Problem Set 7

Problems to Computational Astrophysics, WS 2013/2014

Prof. Dr. Friedrich Röpke, Prof. Dr. Christian Klingenberg, Sebastian Ohlmann
Offices: Campus Hubland Nord, 31.01.017, 30.02.012, 31.01.003

Hand in until Monday, 16.12.2013, 12.00 pm
Tutorial on Tuesday, 17.12.2013, 10.15 am

1. Tree method (P)

In this problem set, we want to show that the tree method has $\mathcal{O}(N \log N)$ time complexity for a system of N particles.

a) To this end, first write pseudocode for an algorithm for constructing an octree by looping over all particles and inserting them into the appropriate position in the tree.

b) This is the first step of the Barnes-Hut algorithm. What is its time complexity?

c) After constructing the tree, it has to be traversed to compute the mass m_{node} and the center-of-mass position $\boldsymbol{x}_{\text{COM}}$ for each node in it. These quantities are then stored for each node. What is the time complexity of this step? The last step is to traverse the tree once for each particle and to calculate the gravitational force on it. A pseudocode for this is given in Algorithm 1.

Algorithm 1 Calculate force

```
for i = 1 to N do                                   ▷ for each particle traverse tree
    f(i) = Force(i, root)                           ▷ Compute force on it by calling function Force
end for

function FORCE(i, node)                             ▷ Compute force on particle i due to node
    Force = 0
    if n contains one particle then
        Force = F calculated from Eq. (1)
    else
        R = |x_i − x_node|                          ▷ Distance from particle i to the node under consideration
        w = width of box corresponding to node
        if w/R < θ then                             ▷ Criterion for discriminating far from near nodes
            Force = F calculated from Eq. (1)
        else
            for all children c of node do           ▷ Break up node into its children
                Force = Force + Force(i, c)         ▷ Recursive call of function Force
            end for
        end if
    end if
end function
```

Here, the force due to a specific node (or leaf) on particle i the tree is determined via

$$\boldsymbol{F} = G m_i m_{\text{node}} \frac{\boldsymbol{x}_i - \boldsymbol{x}_{\text{node}}}{|\boldsymbol{x}_i - \boldsymbol{x}_{\text{node}}|^3}. \tag{1}$$

d) For simplicity, we chose $\theta > 1$ as the criterion for breaking up nodes. What is the complexity of each call to the force calculation outlined in Algorithm 1?

Hint: Consider the two-dimensional situation of a quadtree for a particle i located in the lower right corner of the domain. Show that the amount of work to calculate the force on i is proportional to the level of the tree i resides in.

Note: For $\theta < 1$ which is usually chosen in practical applications, the argument is more complicated, but the complexity class does not change.

e) Reason that our line of arguments implies $\mathcal{O}(N \log N)$ complexity for the full algorithm.

Now that you have installed SciPy (or had it already before), you may as well try the example code for the tree method called "TreeGrav". Download it and start the GUI as before. Run it and play with the different parameters. Don't miss our fabulous "Tree Demo", for which instructions are given in the Readme.

Exercises marked with (P) have to be presented in the exercise, those marked with (H) have to be handed in. Programs can be sent per e-mail to `sohlmann@astro.uni-wuerzburg.de`.