Clustering bodies using the Barnes-Hut algorithm accelerating simulations

In a space with n-bodies, there are $n-1$ forces acting on each body. When simulating all the bodies, $n \cdot (n-1)$ forces need to be calculated for estimating the new position of the individual bodies. With a big enough amount of bodies, this gets problematic. Let's take a real galaxy with $2 \cdot 10^8$ Stars. The total amount of forces that need to be calculated are about $4 \cdot 10^{16}$.

This can be reduced by utilizing the Barnes-Hut algorithm clustering the bodies.

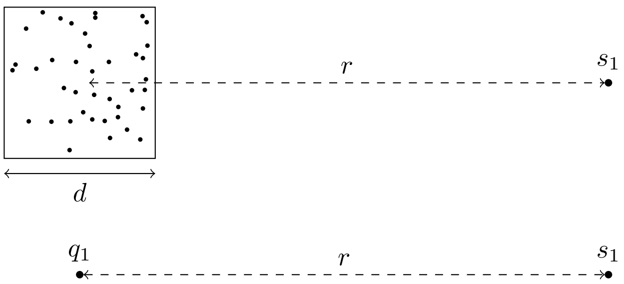

Figure 1: A cluster of stars that is far enough away from a single star can be summed up as a single point in space.

$$\theta = \frac{d}{r} \tag{1}$$

The above equation describes how to cluster the stars. If a body is far away ($>> r$) from a small cluster ($<< d$), θ get's very small. By defining a theta as a threshold, we can define what clusters we take into effect when calculating the forces acting on a single star.

In order to do so, the space in which the objects are defined needs to be subdivided into cells. Such a subdivision can be seen in Figure 2.

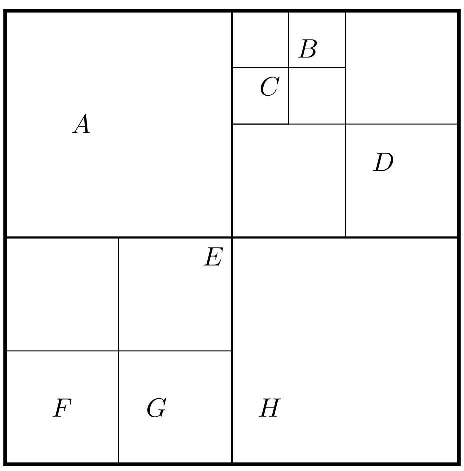

Figure 2: Subdivision of a 2D Space containing some bodies.

When calculating the forces on let's say the object F, not all other objects need to be taken into effect, only the ones that apply to the Barnes-Hut Principle. For the object F, this means that the Objects C and D are not calculated independantly, but as one object (The midpoint of center of gravity is defined as a new object).

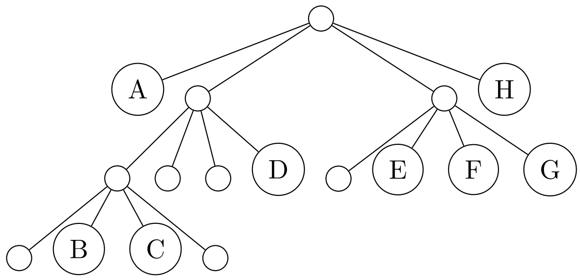

Figure 3: The cells defined in Figure 2 dispayed in the form of a quadtree.

In order to simulate the change of position for all objects in the given space, a tree can be used. The tree in Figure 3 describes the cells from Figure 2 in a form that can be easily programmed. The complete process of simulating works in the following way:

1. Define an empty space.
2. Insert the objects subdividing the space if necessary.
3. Calculate the center of mass and the total mass for all inner nodes in the tree.
4. For calculating the force acting on a star, walk through the tree from the root in direction of the leaves, using the Barnes-Hut Algorithm (1) as an end codition.

In the end, when simulating a lot of bodies, the runtime is optimized from $O(n^2)$ to $O(n \cdot \log(n))$. This means that if you've got $2 \cdot 10^8$ bodies and can calculate the forces acting on $1 \cdot 10^6$ bodies bodies per second, the total runtime is reduced from about 1200 Years to 45 minutes (this is just the calculation of the forces, inserting the bodies into the tree takes alot of time!).