Next step is to apply variational calculus, i.e solve $\delta L = 0$, which yields the familiar Euler-Lagrange equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\mathbf{r}}_i} - \frac{\partial L}{\partial \mathbf{r}_i} = 0.$$

Since **r**, for reach particle, is an (implicit) function of the smoothing length *h*, clearly in taking derivatives it is important to have an educated definition of *h*. In early SPH work the smoothing length was defined using a variable number of neighbors to cope with the issue that in low density regions kernels tend to be large and in high density regions they tend to be small, giving very different volume sampling at high and low density. This results in energy conservation errors.
In modern SPH one fixes the number of neighbors to avoid that (we can also afford more particles now in simulations so we can sample everywhere better). *This ensures mass conservation within the kernel if the masses of particles are equal , i.e.:*

$$\rho_i h_i^3 = const$$

The latter becomes a useful conservation property that we can exploit in deriving the SPH hydro equations from the Euler-Lagrange equations:

$$m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \frac{P_j}{\rho_j^2} \frac{\partial \rho_j}{\partial \mathbf{r}_i}$$

using the conserved mass constraint becomes

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right]$$

$$f_i = \left[1 + \frac{h_i}{3\rho_i} \frac{\partial \rho_i}{\partial h_i} \right]^{-1}$$

To get to the above equation we start from:

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \nabla_i \rho_j + \frac{\partial \rho_j}{\partial h_j} \frac{\partial h_j}{\partial \mathbf{r}_i}$$

$$\frac{\partial \rho_j}{\partial h_j} \frac{\partial h_j}{\partial \mathbf{r}_i} \left[1 + \frac{3\rho_j}{h_j} \left(\frac{\partial \rho_j}{\partial h_j} \right)^{-1} \right] = -\nabla_i \rho_j .$$

Finally we combine:

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \left(1 + \frac{h_j}{3\rho_j}\frac{\partial \rho_j}{\partial h_j}\right)^{-1} \nabla_i \rho_j .$$ with $$\nabla_i \rho_j = m_i \nabla_i W_{ij}(h_j) + \delta_{ij} \sum_{k=1}^{N} m_k \nabla_i W_{ki}(h_i)$$

to get:

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right]$$

Note that this SPH fluid momentum equation is a simple differential equation (with derivatives of the kernel) as opposed to the original PDEs . We have maintained the differential form, though, which will be problematic at discontinuities (eg shocks or shearing flows, such as two fluids with different properties moving over an interface)

Thermodynamics in SPH: entropy and energy

To describe the gas thermodynamics one can use (specific) internal energy *u* or entropy *s* as the fundamental variable. Most popular modern SPH codes use one or the other - eg GADGET (public) uses entropy, GASOLINE/ChaNGa (our codes at ICS) use energy.
The formulation with entropy is simpler in absence of discontinuities in the flow (eg shocks) because then the system is *thermodynamically reversible* and entropy is simply conserved.
The formulation based on internal energy requires that the internal energy equation be solved always, even in a reversible system, but because of that is also automatically generalized to *irreversible conditions*

With entropy (reversible *isentropic* case):

$$P_i = A_i \rho_i^{\gamma} = (\gamma - 1)\rho_i u_i$$

with $A_i(s)$ = constant depending on specific entopy *s*

$$\longrightarrow \quad u_i(\rho_i) = A_i \frac{\rho_i^{\gamma-1}}{\gamma - 1}$$

With internal energy equation (reversible case):

Start with $$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{P_i}{\rho_i^2} \sum_j \mathbf{v}_j \cdot \frac{\partial \rho_i}{\partial \mathbf{r}_j}$$

obtained from total derivative over time of $$u_i(\rho_i) = A_i \frac{\rho_i^{\gamma-1}}{\gamma - 1}.$$

which using the SPH expression for the spatial derivative of the density becomes

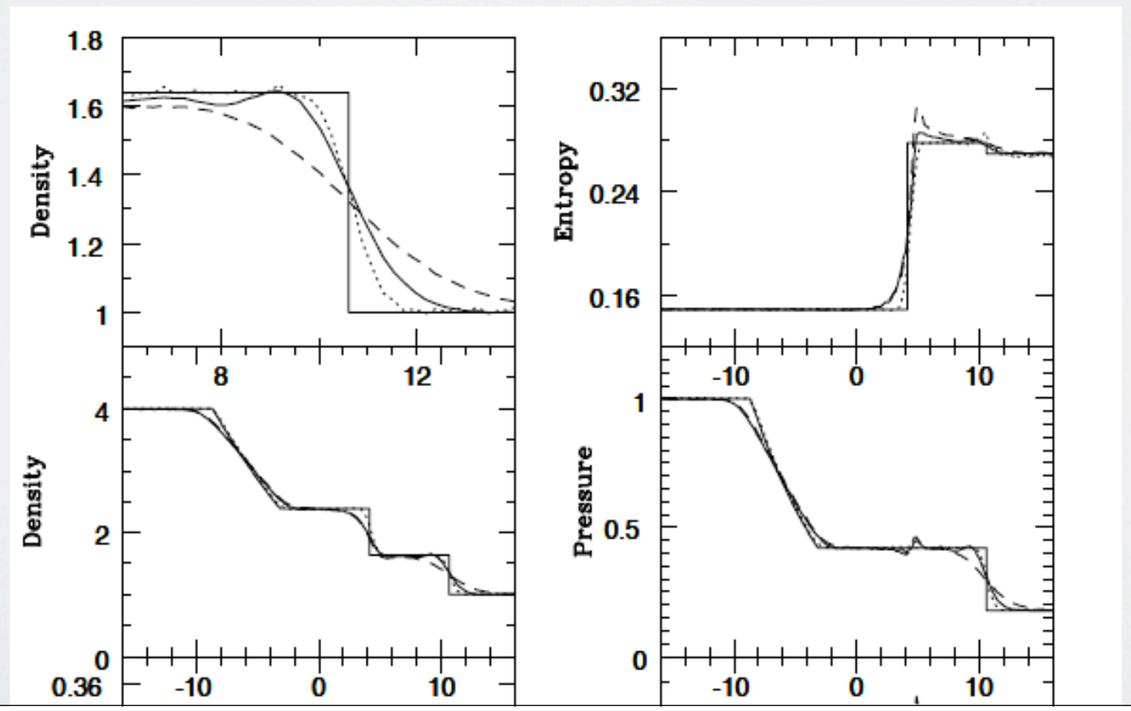
$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = f_i \frac{P_i}{\rho_i^2} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij}(h_i)$$

The differential form of the equations used so far formally gives diverges, hence breaks down, at sharp discontinuities such as shocks. There are, in computational fluid dynamics, two general ways to overcome this issue:

(a) switch to integral (conservation form)
(b) introduce dissipation term --> *viscosity*

Artificial viscosity

In SPH all functions are affected by Poisson noise, namely the fact that the fluid is coarsely sample by particles so that eg positions or velocities deviate from a reference "exact" profile always - the error goes ~ $1/N^{1/2}$, where both the total N and the N_{neigh} inside a kernel matter in the "error". This alone implies that there cannot be really sharp discontinuities by construction, which means equations in differential would not brake down in practice. But if one tries to model a shock front in SPH with the equations used so far ringing arises post-shock, and the shock profile is never sharp (also true for certain forms of artificial viscosity)



In real fluids there is viscosity at the molecular scale, due to Van der Waals forces, while on macroscopic scales there can be other phenomena producing a viscous force, such as magnetic currents or radiation pressure drag, that are not contained in Euler equations.
In shocks kinetic energy is dissipated into heat (otherwise the shock will never end).
However, shocks or contact discontinuities do not occur only in the presence of physical effects not contained in Euler eqs- they can arise in a fluid described as inviscid because dissipation of kinetic energy into heat occurs due to (unresolved) molecular viscosity (where the perfect fluid description breaks down anyway). Analytical "jump" conditions for shocks in an inviscid fluid are well known indeed.

But if one sticks to SPH the first issue is not how to dissipate kinetic energy into heat but how to model a sharp discontinuity in a function in a smooth and accurate way, namely as eg capturing relation between pre-shock and post-shock values in cases where analytical solutions are known, such as in isothermal and adiabatic shocks.

to broaden and smooth the profile of fluid variables across discontinuities, making it numerically tractable.
There are various formulations of artificial viscosity that have been proposed through the years. *One of the main challenges it to formulate viscosity in a way that it turns off or at least becomes sufficiently small away from discontinuities/shocks. Otherwise the fluid does not obey the Euler equations anymore* (and not even Navier-Stokes for a viscous fluid since the viscosity here is introduced only for numerical reasons!).

The standard Monaghan viscosity is introduced for each particle using conservation laws as constraints:

$$\left.\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t}\right|_{\mathrm{visc}} = -\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i \overline{W}_{ij}$$

using

$$\overline{W}_{ij} = \frac{1}{2}\left[W_{ij}(h_i) + W_{ij}(h_j)\right]$$

The symmetrized kernel is such that, if the viscous tensor Πij is symmetric, the viscous force between two particles will be antisymmetric, allowing conservation of linear and angular momentum

In order to ensure conservation of energy one needs to balance the extra kinetic energy term induced by the viscous force with an extra term in the internal energy equation, expressing it with an entropy term or a thermal energy term:

$$\left.\frac{\mathrm{d}A_i}{\mathrm{d}t}\right|_{\mathrm{visc}} = \frac{1}{2}\frac{\gamma-1}{\rho_i^{\gamma-1}}\sum_{j=1}^{N} m_j \Pi_{ij}\mathbf{v}_{ij}\cdot\nabla_i \overline{W}_{ij}$$

$$\left.\frac{\mathrm{d}u_i}{\mathrm{d}t}\right|_{\mathrm{visc}} = \frac{1}{2}\sum_{j=1}^{N} m_j \Pi_{ij}\mathbf{v}_{ij}\cdot\nabla_i \overline{W}_{ij}$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$

The most common formulation of the viscous tensor is:

$$\Pi_{ij} = \begin{cases} \left[-\alpha c_{ij}\mu_{ij} + \beta\mu_{ij}^2\right]/\rho_{ij} & \text{if } \mathbf{v}_{ij}\cdot\mathbf{r}_{ij} < 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu_{ij} = \frac{h_{ij}\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + \varepsilon h_{ij}^2}.$$

*it is thus non-zero only if particles are approaching one another

* c_{ij}, h_{ij} and ρ_{ij} are defined for particle pairs using arithmetic averages (c_{ij} *is* the thermal sound speed)

- the entropy production is positive definite because viscosity turns only for particles that approach each other, for which $\mu_{ij} < 0$

- there is a dependence on the smoothing length, so that at higher resolution the viscous term becomes smaller ($\varepsilon \sim 0.01$ to prevent singularities for particles getting too close).

- the quadratic term was introduced later but it is crucial in shocks because it avoids particle interpenetration, which is an unphysical effect. It broadens and smooths further the discontinuity relative to the linear term

- typical choice for α and β are 0.5-1 and 2. However, when there are no shocks or strong discontinuities in the problem (eg rotationally supported disk) the coefficients might be set to as small as possible.

In differentially rotating flows at low resolution it may appear that particles are approaching due to the noisiness of the flow sampling. Rather than reducing the coefficients a cleaner alternative is to add a **rot v** dependent term in Π so that the viscosity goes to zero in shearing/rotational flows

Eulerian hydrodynamics methods

In this class of methods one solves the eulerian form of the fluid equations onto a mesh. The mesh can be cartesian but also more complex (*unstructured*, eg using tasselation with triangles or tethraedra as in cylindrical or spherical coordinates).
There are different methods for the discretization of the equations:

finite difference, finite volume and finite elements

are the most important. They all start by discretizing the PDEs on the mesh and convert them into algebraic equations, in the same spirit of multigrid for the Poisson equation. Indeed the multigrid method for Poisson is a finite difference method.
In finite volume, which is much used in astrophysics, the key aspect is to construct *fluxes* for all quantities at mesh faces and use integral form of the PDEs to express conservation of such fluxes over a certain small volume (normally the volume of a cell).

In order to compute fluxes volume integrals that contain a divergence term are all converted into surface integral (typical example is mass flux associated with continuity equation). Finite volume elements thus solve the Euler equations in *conservative form.* But as we will see there is a numerical effect, numerical diffusion, that can compromise conservation laws.

In finite element, much used in engineering, one solves the PDEs on subvolumes (elements) with a geometry appropriate to the problem and then connects the solutions
at different domains using algebraic relations

In the context of finite difference, a specific "sub-method" is the *method of lines*, in which ones uses finite difference to represent all derivatives except one, reducing a set of PDEs into a set of ODEs. This is useful in time dependent problem, where the only derivative that is not finite differenced is the time derivative

An example of application of the method of lines is that of the heat diffusion equation (we will use it also later);

$$\frac{\partial u}{\partial t} + \lambda \frac{\partial^2 u}{\partial x^2} = 0.$$

which can be discretized into:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + \lambda \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} = 0.$$

which makes it clear that now, for each cell i we can integrate the PDEs as a set of time dependent ODEs. We will see that numerical instabilities can arise in certain types of time dependent PDEs, namely in hyperbolic PDEs in which a clear direction of propagation of the *information* on the state of the system exists. A prototypical example is the wave equation, in which the propagation of the information occurs at the finite speed of light. On the contrary elliptical PDEs such as the Poisson equation have instantaneous propagation of information

Finally there are *spectral methods* in which the solution of the PDEs is approximated as linear superposition of simpler functions, e.g. wave-like functions. An example of spectral method is the FFT method for the Poisson equation. The PDEs are transformed in algebraic equations or ODEs

With spectral methods there are flow conditions that are impossible to capture because they cannot be reduced to a linear expansion -- one example is high compressibility behaviour, shocks being the extreme case, or contact discontinuities in a multi-phase fluid. Since these situations are common in astrophysics these methods are not popular. But for example in problems in which the fluid can be treated as relatively smooth and incompressible they are commonly adopted (eg atmospheric/climate science)

The advection problem

Numerical eulerian hydrodynamics stems from coping with the notion of advection.
Think of a fluid laid down on a cartesian grid at some time *t=0*: integrating the fluid equations means essentially to advect the functions defining the fluid (density, velocity, pressure) through the grid, namely find their values on the grid at a later time *t*.
These functions are defined at each point **x** of the grid at time t=0. *Example: advect a uniform gas cloud moving along x (1D)*

In 1D the kind of equation that we want to solve is an advection equation

$$\frac{\partial u}{\partial t} + v \cdot \frac{\partial u}{\partial x} = 0,$$

with *u=u(x,t)*
with *v* constant

Note that e.g. the continuity equation belongs to this category but ---> advection of **ρ** for a spatially dependent *v*
In general this is a kind of hyperbolic equation (wave-type equation)

One can show that for any q(x) the function u(x,t) = q(x - vt) is a solution to the advection equation, and q(x) can be simply interpreted as the initial condition (q (x, 0) for any *v*).

Then the solution at any given time t is simply a copy of q(x) translated by vt

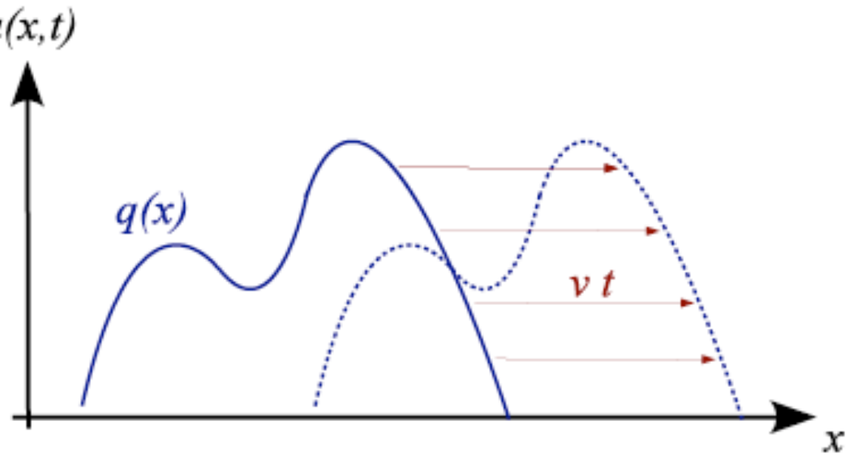


Fig. 20 Simple advection with constant velocity to the right.

Points that star at a certain location x_0 are advected to a new location $x_0 + vt$. They values at different times are connected by the *characteristics* , straight lines that follow the propagation of information (e.g. density) from *upstream* to *downstream*. Note in this terminology *downstream is where the flow is going to*

For this simple example we know the solution but we could as how we could recover it numerically (and how well).

We can use straightforward finite differencing using the method of lines

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + v\frac{u_{i+1} - u_{i-1}}{2h} = 0.$$

and for the time derivative

$$u_i^{(n+1)} = u_i^{(n)} - v\frac{u_{i+1}^{(n)} - u_{i-1}^{(n)}}{2h}\Delta t$$

We have not covered time integration yet but the above method (Euler method) is intuitively the analogous on finite differencing at fixed time (n and n+1 refer to the values of the variable at two subsequent timesteps and Δt is the timestep, namely the chosen discretization scale on the time axis

If one uses these two equations to update the function u(x,t) one finds out that the solution is numerically unstable - e.g. if *u* is a step function the returned solution is oscillatory.

The reason for the numerical instability is that the flow propagates as the characteristics, namely from upstream to downstream. The update formula includes values from both upstream (u_{i-1}) and downstream (u_{i+1}).
But this is inconsistent with how the information flows, namely along the characterstics - the downstream value has not happened yet when we want to estimate u_i (causality problem) ----> we can switch to a one-sided estimate which only involves upstream values

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + v\frac{u_i - u_{i-1}}{h} = 0.$$

With this *upwind differencing* numerical instability disappears

There is however a new issue that arises with this form of the update. It is the *numerical diffusion* that affects the robustness of the solution (namely the solution is now stable but not the exact solution due to diffusion)

We can figure out where this diffusion come from. Let us rewrite the finite differencing update as

$$\frac{u_i - u_{i-1}}{h} = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{u_{i+1} - 2u_i + u_{i-1}}{2h}$$

Hence this upwind scheme can be also written as:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + v\frac{u_{i+1} - u_{i-1}}{2h} = \frac{vh}{2}\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

But remembering the diffusion equation we have also:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \simeq \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \longrightarrow \frac{\partial u}{\partial t} + v \cdot \frac{\partial u}{\partial x} = D\frac{\partial^2 u}{\partial x^2},$$

The diffusion coefficient D is $vh/2$ and therefore if we increase resolution (=reduce grid spacing h) the diffusion decreases (it will be zero for $h \to 0$.)
Diffusion becomes larger also for a larger velocity v. In problems that involve supersonic flows (e.g. in cosmology this is common, as in the collapse that forms galaxies) this can be a major problem, but we will see there are ways to handle such situations.

For the stability of the time integration one needs to pick the timestep in a physically self-consistent way. That is the timestep cannot be larger than the time it takes to propagate the flow from upstream and downstream, namely $\Delta t_{max} = h/v$. Or else one should include a term more deeper upstream (e.g. u_{i-2}).

This timestepping considerations lead to Courant-Friedrichs-Levy (CFL) timestep condition, $\Delta t_{max} <= h/v$. Similar conditions apply for the integration of other types of hyperbolic equations. This is a necessary condition that holds also for the integration of the SPH equations by the way (in that case it applies to individual particles rather than to the solution of an equation because they carry the flow properties directly).