ta/hw5/build.ninja` otherwise. If code compilation is successful, it will then run all the sample test cases, show you the results as well as update the scoreboard.

Note: `hw5-judge` and the scoreboard has nothing to do with grading. Only the code submitted to iLMS is considered for grading purposes.

Type `hw5-judge --help` to see a list of supported options.

Judge Verdict Table

Verdict	Explanation
internal error	there is a bug in the judge
time limited exceeded+	execution time > time limit + 10 seconds
time limited exceeded	execution time > time limit
runtime error	your program didn't return 0 or is terminated by a signal
no output	your program did not produce an output file
wrong answer	your output is incorrect
accepted	you passed the test case