Computer Graphics: Geometry and Simulation
Coursework 1: Simulation

Oliver Jones (s2153980)

This report makes references to videos, which can be found in the `videos` directory of the submission. The videos are numbered in the order they are referenced in the report.

The code for sections 1 and 2 can be found in the `section-12` directory of the submission, containing the `scene.h`, `mesh.h` and `constraints.h` files. The code for section 3 can be found in the `section-3` directory, containing the same `mesh.h` and `constraints.h` files, with an updated version of the `scene.h` file. All other files from the original codebase remain unchanged.

1 Rigid Body Simulation

1.1 Friction (or the Lack Thereof)

One the most significant issues with the implementation of free movement and collisions, is the lack of friction. This can often result in objects sliding around the scene, and never coming to a stop, as show in `video-1`. The simulation depicted in the video uses a coefficient of restitution of `0.5`, with the `mixed-scene.txt` scene file, and the `no-constraints.txt` constraint file.

1.2 Too Perfect of a World

The system lacks the randomness of the real world, meaning objects move predictably, and can often come to rest in unrealistic positions, as shown in figure 1.

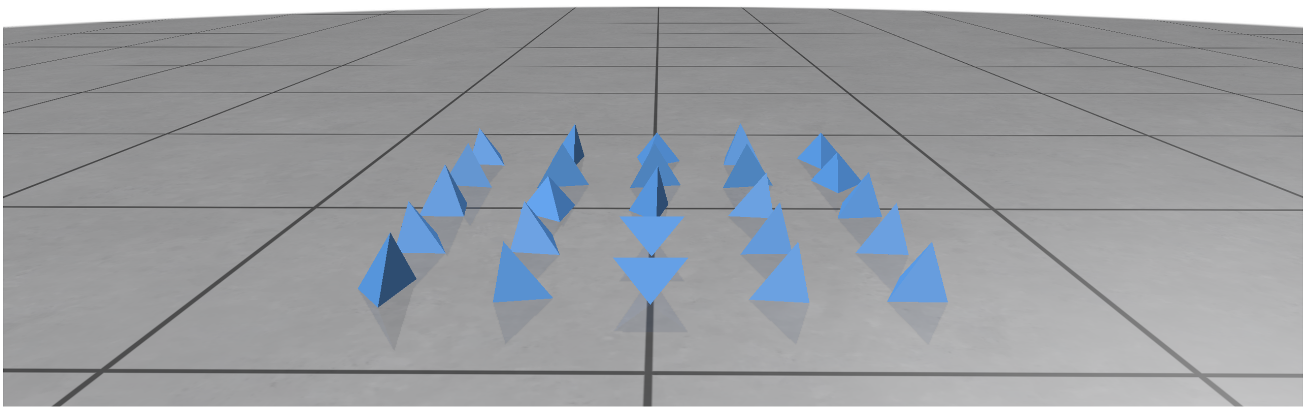

Figure 1: The final resting state of the `tet-scene.txt` scene file, with the `no-constraints.txt` constraint file, and a coefficient of restitution of `0.5`.

Two of the tetrahedra in the image (centre of the front two rows) have come to rest balancing along one of their edges. In a more realistic environment, these objects would have fallen over due to imperfections in the environment, such as the surface they are resting on.

The lack of randomness in the simulation can also be seen in `video-2`, depicting the `tet-scene.txt` scene file, with the `no-constraints.txt` constraint file, and a coefficient of restitution of `1.0`. The video shows the various tetrahedra falling and bouncing in much the same way as one another, consistently bouncing off the same spots, at the same angles. As a result, the objects never collide with one another, as they would in a more realistic environment, as they would bounce off at different angles.

2 Working with Distance Constraints

2.1 Performance Issues

The implementation of distance constraints has a significant impact on the performance of the simulation. This impact is noticeable when the program is not running in release mode, as depicted in `video-3` and `video-4`. The videos show the `two_chain-scene.txt` scene file, with a coefficient of restitution of `0.5`. The `no-constraints.txt` constraints file was used for `video-3`, while the `two_chain-constraints.txt` constrains file was used for `video-4`. Looking in the top left of the videos, you can see a clear drop in performance when the distance constraints are enabled, going from around 30 frames per second, to around 20 frames per second.

3 Scaleable and Efficient Rigid-body Simulation

3.1 More Efficient Constraint Scheduling

As demonstrated in `video-4`, the implementation of distance constraints resulted in a roughly 10 frames per second drop in performance. To help mitigate this, the implementation of the constraints was updated to only check the constraints that are relevant to the current object, using a dependency graph to keep track of the relevant constraints. This change can be seen in `video-5`, which shows the same scene as `video-4`, but with the updated implementation, resulting in a performance increase of around 10 frames per second, back to around 30 frames per second seen in `video-3`, without the constraints.

3.2 Broad-Phase Space Subdivision

Another optimisation that was implemented was broad-phase space subdivision. This optimisation splits the space into a grid, and only checks for collisions between objects that are close enough to one another, resulting in a marginal performance increase.

3.3 Overall Performance Improvements

The combination of the two optimisations resulted in a noticeable performance increase, as shown in figure 2, which shows the frames rates of the simulation with and without the optimisations over a 15-second period.

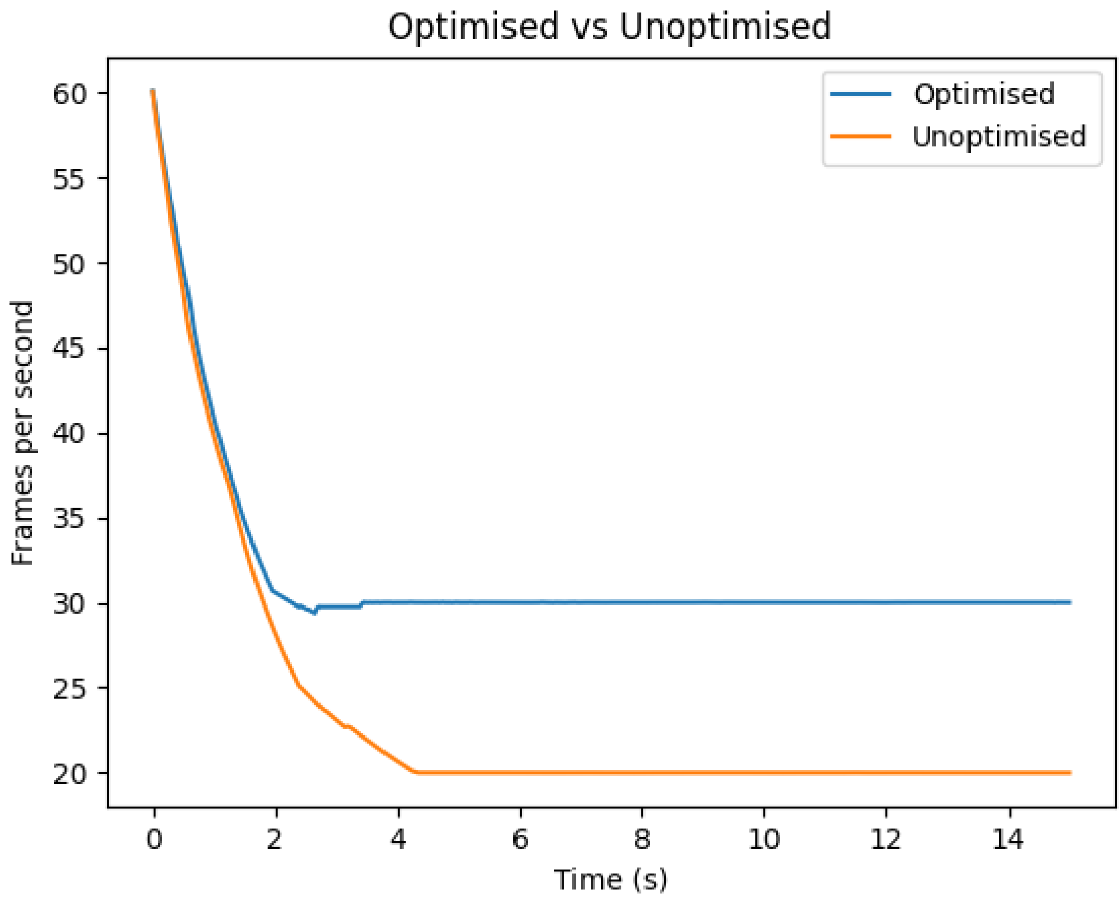

Figure 2: The frame rates of the simulation with and without the optimisations over a 15-second period.