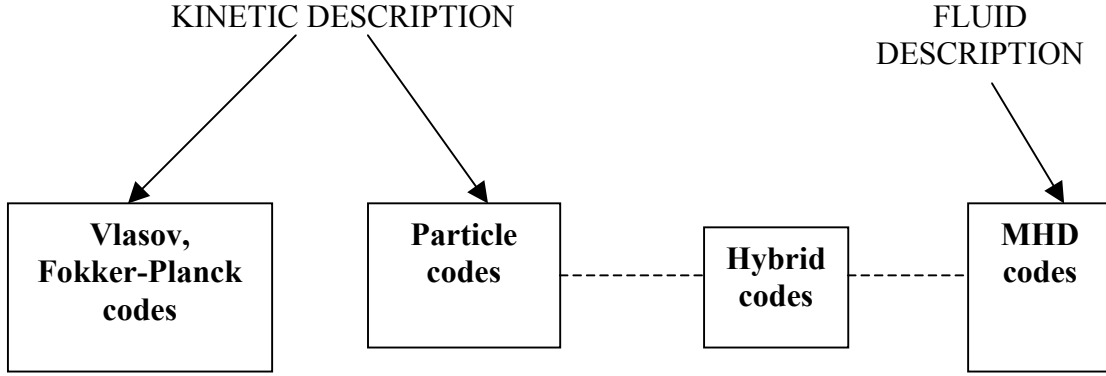


# PLASMA PHYSICS

## VI. SIMULATION OF PLASMA PHENOMENA

### Road Map of Simulation Methodologies

Broadly speaking there are two main types of approaches to the simulation of plasma behaviour, they are based on kinetic and fluid descriptions as shown in the schematic diagram below.



Kinetic codes are used when the velocity distribution of the particles in the plasma is non-Maxwellian (ie. the plasma is not in thermal equilibrium and we can not characterise the ions and electrons with a temperature,  $T$ ). In this case we must deal with a time varying particle distribution function  $f(x, y, z, v_x, v_y, v_z, t)$ . This distribution function must satisfy the **Boltzman equation** :

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = \left( \frac{\partial f}{\partial t} \right)_c$$

This essentially says that any change in the distribution function measured moving with the particles (i.e. convective time derivative) will be a result of collisions (term on RHS). This is because in the absence of collisions any particle moving with velocity  $\mathbf{v} = (v_x, v_y, v_z)$  will trace out a path according to its equation of motion. Taking snap shots at times  $t$  and  $t + \Delta t$  would give  $f(x, y, z, v_x, v_y, v_z, t) = f(x + v_x \Delta t, y + v_y \Delta t, z + v_z \Delta t, v_x + (F_x/m) \Delta t, v_y + (F_y/m) \Delta t, v_z + (F_z/m) \Delta t, t + \Delta t)$ . Any deviations from this must be caused by collisions between particles.

**Vlasov equation** assumes that there are no collisions (e.g. plasma sufficiently hot) and that the force  $\mathbf{F}$  is entirely electromagnetic.

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

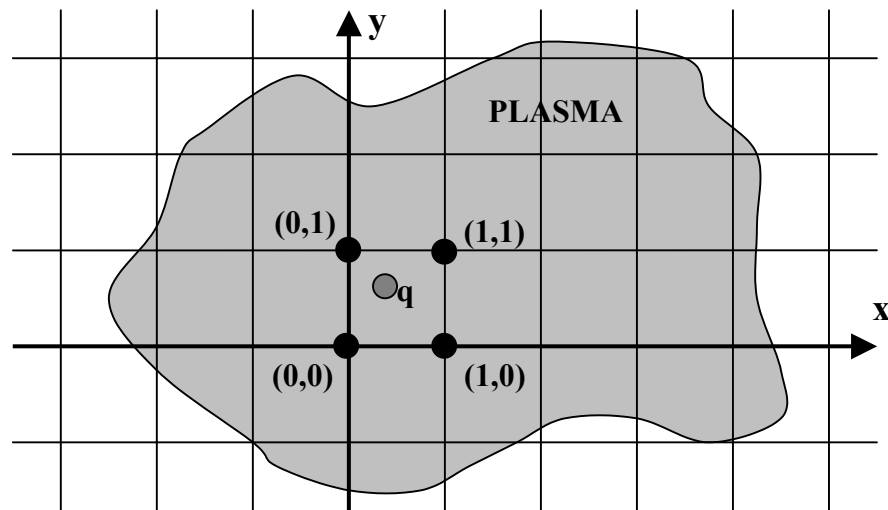
Kinetic codes solve such equations numerically.

**MHD codes** on the other hand are based on the fluid equations we derived in Chapter 3. Depending on the nature of the problem the plasma can be treated by the multiple fluid equations describing the different species present or the single fluid description can be used. The resulting differential equations are solved numerically assuming approximate values for the required transport coefficients.

More recently codes which combine a fluid description simulation in one part of the plasma with a full kinetic treatment elsewhere have been developed. These types of codes are called **hybrid codes**. A major issue here is how to marry the two models at the interface between the regions. This is an on-going area of research.

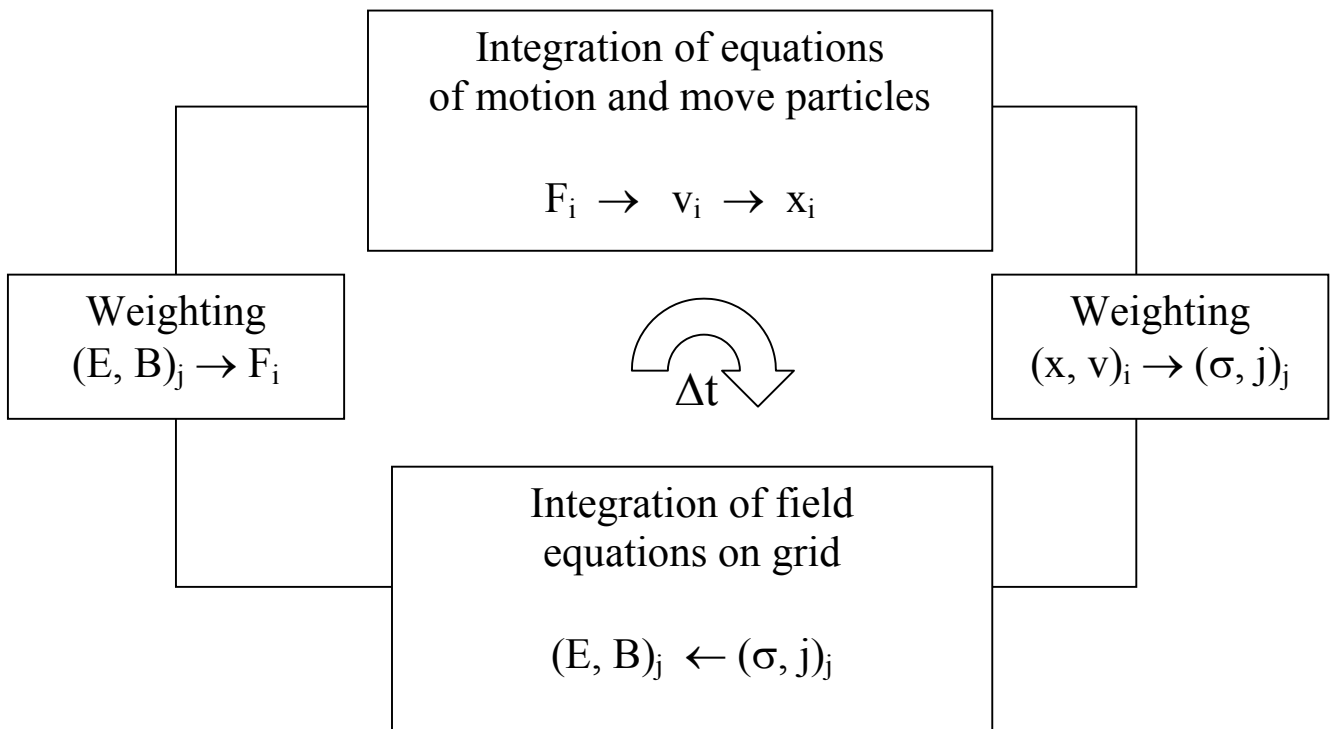
## Particle-in-cell codes (PIC)

In this method the trajectories of ions and electrons are calculated explicitly. Computers aren't powerful enough to simulate all of the particles in even a low density plasma so we use a single particle to represent many particles (e.g.  $10^4$ ).



A grid is set up in the plasma region. Charge and current densities,  $\sigma$  and  $\mathbf{j}$ , are determined, from which the electric and magnetic fields are calculated at the grid points. A charged particle  $q$  at  $(x,y)$  will typically be counted in terms of  $\sigma$  at the nearby grid points  $(0,0)$ ,  $(1,0)$ ,  $(1,1)$ ,  $(0,1)$  and in terms of  $\mathbf{j}$  at the faces between these points. Once the  $\mathbf{E}$  and  $\mathbf{B}$  fields have been calculated at the grid points the force on  $q$  and its subsequent motion can be calculated. The procedure is then repeated for the new position of  $q$ .

The particle quantities, such as velocity and position are known at the particle and may take on all values in phase (i.e.  $\mathbf{v}$  and  $\mathbf{x}$  space). These quantities are labelled with the index  $i$  and the associated velocity and position vectors are denoted  $\mathbf{v}_i$  and  $\mathbf{x}_i$ . The field quantities are determined only on the spatial grid, known only at discrete grid points labelled with the index  $j$ , the field vectors at the  $j^{\text{th}}$  grid point are denoted by  $\mathbf{E}_j$  and  $\mathbf{B}_j$ . A typical calculation cycle is shown in the diagram below



There are many variations of this cycle. For example the electrons (with relatively large  $\omega_{pe}$  and  $\omega_{ce}$ ) may be advanced at a relatively small  $\Delta t_e$ ; the ions (with much smaller  $\omega_{pi}$  and  $\omega_{ci}$ ) may be advanced with a relatively large  $\Delta t_i$ ; and the fields may be obtained on yet a third timescale,  $\Delta t_f$ , possibly relatively short for electromagnetic fields (waves move faster than particles) or relatively large for observing low frequency effects. The differential equations linking the particle and field quantities are integrated numerically, usually by finite difference methods.

### Weighting

It is necessary to calculate the charge density on the discrete grid points from the continuous particle positions and (after the fields are obtained), to calculate the force at the particles from the fields on the grid points. Essentially some form of interpolation among the grid points nearest to the particle is required. As always there is a trade off between computational efficiency and accuracy.

The simplest and fastest weighting procedure is called **zero-order** or **nearest-grid-point (NGP)** weighting. In this system particles are simply assigned to the nearest grid point and the resolution to which we can track a particle as far as the grid is concerned is the limited to the distance between grid points. From the grid's point of view when a moves in through one wall the density jumps up sharply and as the particle leaves through another wall the density drops sharply. The particles then appear to have rectangular shape with width  $\Delta x$ . Because the density changes so abruptly at grid points, zeroth order weighting results in noisy electric and magnetic fields.

***First order*** weighting smooths the density and field fluctuations relative zero-order weighting, but requires additional computational expense in accessing two, four or eight grid points per particle in one, two or three dimensional simulations respectively. This type of weighting assigns the density proportionally among the particle's neighbouring grid points according to its relative proximity to each one. From the perspective of the grid each of the particles then appears to be a cloud with peak density at the position of the particle.

Higher-order weighting by the use of quadratic and cubic splines rounds off further the roughness in particle shape and reduces density and field noise, but at the cost of more computation.

## **Verification**

Simulations must be verified before their results are believed. The results of problems with known solutions, both experimental and analytical, should be computed using the code. The code should also be verified by showing invariance of results as the non-physical computer parameters ( $\Delta t$ ,  $\Delta x$ , number of particles etc) are changed.