The next step is to explore how the REA scheme, and more fundamentally the Godunov method, can be extended to *(a) multiple dimensions and (b) higher order accuracy in space and time.*

Let us look at (b) first. We start by defining the error, namely *the deviation from an exact solution*, for example for the density of the flow:

$$L1 = \frac{1}{N}\sum_i |\rho_i - \rho(x_i)|$$

The order of the numerical hydro method is then related to how fast the numerical solution converges to the exact solution as we increase the resolution, namely the number of grid points/ cells, *N.*

We talk about 1st order scheme if convergence goes as N^{-1} (error halves if we double N), and of 2nd order scheme if convergence goes as N^{-2} (error/4 if we double N)

Higher order schemes, eg 4th order, are also possible but they become progressively more difficult to code and more expensive computationally. Also, there might be other numerical errors in the method (eg numerical diffusion for in advection for non-exact Riemann solvers) that could overtake the order of the scheme and make it unnecessary in practice.

In general going from 1st order to 2nd order is well motivated and computationally not prohibitive, so that should be the target.

The first step in constructing a 2nd order Godunov method is to replace the piece-wise constant with a piece-wise linear reconstruction, namely interpolate linearly between boundary values of cells eliminating the "jump" at interfaces.

However this can actually create larger discontinuities at certain cell interfaces which the calls for *slope limiters to avoid numerical discontinuities (i.e. in this case the "jump" occurs for numerical reasons, namely as a result of the linear reconstruction) rather than due to the physical state of the flow (see below)*

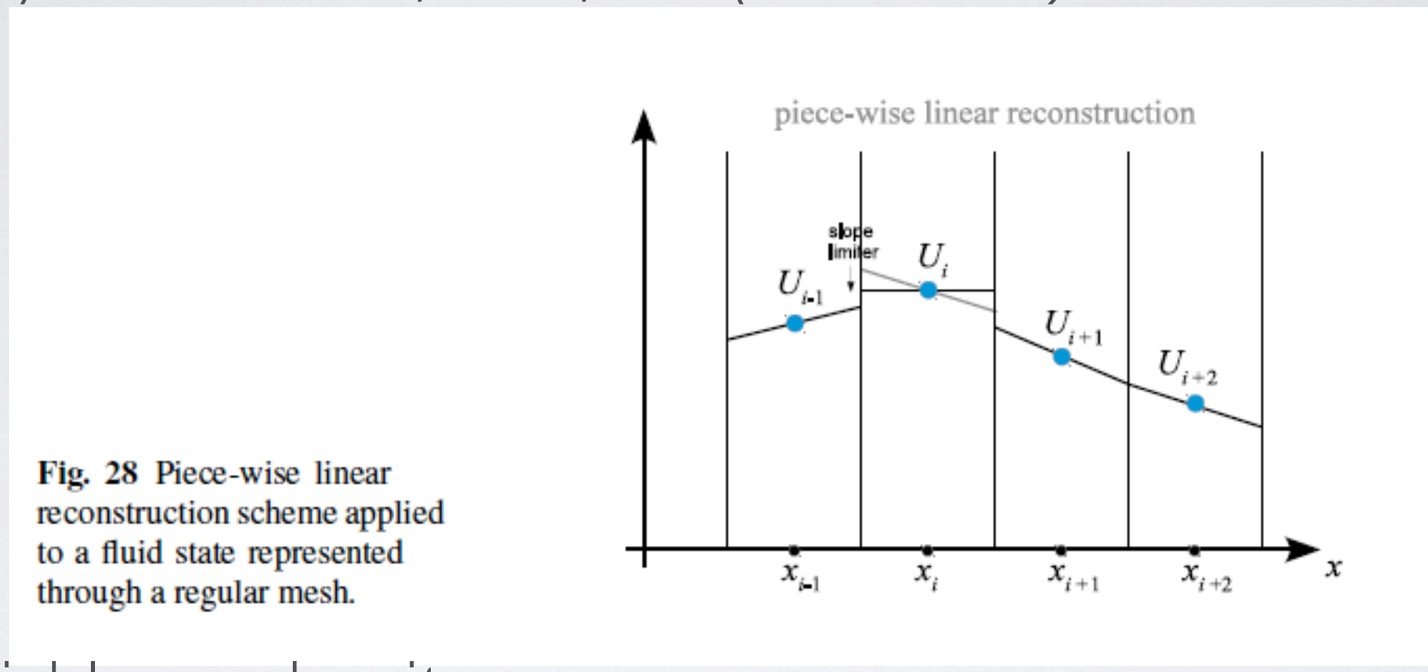


Fig. 28 Piece-wise linear reconstruction scheme applied to a fluid state represented through a regular mesh.

The state variable, eg density, can be then estimated using a finite difference stencil to obtain the "left" and "right" values at cell faces needed as the new boundary values for the Riemann problem

$$\rho^L_{i+\frac{1}{2}} = \rho_i + (\nabla\rho)_i \frac{\Delta x}{2},$$

$$\rho^R_{i+\frac{1}{2}} = \rho_{i+1} - (\nabla\rho)_{i+1} \frac{\Delta x}{2}.$$

This in general will have to be done for the whole state vector of the system, including its time update (note update by half a timestep to guarantee second order accuracy in time and stability)

$$\mathbf{U}^L_{i+\frac{1}{2}} = \mathbf{U}_i + (\partial_x \mathbf{U})_i \frac{\Delta x}{2} + (\partial_t \mathbf{U})_i \frac{\Delta t}{2},$$

$$\mathbf{U}^R_{i+\frac{1}{2}} = \mathbf{U}_{i+1} - (\partial_x \mathbf{U})_{i+1} \frac{\Delta x}{2} + (\partial_t \mathbf{U})_{i+1} \frac{\Delta t}{2}.$$

The spatial derivatives are slope-limited, not just a simple gradient using a cell-based finite difference stencil. The evaluation of the derivatives can be done by exploiting the Jacobian matrix of the Euler equations:

$$\partial_t \mathbf{U} = -\partial_x \mathbf{F}(\mathbf{U}) = -\frac{\partial \mathbf{F}}{\partial \mathbf{U}} \partial_x \mathbf{U} = -\mathbf{A}(\mathbf{U}) \partial_x \mathbf{U},$$

where $\mathbf{A}(\mathbf{U})$ is the Jacobian matrix, so that one can estimate the time derivative based on the spatial derivative as:

$$(\partial_t \mathbf{U})_i = -\mathbf{A}(\mathbf{U}_i) (\partial_x \mathbf{U})_i.$$

so that the full update can be done as:

$$\mathbf{U}^{L}_{i+\frac{1}{2}} = \mathbf{U}_i + \left[\frac{\Delta x}{2} - \frac{\Delta t}{2}\mathbf{A}(\mathbf{U}_i)\right](\partial_x \mathbf{U})_i,$$

$$\mathbf{U}^{R}_{i+\frac{1}{2}} = \mathbf{U}_{i+1} + \left[-\frac{\Delta x}{2} - \frac{\Delta t}{2}\mathbf{A}(\mathbf{U}_{i+1})\right](\partial_x \mathbf{U})_{i+1}.$$

This is the so-called MUSCL-Hancock scheme, which is a 2nd order accurare extension of the Godunov method.
Higher order methods such as PPM (piece-wise parabolic method) use reconstruction at cell interfaces with higher order polynomials such as parabolic functions. ENO and WENO schemes are even higher order than PPM. They involve finite differencing between more cells to reconstruct values of cell interfaces (a bit like higher CIC vs. nearest grid point method). All these are conservative in the sense that when integrating over the cell the values are conserved (eg mass as an integral of density) is conserved.

Godunov-type solver at higher dimensionality

There are two major types of schemes to solve the 3D Euler equations with Godunov; the *split* and *unsplit* schemes
In split schemes one reduces the 3D problem to sequential 1D problem. Recall each of the four Euler equations can be written as an hyperbolic equation with the form:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0,$$

In the 3D case one can show that the full system of equations can be written using flux operators for the various quantities by "grouping" terms involving derivatives in one spatial direction to obtain:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} + \partial_z \mathbf{H} = 0.$$

One can then reduce this to three 1D problems to be solved sequentially for the three fluxes. Note however that each flux still contains the full 3D information (*augmented* 1D)

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0,$$
$$\partial_t \mathbf{U} + \partial_y \mathbf{G} = 0,$$
$$\partial_t \mathbf{U} + \partial_z \mathbf{H} = 0.$$

In practice what one does is to solve the Riemann problem for one dimension at a time, updating one flux at a time, and then doing the full time update at the end. Each step to update in one direction is called a *sweep*.

The full time update can be first-order or second-order accurate. For higher order accuracy the time evolution operators must be applied alternating in reverse order (see Lecture notes pag. 73)

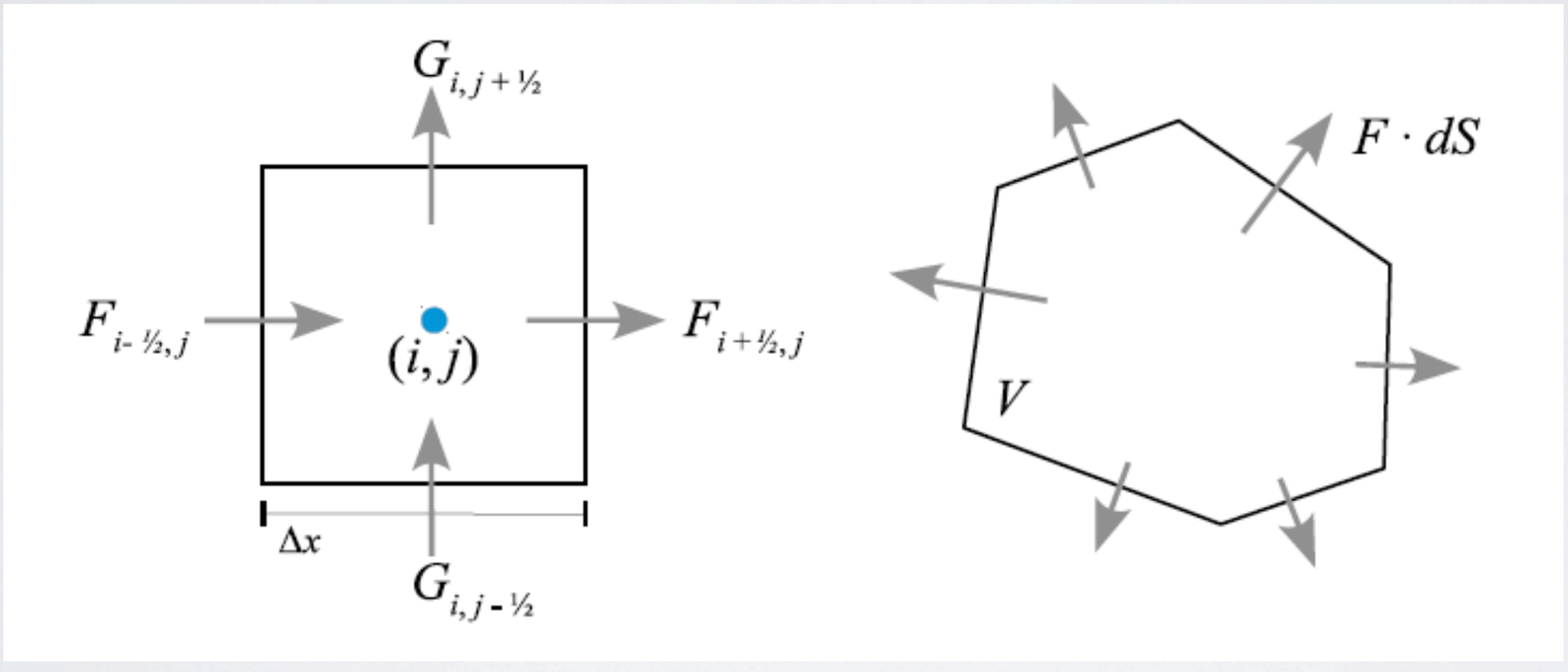
In *unsplit* schemes the flux updates are applied simultaneously to each cell. This is the choice for irregular meshes (eg in modern lagrangian mesh codes in which each cell has a varible shape).

For example in 2D for structured cartesian meshes:

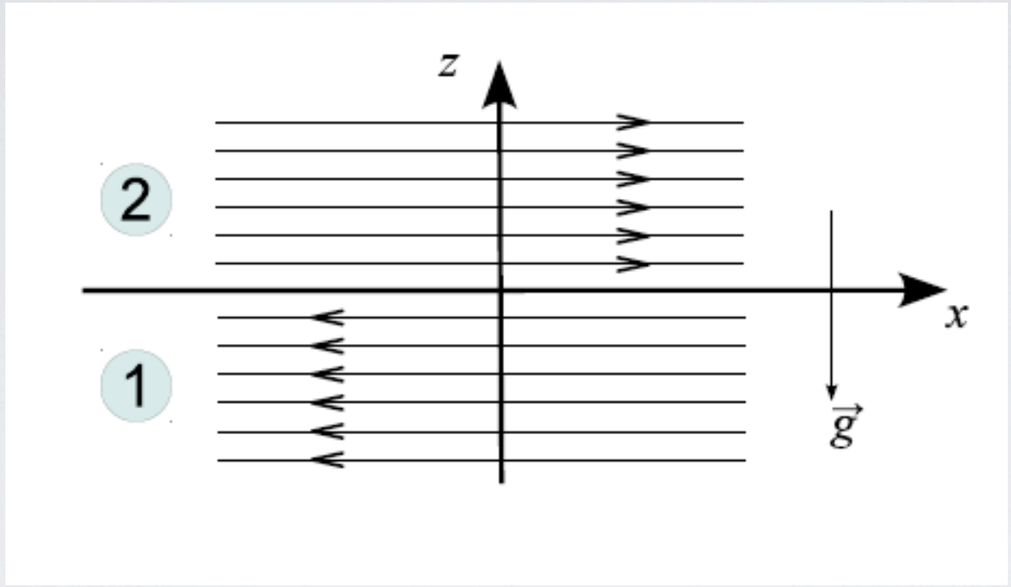
$$U_{i,j}^{n+1} = U_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2},j} - \mathbf{F}_{i+\frac{1}{2},j}\right) + \frac{\Delta t}{\Delta y}\left(\mathbf{G}_{i,j-\frac{1}{2}} - \mathbf{G}_{i,j+\frac{1}{2}}\right)$$

For unstructured (irregular) meshes one can use the divergence theorem to compute the state vector updates through a cell:

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \frac{\Delta t}{V} \int \mathbf{F} \cdot \mathbf{dS},$$

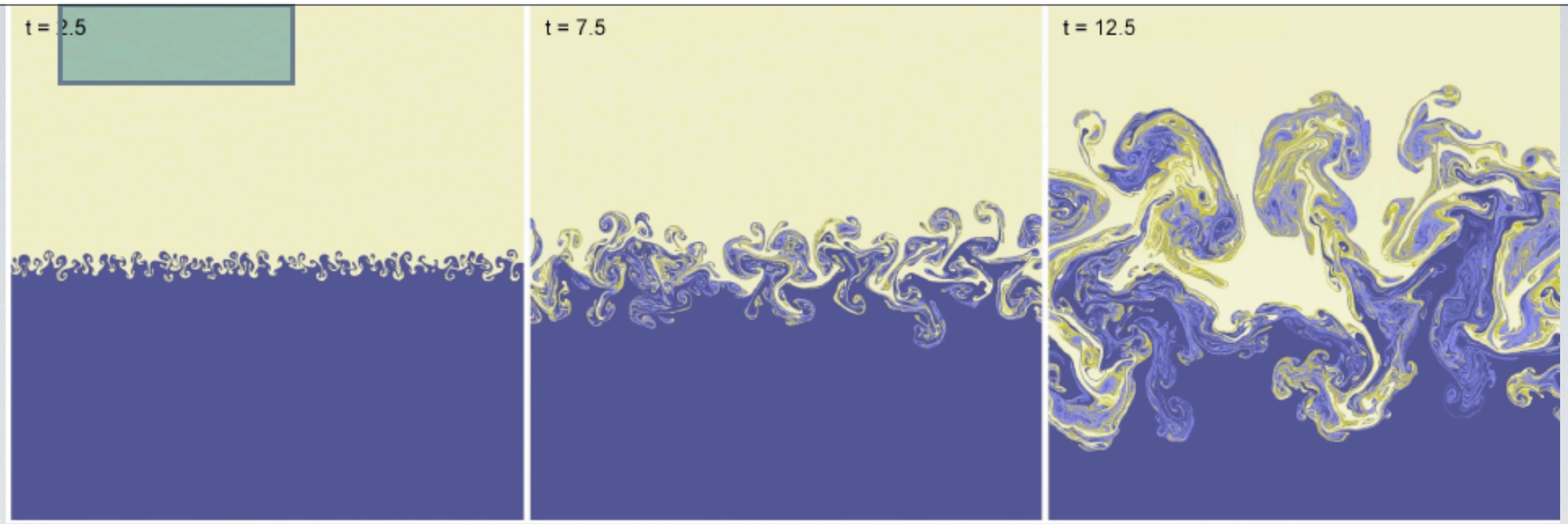


A testbed for numerical hydro: two-fluid instabilities

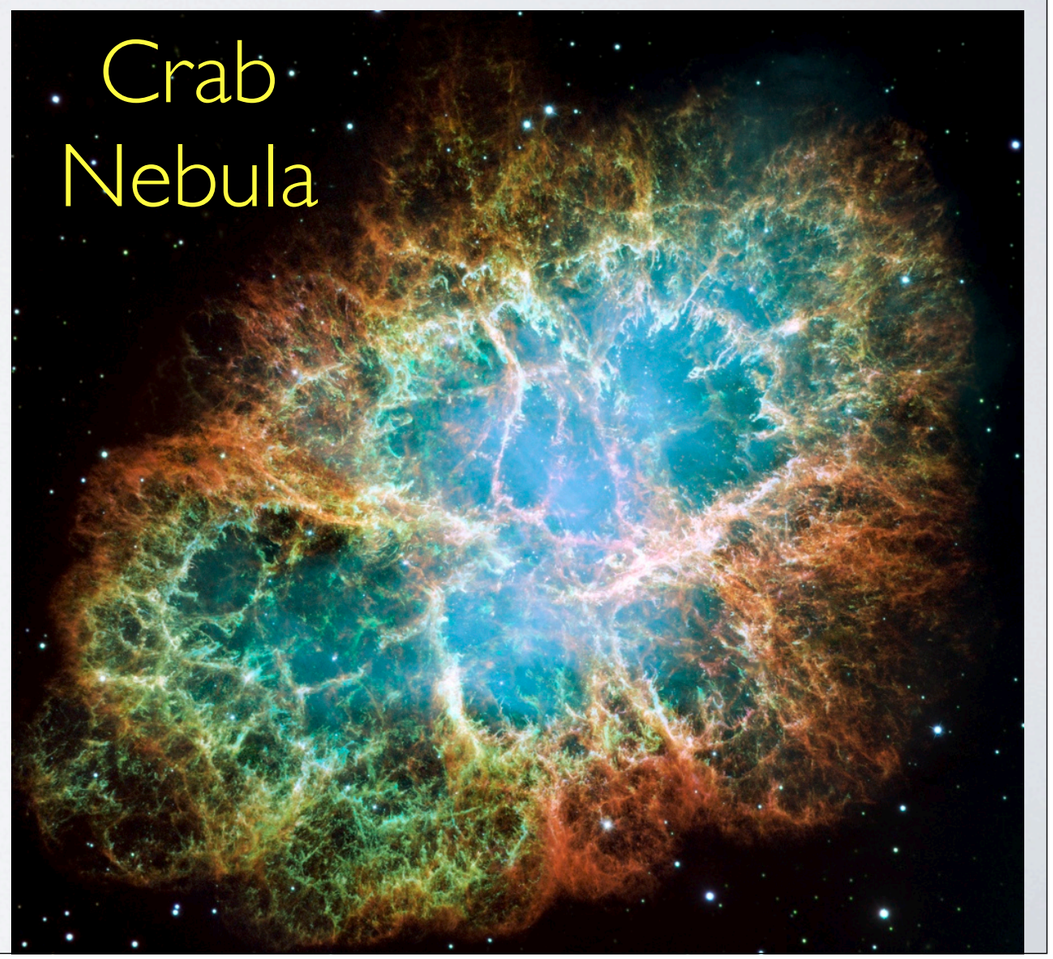


Consider two fluids shearing one past each other, having different densities. There can or cannot be an external force, such as gravity, and the relative velocity can also be null.

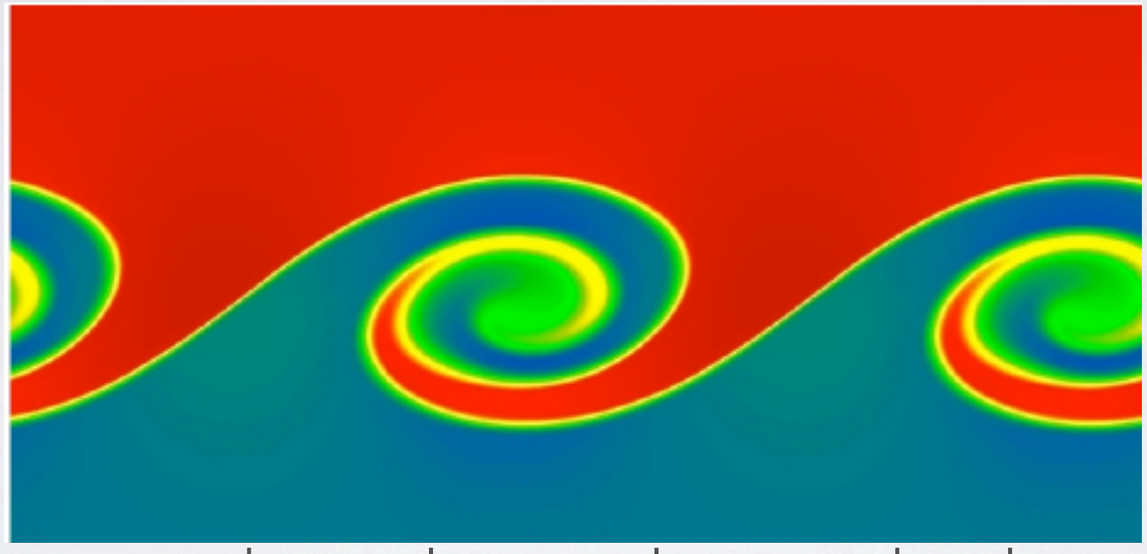
In the simplest case in which $\Delta V = 0$ and the heavier fluid sits above the lighter fluid one can show analytically, using linear analysis, that such a configuration is unstable.
The fluids rearrange then themselves and mix until the lighter fluid is above the heavier fluid. A simulation done with a Godunov-type grid-based method shows the development of turbulence



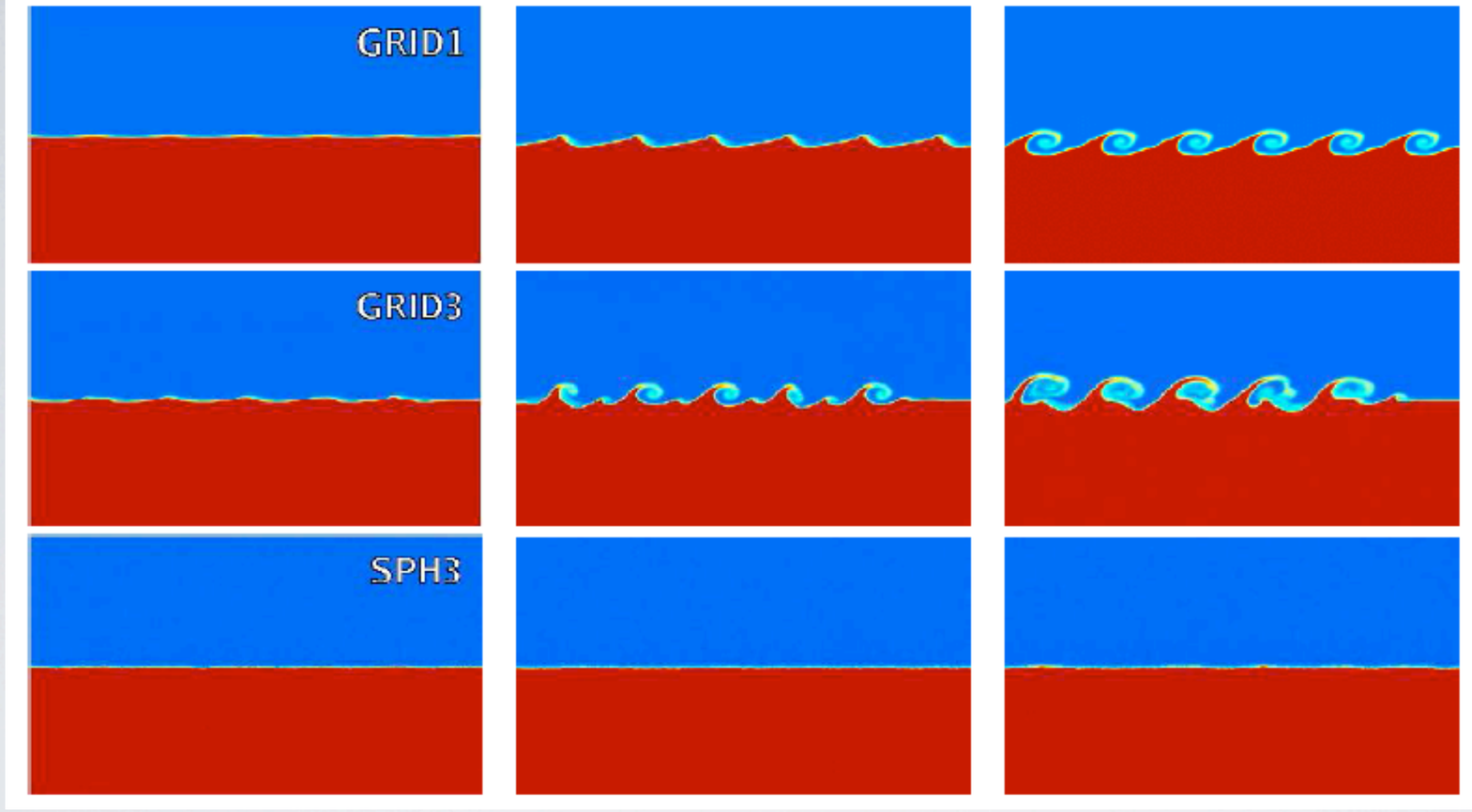
This type of instability, called the Rayleigh-Taylor instability, is common in astrophysics, for example in the solar atmosphere or in expanding supernovae remnants. Linear analysis and Godunov codes agree on the growth rate of the first-order modes (higher order modes cannot be studied analytically and are related to the development of hydrodynamical turbulence)



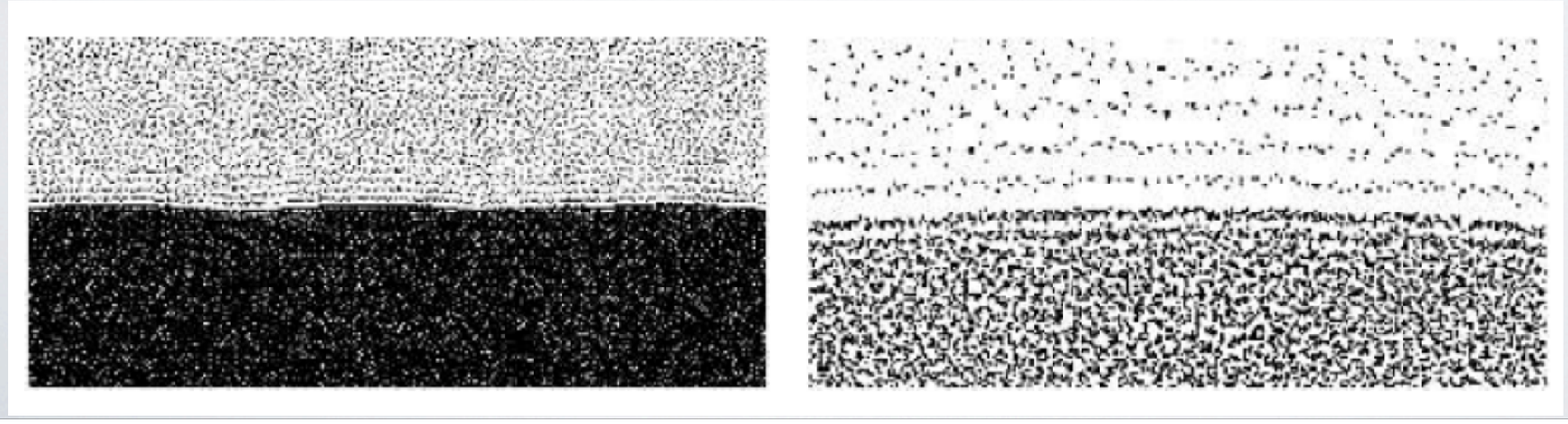
If the two fluids have DeltaV > 0 linear analysis shows that the configuration is unstable no matter the density difference, i.e. a sharp velocity gradient always causes the Kelvin-Helmoltz instability. Again there is agreement between first-order mode growth calculated by perturbation theory and grid-based simulations. Graviy can have a mildly stabilizing effect but cannot suppress the instability.



One would expect that a lagrangian method such as SPH would be very good at computing such an instability since it can simply follow the fluids as they shear past each other rather than having to cope with reference frame issues and related possible advection errors or computationally costly Riemann solver for higher accuracy



Instead standard SPH fails to capture the KH instability due to large errors in the pressure at the contact discontinuity that prevent fluids from really being in contact



We pointed out this problem first in 2007. Since then there has been strong activity worldwide in improving SPH by reducing errors in the pressure estimates. Various formulations and corrections have been proposed which can capture fluid instability and improve results on many fronts. For example, using the lagrangian formulation that we have introduced earlier and the pressure and entropy as the main variables: