Durham University
School of Engineering and Computing Sciences

Assignment: Space Boddies

Computing Methodologies III – Numerical Algorithms

This assignment is to be completed and handed in on 25 November 2015 via DUO. For an exact deadline (incl. a last submission time), please consult DUO.

Assessment outline
You are to implement a very simple multibody simulation (with stars and rockets/satellites) in two steps and to study numerical properties of these. Marks will be awarded for the implementation of all steps, for the quality of the numerical experiments, and the discussion of the implementation as well as the results. A code template/starting point for the assignment is given in C/C++ on DUO. This code is to be extended. Students are strongly encouraged to augment their report with videos illustrating the code behaviour/numerical results, to upload these videos to their University webspace, YouTube or any other platform, and to add links to these videos to their report.

Step 1
The code fragment on DUO studies three particles in space: two fixed bodies (planets, moons, e.g.) and a smaller body (satellite or asteroid, e.g.) that moves between these two bodies. In the first step, this code is to be extended to handle multiple bodies:

- In reality, the planets/moons/..., i.e. every object in space, moves as well, i.e. is subject of attraction from all other bodies. Extend the code such that all objects move.
- Extend the code such that an arbitrary number of space objects can be simulated (rather than only three). Conduct experiments with at least 1.000 objects. All of them have to move and interact with each other.
- Add a new code fragment that fuses two bodies into one body if they collide with each other. The new body shall have a new mass that is equal to the mass of the original bodies and an averaged velocity (derive a formula that makes physically sense to you and quickly comment in source code).

Conduct some experiments with different choices of N and start positions/start velocities. Construct initial values (initial velocities and positions, e.g.) that you find interesting. Write a brief summary of your observations (at most two pages incl. screenshots/diagrams). In particular, please address:

- How does the runtime of the simulation depend on N?
- How does the performance of your code change in time?

- Which choices of N and initial conditions yield interesting results in your opinion?
- How does the solution change if you alter the time step size (start with N=3 bodies where only one moves, then let all bodies move and increase N)?

Step 2
Pick one out of the following three topics:

1. *Variant 1: Algorithmic optimisation/local time stepping*. The goal in this work package is to reduce the runtime. Start from the observation that particles with a big mass move very slowly. For those, big time steps yield satisfying accuracy (the derivative does not change quickly). Light particles in contrast have complicated trajectories and thus have to be updated with smaller time step sizes. Derive an algorithm optimisation from these observations, describe it in your report, implement it, measure the performance improvements and document your results.
2. *Variant 2: Performance engineering*. The goal in this package is to improve the code's performance. Apply the techniques you have studied in Parallel Programming or related courses to speed up the code on a multicore/parallel computer. This variant is to be followed only if you have already experience in OpenMP, MPI or graphics cards programming. They are not the subject of the course. Document the performance engineering you decided to follow, realise it, measure the performance improvements and document your results.
3. *Variant 3: Algorithmic optimisation/cut-off radii.* The goal in this work package is to reduce the runtime. Start from the observation that particles with a big mass have a severe impact on all other bodies in space while light particles affect only nearby particles. It thus does not make sense to compare these with all other particles. Derive an algorithm tuning from these observations, describe it in your report, implement it, measure the performance improvements and document your results.

The documentation of step 2 shall not exceed two pages (incl. pictures).

Feedback sheet

Criterion	Mark	Comments
Step 1 is implemented.	(/ 20)	
Brief description of step 1 is precise, clear and addresses all questions.	(/ 20)	
Step 2 is implemented.	(/ 20)	
Brief description of step 2 is precise, clear and addresses all questions.	(/ 20)	
The performance improvement of step 2 is significant.	(/ 10)	
Quality of writing and quality of presentation of simulation outputs (graphs, snapshots, videos).	(/ 10)	

Total: (/ 100)