SPH with sudo MHD

Zahra Forootaninia

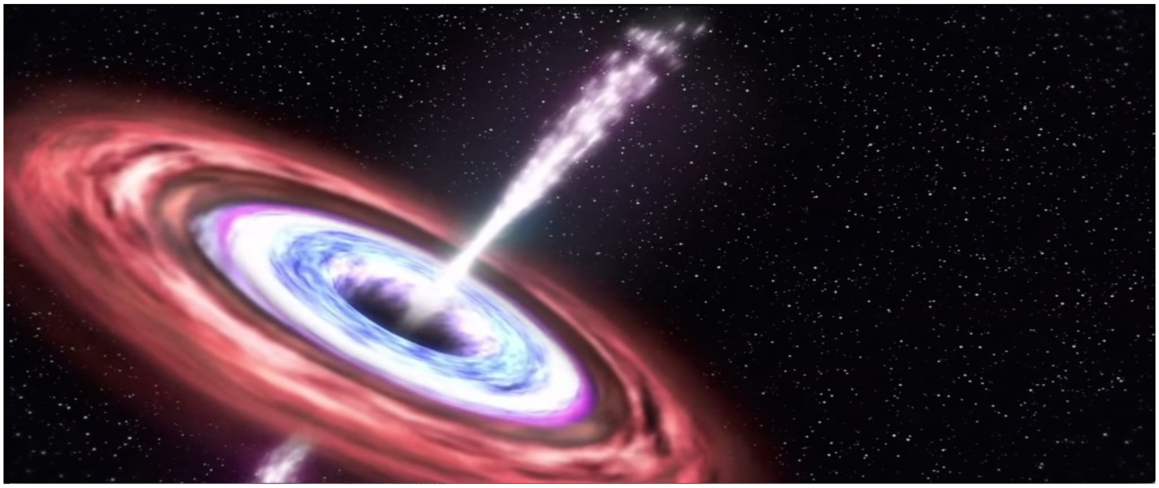

Figure 1: *Accretion disk*

Abstract

Smoothed particle hydrodynamics (SPH) and Magnetohydrodynamics (MHD) are a particle-based method for simulating the behavior of fluids. Each particle have information about the fluid in a small region, such as the velocity, density and electrical charge. During the course of the simulation, these particles interact with each other in a way that models the dynamics of a fluid. In this project, I will tune a simple 3D version of an SPH with sudo MHD for the fluid.

1 Introduction

Smoothed particle hydrodynamics (SPH), first developed by Lucy (1977) and Glingold and Monaghan (1977) for Astrophysics prepossess and now it has been used in so many other fields.
In the SPH method we dividing the fluid into a set of discrete particles with mass m. Each particle in the fluid has a distance from its neighbor known as "smoothing length"(h). Properties of the particle will be smoothed by a kernel function, which in case of SPH is a Kernel functions(W). The common Kernels are Gaussian functions and the cubic spline.The advantage of cubic spline function over a Gaussian is that it is finite. So, the particles at distances that their contribution is effectively zero will not be considered. The cubic spline kernel function is

$$w(q) = \sigma \begin{cases} \frac{1}{2}(2-q)^3 - (1-q)^3 & 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3 & 0 \leq q < 2 \\ 0 & q \geq 2 \end{cases} \tag{1}$$

in which $W(r_{ij}, h) \equiv \frac{1}{h^d} w(q)$, $q \equiv r_{ij}/h$ and $\sigma = [2/3, 10/(7\pi), 1/(\pi)]$. In this case W will be zero at any point larger than $2h$.

We can construct the density based on the distribution of particles.

Using the kernel function, we can calculate the density at any position in space as following:

$$\rho_i = \sum_{j=1}^{N} m_j W(r_{ij}, h) \tag{2}$$

where r_{ij} is the distance between the position i in space and particle j, m_j is the mass of particle j, h_j is the smoothing length of particle j, and the sum is over all other particles in the system (N). The remaining steps of the SPH algorithm can be derived based on the density.

It is necessary to consider Magnetohydrodynamics (MHD) for simulating magnetic properties of fluids for astronomical purposes. However, it can be expensive and complicated. We can animate the simpler version of MHD by assuming the magnetic field in the background, instead of the fluid itself cary the magnetic field. Therefore, each particle has an electric charge of q_i as one of its properties while magnetic field B is in the background.

There are four types of forces: Attractive, repulsive, damping, and external. The first three forces are from the interaction between the particles in the system. Therefore, for every position we have to find all the interactions of the particle with others. In this simulation, I account for pressure, viscosity and self-gravity as sources of internal forces while the magnetic field will enforce an external force on the system as following

$$F_i^{pressure} = -\frac{m_i}{\rho_i} \nabla p_i \tag{3}$$

$$F_i^{viscosity} = m_i v \nabla^2 v_i \tag{4}$$

$$F_i^{self-gravity} = m_i g \tag{5}$$

$$F_i^{magnetic} = q_i v_i \times B \tag{6}$$

$$F_i(t) = F_i^{pressure} + F_i^{viscosity} + F_i^{self-gravity} + F_i^{external} \quad (7)$$

in which $F_i^{external} = F_i^{magnetic}$.

2 Technical description of the algorithm

In this animation, we solve a system of differential equations with the following initial conditions: a collection of particles with equal masses m and interaction radii of h within the disk. Each particle i has a position r_i , velocity v_i, density ρ_i and electric charge of q_i. Magnetic field is in the direction of ϕ in the cylindrical coordinates in the background. Particle i interacts with N_i particles.

The main steps of algorithm are:

1. compute density $\rho_i(t)$
2. compute pressure $p_i(t)$
3. compute forces $F(t)$
4. (optional) boundary condition check
5. update velocity $v_i(t+1)$
6. update position $p_i(t+1)$

Time integration is a fundamental part of any animation algorithm. The leapfrog time integration is usually used in particle simulation algorithms due to its well-known properties that makes it as one of the best choices for SPH simulation. It is explicit, second-order accurate and it conserves energy with short time steps in which the system will remains stable.

In leapfrog time integration algorithm the velocities are updated on half steps and the positions on integer steps as is shown in the following equations.

$$v^{i+1/2} = v^{i-1/2} + a^i \Delta t \quad (8)$$

$$r^{i+1} = r^i + v^{i+1/2} \Delta t \quad (9)$$

3 Goals

The main goal is to implement a 3D SPH with a simple version of MDH in which the magnetic field is in the background. Since particles in the fluid has electric charge, they feel the magnetic force, and start to move in perpendicular direction to the magnetic field (based on Right-hand rule). Therefore, we expect to see the fluid flows in a specific direction which produce a jet stream as can be seen in Figure 1.

References

GOSWAMI, P., AND PAJAROLA, R. 2011. Time adaptive approximate sph. In *Workshop in Virtual Reality Interactions and Physical Simulation*, The Eurographics Association, 19–28.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 154–159.

PRICE, D. J. 2012. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics 231*, 3, 759–794.