

Proposal for Astro 585 : Cluster finding algorithm

Moupiya Maji

February 21, 2014

1 The idea

I want to build a algorithm which given a set of position and velocities of particles, can find clusters of particles inside it. I also want to check the boundness of the particles, so that the clusters are physical bound systems.

2 Project summary

A common problem recurring in astronomy and many areas in physics involve detecting clusters in a field of particles. Common examples could be the task of finding galaxy clusters in a universe simulation, finding star clusters in a galaxy simulation, finding planets in a protoplanetary disk etc. There are quite a few cluster finding algorithm like Amiga Halo Finder (AHF), Friends of Friends (FOF) etc. But they are somewhat specific to each area of specialization, such as AHF is particularly useful for a cosmological simulation but not so much for galaxy simulations. FOF has a standard problem of finding the linking length which could be quite arbitrary resulting in arbitrary halos.

I want to build a simple cluster finding algorithm which could be used for any dataset where we have the positions and velocities for the particles. I also want to parallelize the code to make it faster, since I assume it will take a fair amount of time to run for a million or particles.

The inputs of my code will be positions, velocities and masses of the particles. That should be sufficient to calculate to find clusters in the spatial dimensions first and then inspect whether they are gravitationally bound or not. The output of my code should be a list of clusters with the description particle ids belonging to the cluster. I plan to work alone.

3 Initial Implementation Plan

What do you anticipate will likely be the most time consuming portions? For finding the clusters I can use either a optimal length or a overdensity criterion. In the first aproach I find a physically relevant scale, say for star clusters the typical size is l . Now all particles around a particle within radius l will be considered part of the cluster. Repeat this for all particles. In general this will be a $O(N^2)$ algorithm. I can divide the whole region into grids and do it grid by grid. That reduces the computation time by a factor of M where M is the number of grids.

I suspect that the find the closest patticles to select cluster candidates iwll take the most time. Also time consuming is the potential energy computation. If applied naively, this will also be a $O(N^2)$ algorithm.

I predict that run time will vary a great deal with size of the problem. I will vary the number of particles. I can benchmark it with tens , hundreds to millions of particles which is a realistic number for typical simulations.

I am going to use Julia for this problem. Julia has a great structure for parrallelizatiion which will be extremely helpful in implementing the code.

4 Testing Plan

First I will test it with a very simple system of 1 dimesional position of a few (~ 5) particles with obvious clustering of two groups. Once it passes that I will increase the dimensions and number of particles, until it reaches 3d and a million particles. at each step I will do regression test with the previously completed test conditions to make sure that it can reproduce the previous results.

5 Parallelization Plan

I think that parallelization will be very important for running this program . Since there will be typically so many particles, and I will want to use it several times for my project, it would be ideal to paralleize it properly. I want to use rcc cluster formy parallelization, since they have setups with infini bands which will help the communications between different parts of code more efficient. I am thinking I will use MPI or OPENMP for that. I will use my linux laptop/Mac desktop for the initial development and finally run it in lionx clusters.

6 Overall Strength of Proposal

This problem is well suited for parallelization. This will be very useful for the research I am doing now. Also, if I can build it well, I assume I will use it multiple times in my future projects too. Maybe other people can also benefit from my code too since this is a common and important problem with no general simple code. I am hoping to finish this code in a reasonable amount of time. I am excited that I will get to learn a lot of new things along the way which will help me prepare for my research.