The Fluid Approximation

This is a moment-based approach to the Boltzmann equation in which the interaction wavelength is much smaller than the system's size and leads to establish rapidly (and restore rapidly) a thermal equilibrium as described by the Maxwellian distribution function. In its simplest form one neglects viscosity, thermal conduction and shear stresses between particles, which originate mainly from electromagnetic forces *(ideal fluid)*, and considers only *isotropic pressure (perfect fluid)*

One starts from taking velocity moments of the Boltzmann equation.

Zeroth order moment gives continuity equation

First order moment gives momentum equation

Second order moment gives energy equation

Zeroth moment

$$\int d\mathbf{v}\, m\left\{\frac{\partial f}{\partial t}+\mathbf{v}\cdot\frac{\partial f}{\partial \mathbf{x}}+\frac{\mathbf{F}}{m}\cdot\frac{\partial f}{\partial \mathbf{v}}\right\}=\int d\mathbf{v}\, m\left(\frac{\partial f}{\partial t}\right)_c$$

with $\int d\mathbf{v}\, mf = mn = \rho \qquad \int d\mathbf{v}\, m\mathbf{v}f = \rho\mathbf{u}$

"mean" fluid velocity

yields continuity equation for f⟶0 as v⟶0 and conservation of particle number, which sets right-end side to zero

$$\frac{\partial \rho}{\partial t}+\nabla\cdot\rho\mathbf{u}=0$$

First moment from integral of continuity equation multiplied by m**v**

$$\frac{\partial}{\partial t}\rho\mathbf{u}+\nabla\cdot[\rho\mathbf{u}\mathbf{u}+\mathbf{P}]-n\mathbf{F}=\int d\mathbf{v}\, m\mathbf{v}\left(\frac{\partial f}{\partial t}\right)_c$$

with pressure tensor $\mathbf{P}=\int d\mathbf{v}\, mf\,(\mathbf{v}-\mathbf{u})(\mathbf{v}-\mathbf{u})$ that would need second moment

but can be reduced by assuming scalar (isotropic) pressure and an associated equation of state (eg ideal gas)

$$\mathbf{P} \approx p\mathbf{I}$$

$$p = \rho RT = nk_B T.$$

The two above ansatz correspond to a *perfect ideal fluid*

$$\rho \frac{d\mathbf{u}}{dt} = \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + n\mathbf{F} + \int d\mathbf{v}\, m\mathbf{v}\left(\frac{\partial f}{\partial t}\right)_c$$

Assuming a maxwellian for the distribution function (=thermodynamical equilibrium) the above equation reduces to the well known Euler equation for perfect inviscid fluids since the rightmost term cancels out (the Maxwellian does not depend on time!)

Most astrophysical systems that cannot be described by the collisionless or (weakly) collisional approximation can be well treated as ideal perfect fluids.
The approximation can be extended also to charged fluids with magnetic fields for which the magnetohydrodynamical equations can be derived by coupling fluid and Maxwell's equations. *Examples are*:

Stars, accretion discs around stars and black holes, planets, stellar explosions, interstellar gas in galaxies, star forming clouds....and many others!

Now that you should be convinced that a "Boltzmann-type" equation is the general mathematical model applicable to nearly all astrophysical systems we can finally start discussing numerical discretization of the equations and their integration.

For collisionless systems, rather than solving the Vlasov-Poisson system in the "mean field" regime we will forcefully recover the particle nature of the system by discretizing using superparticles. The coarse discretization of the *f* will make sense as long as the relaxation timescale in the discretized representation of the system is still long enough to enforce the collisionless behaviour,
---> *N of superparticles should be large enough*!

N-Body models of collisionless systems

In a gravitational system one solves a simpler, alternative system of ordinary differential equations for the mean potential Φ rather than solving directly the Vlasov-Poission equation. The resulting trajectories in phase space should coarsely sample *f* as long as the relaxation time of the N-Body model is long compare to the duration of the simulation. However a single N-Body experiment with a computer will only represent a noisy representation of *f* because no ensemble averaging is performed (one should run many experiments and then average out the resulting forces to get trajectories in the mean potential)

$$\ddot{\mathbf{x}}_i = -\nabla_i \Phi(\mathbf{r}_i),$$

$$\Phi(\mathbf{r}) = -G \sum_{j=1}^{N} \frac{m_j}{[(\mathbf{r}-\mathbf{r}_j)^2 + \varepsilon^2]^{1/2}}$$

The gravitational softening ε is introduced for both computational efficiency and to enforce numerical robustness (=smoothness) of the model for the collisionless fluid. To avoid correlation between particles

$$\langle v^2 \rangle \gg \frac{Gm}{\varepsilon}$$

In practice the choice is dictated by a compromise between accuracy of the force calculation and speed of computation (*computational efficiency*). Choice will be application dependent.

Note that accuracy should not be pushed too much since there are inevitable *truncation errors* (eg discretization errors and finite timestep choice for time integration, to be discussed) and *round-off errors of computers (=error due to truncation in floating point number representation)*

Among the methods:

(1) Particle mesh (PM)
(2)Fourier transform-based solvers of Poisson
(3) Iterative solvers of Poission on a grid (multigrid)
(4) Tree-based methods (hierarchical multipole methods)
(5) Hybrid methods, such as TreePM or P^3M

Particle mesh (PM) technique

Underlying idea: construct an auxiliary mesh on which forces are computed by computing the potential

Note: the mesh introduces *a second discretization* procedure after that introduced by super-particles.

The technique involves four steps:

1. Construction of a density field ρ from particle distribution.

This effectively smooths out the discrete representation of the collisionless fluid (reduces N as wanted)

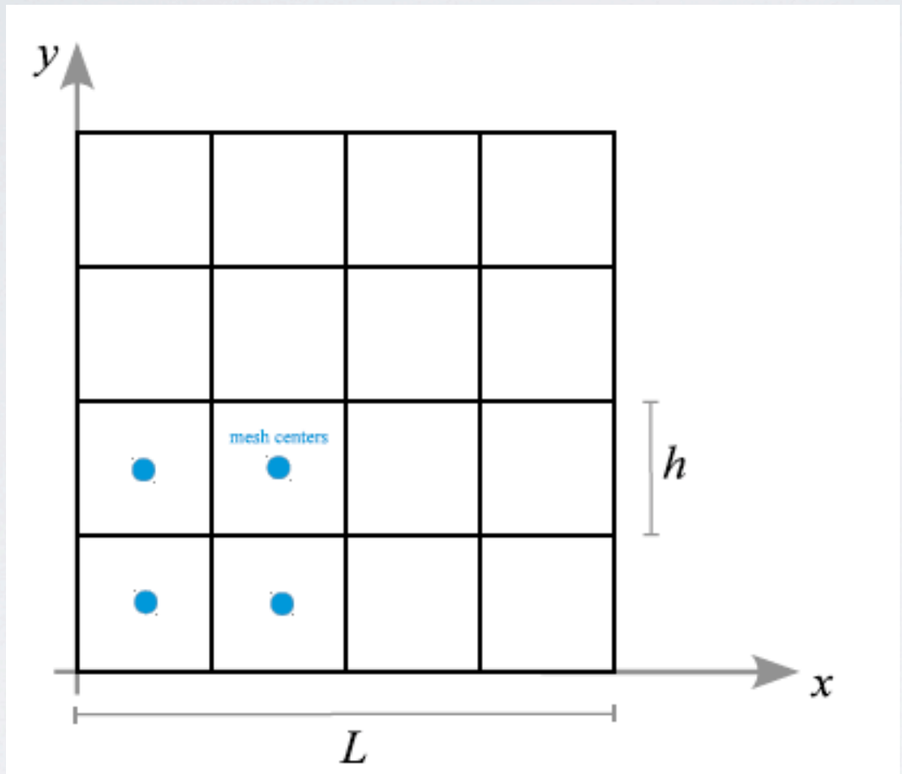
But note this is a smoothing of the super-particle model not of the original collisionless system (eg galaxy) hence it does not eliminate the discretization errors of the particle representation rather can introduce new ones!

2. Computation of the potential on the mesh by solving the Poisson equation with the density field ρ mapped from particle onto the mesh

3. Computation of the force field from the potential (on the mesh)

4. Calculation of the forces at the original particle positions (interpolation step)

Mass assignment procedure



Lay down N_g grid cells that surround particles. For simplicity we assume cubical cells (squares in 2D) with uniform spacing $h=L/N_g$ (*uniform grid*). Labels cells with integer index $\boldsymbol{p}=(p_x,p_y,p_z)$, running from *0 to* N_g, corresponding to cell center position $\boldsymbol{r}_p$,

Now we need to define how to "spread" the mass of particles across cells, which is accomplished by a *shape function* $S(\boldsymbol{x})$ appropriately normalized:

$$\int S(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1.$$

We then need define *the assignment function* $\boldsymbol{W_p(x_i)}$ which expresses the fraction of mass of particle *i* with coordinate $\boldsymbol{x}_i$ that falls into a cell with integer index $\boldsymbol{p}$

$$W_{\mathbf{p}}(\mathbf{x}_i) = \int_{\mathbf{x_p}-\frac{h}{2}}^{\mathbf{x_p}+\frac{h}{2}} S(\mathbf{x}_i - \mathbf{x_p})\,\mathrm{d}\mathbf{x}.$$

Note that integration extends over the cubical cell **p** with size *h*
Then a mathematical step to reformulate for arbitrary integration boundaries using the top-hat Π function:

$$\Pi(\mathbf{x}) = \begin{cases} 1 \text{ for } |\mathbf{x}| \leq \frac{1}{2}, \\ 0 \text{ otherwise,} \end{cases}$$

So that one writes the assignment function *W* as the convolution of **Π** with *S:*

$$W_{\mathbf{p}}(\mathbf{x}_i) = \int \Pi\left(\frac{\mathbf{x} - \mathbf{x}_{\mathbf{p}}}{h}\right) S(\mathbf{x}_i - \mathbf{x}_{\mathbf{p}})\, d\mathbf{x}.$$

The full density in grid cell **p** will be given by the sum of the fractional mass contributions of particles to cell p, which thus depends on the choice of *S* through ***W****:*

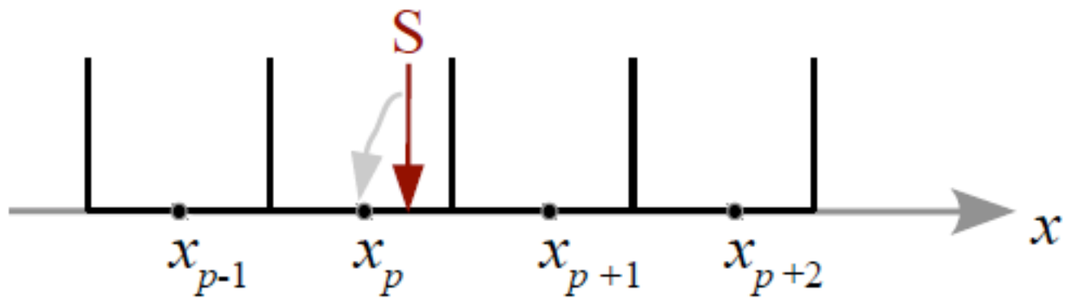
$$\rho_{\mathbf{p}} = \frac{1}{h^3} \sum_{i=1}^{N} m_i W_{\mathbf{p}}(\mathbf{x}_i).$$

The different choice of *S* leads to different mapping between particle distribution and density on the mesh.
The more particle masses are "spread" over more cells via a more complex shape function the smoother is the density function that results, as measured by the continuity of first and second derivatives.

Choices of shape functions *S*

The most trivial choice is $S(\mathbf{x}) = \delta(\mathbf{x})$, which leads *to the nearest grid point mass assignment*, essentially meaning that the entire mass of a particle is assigned only to the cell in which it is embedded. The coordinate of the particle is shifted to that of the center of mass of the cell:

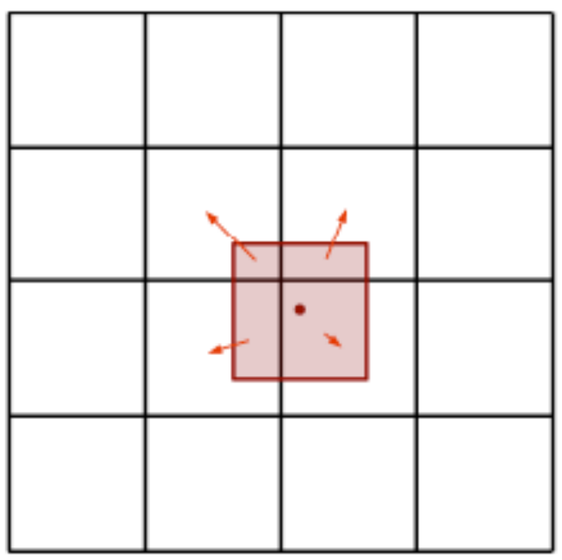
$$W_{\mathbf{p}}(\mathbf{x}_i) = \int \Pi\left(\frac{\mathbf{x}-\mathbf{x}_{\mathbf{p}}}{h}\right) \delta(\mathbf{x}_i - \mathbf{x}_{\mathbf{p}})\,\mathrm{d}\mathbf{x} = \Pi\left(\frac{\mathbf{x}_i - \mathbf{x}_{\mathbf{p}}}{h}\right)$$



This method is easy but noisy as the assigned density changes abruptly as particle cross cells, leading to a noisy, discontinuous force (recall force obtained from potential obtained from density). It is especially bad for clustered mass distribution (common case in astro and cosmology)

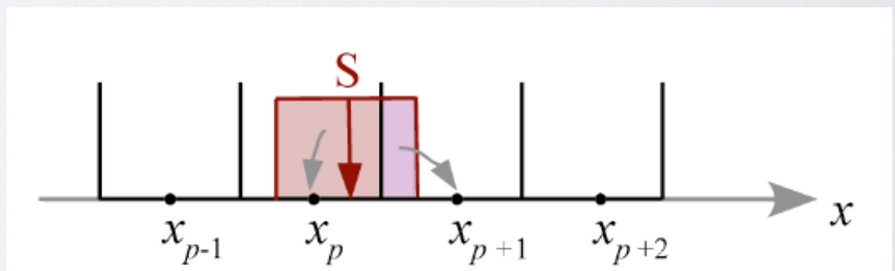
A less noisy mass assignment method (force continous, its first derivative discontinous) is the cloud-in-cell method (CIC).

Underlying idea: embed particles in a cubical cloud (square in 2D), centered on the particle's position, that can overlap with more than one cell. Mass is then redistributed over the cells with non-zero overlap by computing $W_{\mathbf{p}}$, which then yields the "fractional mass weight" associated to each cell, namely what fraction of the particle's mass is attributed to a given cell. $W_{\mathbf{p}}$ computes the weights by computing the fraction of cloud's area that overlaps w/cell **p**



2D CIC
(4 weights
to be computed)

1D CIC
(2 weights
to be computed)



In 3D 8 weights have to computed (mass spread over 8 cells)

Mathematically, one uses following shape function:

$$S(\mathbf{x}) = \frac{1}{h^3}\Pi\left(\frac{\mathbf{x}}{h}\right),$$

resulting in the following mass assignment function:

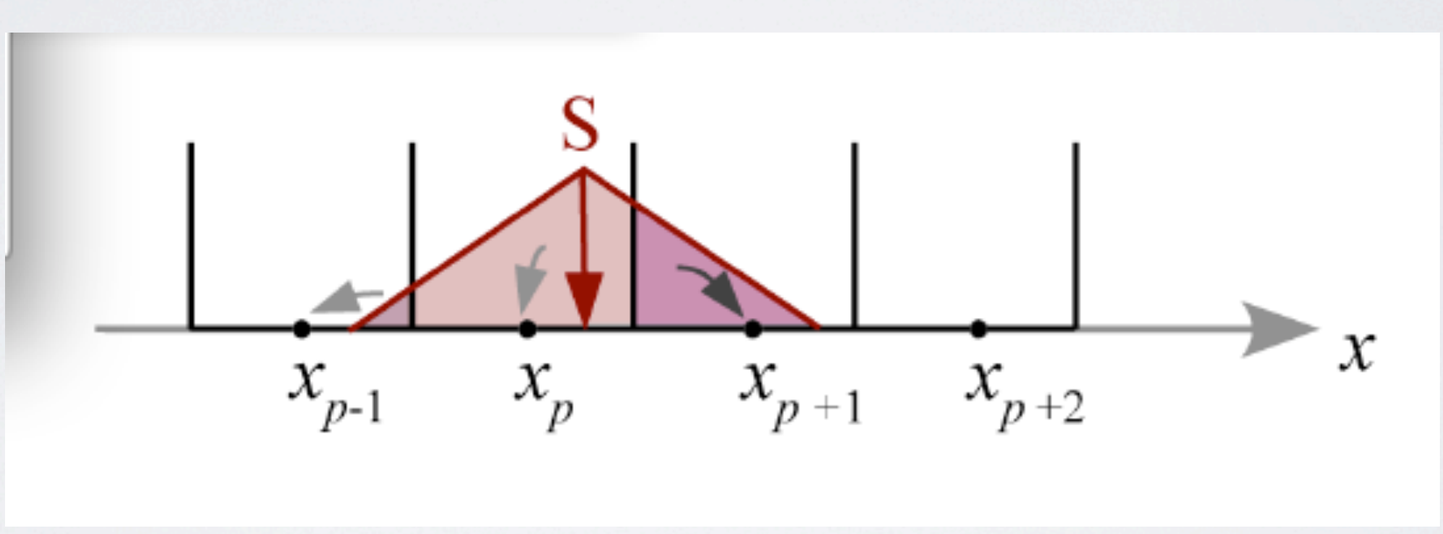
$$W_{\mathbf{p}}(\mathbf{x}_i) = \int \Pi\left(\frac{\mathbf{x}-\mathbf{x}_{\mathbf{p}}}{h}\right)\frac{1}{h^3}\Pi\left(\frac{\mathbf{x}_i-\mathbf{x}_{\mathbf{p}}}{h}\right)\mathrm{d}\mathbf{x},$$

which only has a non-zero (constant) integrand if the "cubical cloud" of the particle and the cubical mesh cell overlap

Note that CIC is naturally computationally more expensive than NGP because it has to interpolate (=compute fractional weights) across 2^d cells (*d* being the dimension) while NGP uses only the initial cell in which a particle is embedded

Higher-order mass assignment methods can enforce continuity of the derivative of the force by smoothing across an even larger number of cells than CIC, but they become progressively more computationally expensive because more fractional weighs have to be computed

One example is the triangular shaped cloud (TIC), which spreads mass over 3^d cells:



Triangle is defined to have total base length 2h. Assignment function *W* computes area of overlap of triangle with cell