Barnes Hut Simulator

Data Structures and Algorithms: Lab

Ojaswi Jain
IIT Bombay

Contents

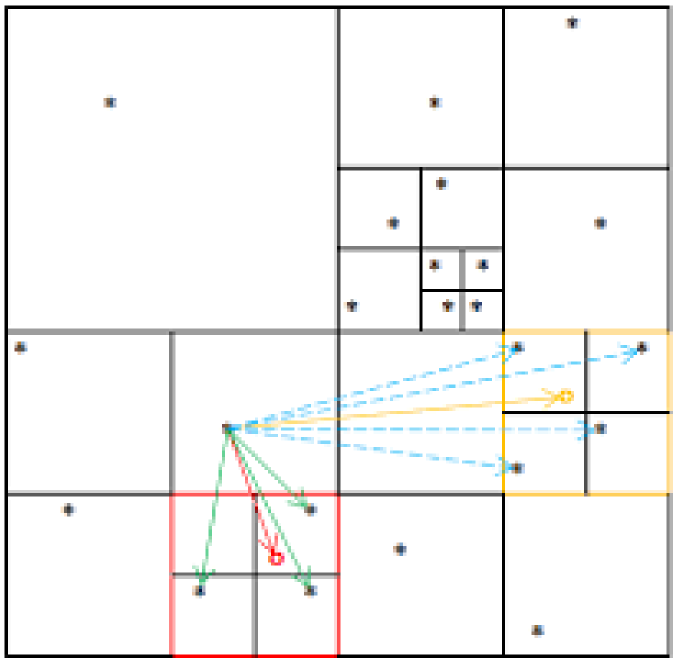
1 The Algorithm

The Barnes-Hut Algorithm describes an effective method for solving n-body problems. It was originally published in 1986 by Josh Barnes and Piet Hut. Instead of directly summing up all forces, it is using a tree based approximation scheme which reduces the computational complexity of the problem from $O(N^2)$ to $O(NlogN)$.

It works by reducing the number of force calculations by grouping particles. The basic idea behind the algorithm is that the force which a particle group exerts on a single particle can be approximated by the force of a pseudo particle located at the groups center of mass.

For instance, the force which the Andromeda galaxy exerts on the milky way can be approximated by a point mass located at the centre of the Andromeda galaxy. There is no need to integrate over all stars in the Andromeda galaxy provided the distance between the two galaxies is large enough.

This approximation is valid as long as the distance from a point group to a particle is large and the radius of the group is small in relation to the distance between the group and the particle.



]

2 Implementation

Lets start with a random distribution of points in a two dimensional domain. The first step of the Barnes-Hut algorithm is sorting the particles into a hierarchical tree structure. This structure is the so called quadtree.

The algorithm starts by subdividing the domain into quadrants. In order to add a particle to the tree its quadrant is determined. If there is another particle in the same quadrant the quadrant is subdivided again. This is repeated until each particle is located in its own quadrant.The creation of the quadtree is a recursive process.

The tree has to be recreated for each step of the simulation. The computational cost of recreating the quadtree depends on the distribution of the particles. The cost of adding a particle to the tree is proportional to its distance from the root node. Distributions with many densely packed particles require more operations because the tree must be subdivided often to place all particles in their own subdivided quadrant.

Up until now the tree only contains the spacial distribution of the particles. The next step deals with calculating the mass distribution of the quadtree. It consists of computing the total mass of all particles contained in the child nodes of each tree node as well as their center of mass. This data must be computed for each tree node. The calculation can be implemented recursively by scanning all child nodes of each tree node.

The last step of the algorithm is computing the forces exerted on the particles.

If $\frac{s}{d}$ lies below a certain threshold the force can be approximated by inserting the quadrant nodes mass and its center of mass in the law of gravity. The child nodes don't have to be summed up separately. A reasonable criterion for this decision is $\frac{s}{d} < 0.5$. If $\frac{s}{d}$ is larger than the threshold the quadrants effect can't be approximated and all of its child nodes have to be tested again. The iteration stops only after all nodes have been tested.

The worst case scenario is having to test all nodes without finding any node that meets the threshold criterion. In such a case the result is similar to summing up all mutual particle forces. The iteration depth can be fine-tuned by adjusting the critical threshold.

3 Graphics

SFML is a simple, fast, cross-platform and object-oriented multimedia API. It provides access to windowing, graphics, audio and network. It is written in C++.

Although the library is archaic and has plenty of bugs and limitations, it does serve the purpose for a small-scale version of the simulator.

I intend to implement the same algorithm using Manim framework developed in Python.