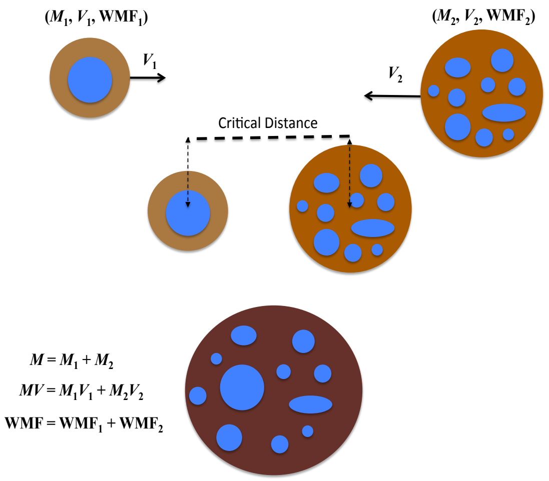
Incorporating Collisions and Mixing into an N-body Integrator

Nader Haghighipour

When modeling the formation and orbital evolution of planetary systems, collisions play important roles. They are the main processes through which planets grow in mass and size, and it is through collisions that material is transferred from one body to another. Unfortunately, due to the unpredictability of the outcome of collisions, and the general complexities of this process, especially the state of a system after a collision, it is not possible to formulate the process of the collision between two bodies. As a result, N-body integrators have been using different approaches to make their internal code understand that two objects are about to collide, and how to proceed after they have collided.

A common approach in handling collisions is to define a critical distance such that when the centers of two bodies are closer than that distance, they are considered collided. In this approach, collisions are considered to be perfectly inelastic meaning that the mass of the final body is equal to the sum of the masses of the two colliding objects, and the final velocity of the resulting body is calculated using the conservation of momentum. Because during a collision, material is transferred from one body to another, a procedure for handling collisions can also be used to determine the transfer and mixing of materials during the formation and evolution of the system. Figure below shows this schematically. The term WMF refers to the water-mass fraction of each body.



Projects

Our goal is to develop a computational code to incorporate collisions and mixing in an N-body integrator. In doing so, we answer the following questions (projects):

1) What would the best way to define the critical distance?

2) From answer to question 1, what would be the best value for the critical distance?

3) What would be the most efficient way to code collisions in an N-body integrator?

4) Defining a collision based on the critical distance as in question 2, what would be the most efficient way to code transfer of material from one body to another?