Mathematical foundations of relativistic hydrodynamics

José Antonio Font
University of Valencia

NewCompStar School 2017, University of Sofia

Outline

Lecture 1: Relativistic Hydrodynamics

Lecture 2: Numerical methods

R. Leveque, "Nonlinear conservation laws and finite volume methods for astrophysical fluid flow", Springer (1998)

E. Toro, "Riemann solvers and numerical methods for fluid dynamics", Springer (1997)

J.A. Font, "Numerical hydrodynamics and magnetohydrodynamics in general relativity", Living Reviews in Relativity (2008)

L. Rezzolla & O. Zanotti, "Relativistic hydrodynamics", Oxford University Press (2013)

Lecture 2: Numerical methods

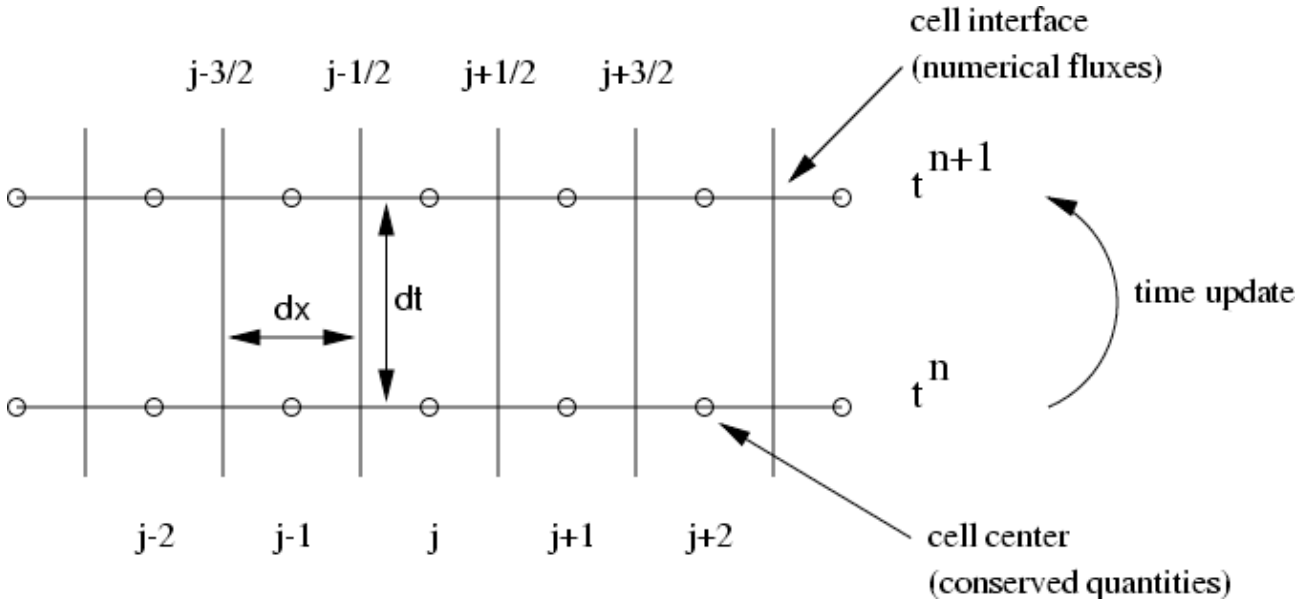
$$\partial_t \mathbf{u} + \partial_i \mathbf{f}^i = \mathbf{s}$$

Partial differential equations (PDEs) are commonly solved numerically by approximating the derivatives with difference operators. Schemes of different orders can be obtained depending on the truncation of the corresponding Taylor series for the derivatives.

Finite-difference schemes are based on a discretization of the x-t plane with a mesh of discrete points (x_j,t^n):

$$x_j = (j - 1/2)\Delta x, \quad t^n = n\Delta t, \quad j = 1, 2, \cdots \quad n = 0, 1, 2, \cdots$$

where Δx and Δt stand for the cell width and time step.



Let us consider the following scalar PDE

$$u_t + f_x = 0\,, \;\; u_0 = u(0,x)\,, \;\; f = f(u)$$

A FD scheme for this eq is a time-marching procedure to obtain approximations to the solution in the mesh points u_j^{n+1} from approximations in the previous time steps u_j^n

We can approximate the time derivative with a first-order forward (Euler) difference

$$u_t = \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

and the spatial derivative with a first-order central difference

$$f_x = \frac{f_{j+1}^n - f_{j-1}^n}{2\Delta x}$$

which yields the explicit first-order central scheme:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}(f_{j+1}^n - f_{j-1}^n)$$

Many other 1st-order and higher-order approximations are available in the literature.

Method of lines (MoL)

Generic name of a family of discretization methods in which space and time variables are dealt with separately (3+1 approach). Time is discretized with finite differences and space discretization is done in various ways (finite differences, finite elements, spectral methods).

The hydro and ADM or BBSN equations can be written in a compact way as a "semi-discrete" system

$$\partial_t \mathbf{u} = \mathbf{S}$$

$\mathbf{u}$ array of dynamical fields

$\mathbf{S}$ remaining term in evolution eqs. including spatial derivatives

PDE "disguised" as an ODE (standard ODE integrators can be applied)

1st-order forward in time (Euler step)

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{S}(t^n, \mathbf{u}^n)$$

2nd-order Runge-Kutta scheme

$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}^n + \Delta t\, \mathbf{S}^n \\ \mathbf{u}^{n+1} &= \frac{1}{2}\mathbf{u}^n + \frac{1}{2}\mathbf{u}^* + \frac{\Delta t}{2}\mathbf{S}^* \end{aligned}$$

higher-order in time ... many schemes available

MoL can be used with any space discretization method. FD is a fairly standard choice.

Field values at grid nodes are represented by the array

$$\mathbf{u}_{i,j,k} = \mathbf{u}(t, x_i, y_j, z_k)$$

Space derivatives:

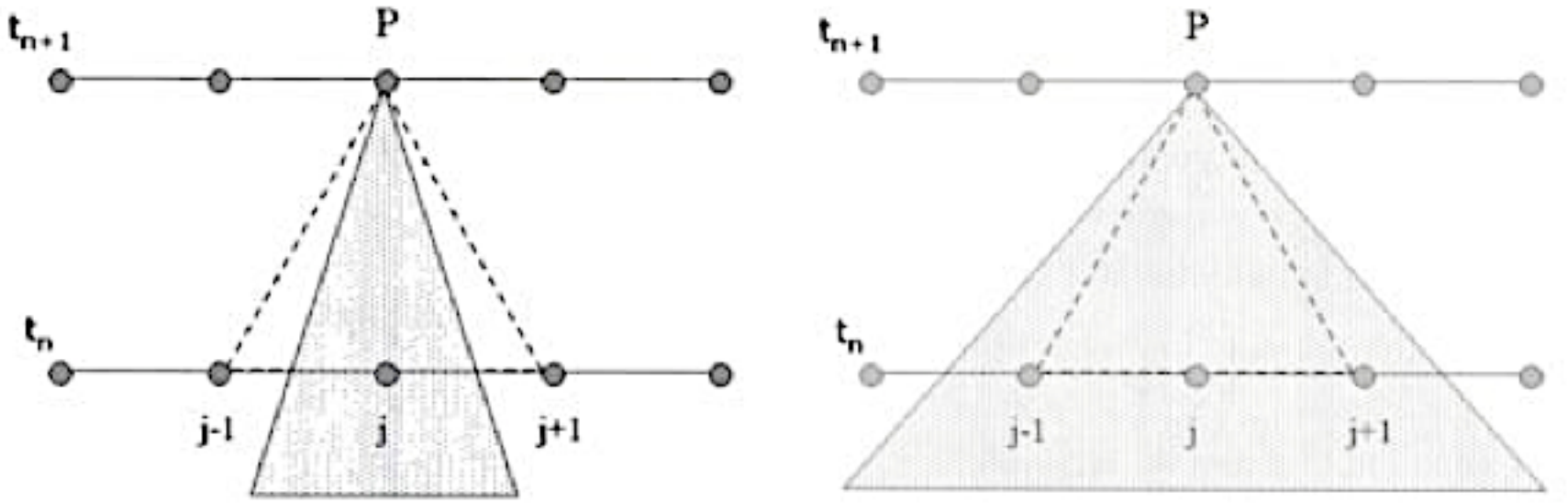
$$\begin{aligned}
2\partial_x u &\sim (u_{i+1,j,k} - u_{i-1,j,k})/\Delta x \\
2\partial_{xx} u &\sim (u_{i+1,j,k} + u_{i-1,j,k} - 2u_{i,j,k})/(\Delta x)^2 \\
2\partial_{xy} u &\sim (u_{i+1,j+1,k} - u_{i-1,j+1,k} \\
&\quad -u_{i+1,j-1,k} + u_{i-1,j-1,k})/(\Delta x \Delta y) \\
&\ldots
\end{aligned}$$

Stencil: set of grid points needed to discretize space derivatives at a given point *P*. Provides the numerical domain of dependence of selected point, i.e. any perturbation at one of the stencil points will change the computed value at *P* after a single time step.

Numerical propagation speed: $v^i_{\text{num}} = s\dfrac{\Delta x^i}{\Delta t}$ s stencil size

Physically, for a system describing wave propagation with some characteristic speeds, the field values at *P* are causally determined by the values inside the past half-cone with vertex at *P*, whose slope is given by the inverse of the largest characteristic speed of the system.

This provides the physical domain of dependence of *P*.



Courant (necessary) condition for numerical stability: the physical domain of dependence of P must be included in the numerical domain of dependence.

$$v_{\max} < n_i \, v^i_{\text{num}}$$

Provides an upper limit for the numerical time step.

The hydro equations involve wave propagation (hyperbolic equations). Remember they can be written in 1st-order quasi-linear form:

$$\partial_t \boldsymbol{U} + \boldsymbol{A}\partial_x \boldsymbol{U} = 0 \qquad \boldsymbol{A}(\boldsymbol{U}) \equiv \partial \boldsymbol{F}/\partial \boldsymbol{U}$$

Jacobian matrix

If Jacobian matrix has constant coefficients (linear case), the solution procedure is simple. First we diagonalize the Jacobian matrix so that

$$\Lambda = \boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{R} \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$$

eigenvectors matrix

eigenvalues matrix

If we define the characteristic variables $\boldsymbol{W} \equiv \boldsymbol{R}^{-1}\boldsymbol{U}$

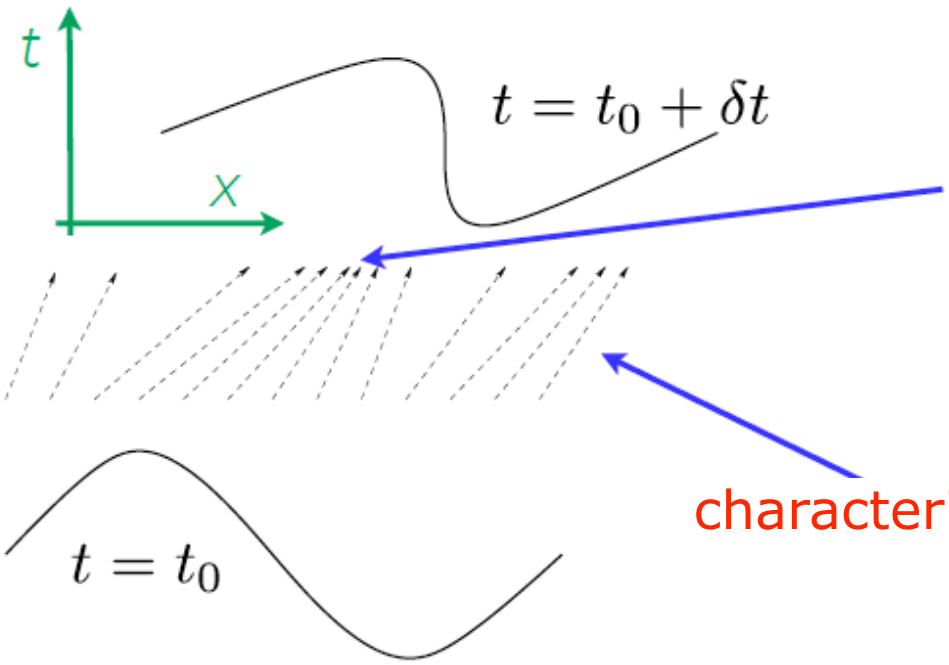
we can decouple the original system of equations:

$$\partial_t \boldsymbol{W} + \Lambda \partial_x \boldsymbol{W} = 0$$

$$\partial_t \bar{w}_i + \Lambda \partial_x \bar{w}_i = 0 \Longleftrightarrow \frac{\mathrm{d}\bar{w}_i}{\mathrm{d}t} = 0 \quad \text{along} \quad \frac{\partial x}{\partial t} = \lambda_i(\boldsymbol{U}(x,t))$$

Therefore, characteristic variables are constant along the curves of the (x,t) plane whose slopes are the corresponding eigenvalue.

Such curves are called characteristic curves and their slopes are given locally by the characteristic speeds.



the characteristic speeds are different and the characteristic curves may "focus"

Since they are constant along the characteristics, the value of the characteristic variables at any given time is known once the initial value is known, that is

$$W^i(x,t) = W^i(x - \lambda_i t, t = 0)$$

Once the solution is known in terms of the characteristic variables, it is straightforward to obtain the solution in terms of the original state vector:

$$\boldsymbol{W} = \boldsymbol{R}^{-1}\boldsymbol{U} \quad \Longrightarrow \quad \boldsymbol{U} = \boldsymbol{R}\boldsymbol{W}$$

$$\boldsymbol{U}(x,t) = \sum_{i=1}^{N} W^i(x,t)\boldsymbol{R}^{(i)} = \sum_{i=1}^{N} W^i(x-\lambda_i t, 0)\boldsymbol{R}^{(i)}$$

Thus, the solution is the linear superposition of *N* waves, each propagating independently of the rest with a speed given by the corresponding eigenvalue of the Jacobian matrix of the system.

The so-called Godunov-type methods extend these concepts to nonlinear hyperbolic equations, solving Riemann problems of a new system of equations obtained by writing the original system as a quasi-linear system. Spectral information of Jacobian matrices is the basis of such solvers, as for linear systems.

Some representative examples: advection equation

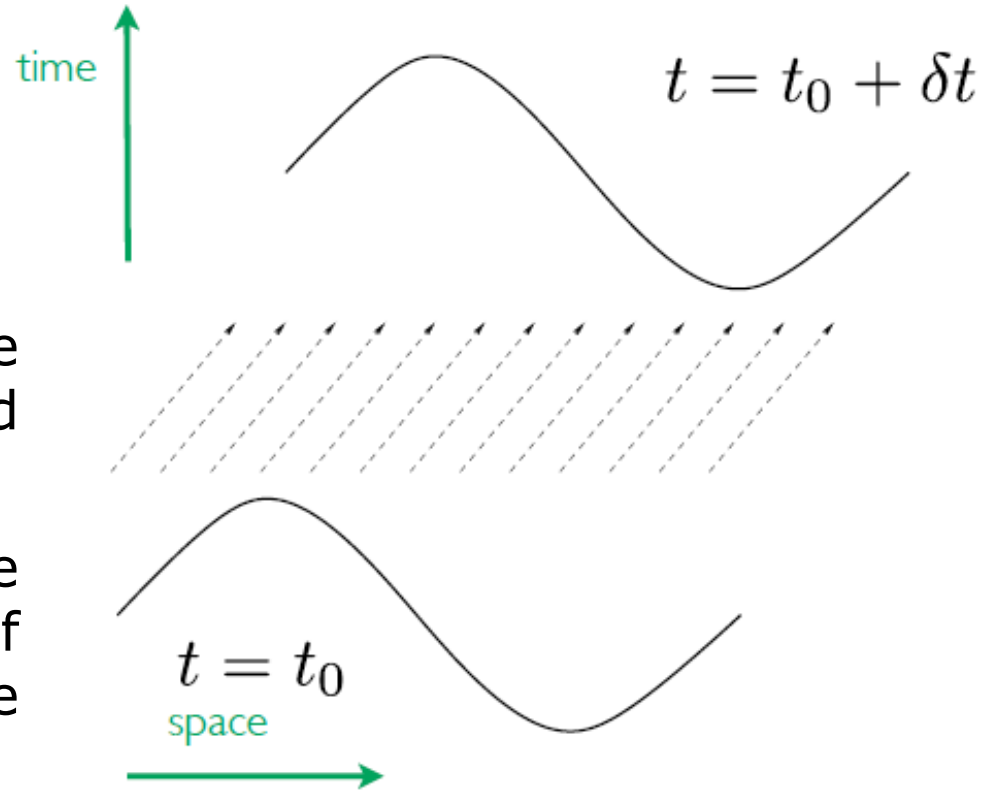
Before discussing the solution of the hydrodynamics equations there are aspects of their nonlinear nature to point out.

The simplest **linear** hyperbolic equation is the advection equation:

$$\partial_t u(x,t) + \partial_x u(x,t) = 0$$

The solution is the initial one simply translated in space and time.

The propagation speeds are constant in every point of space (linear nature of the equation).

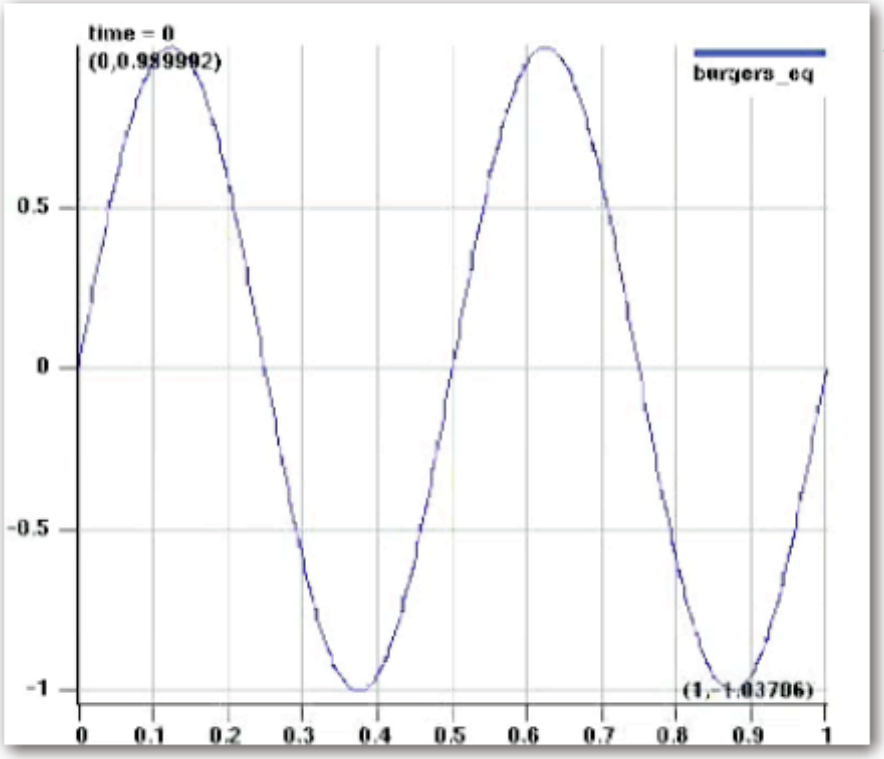


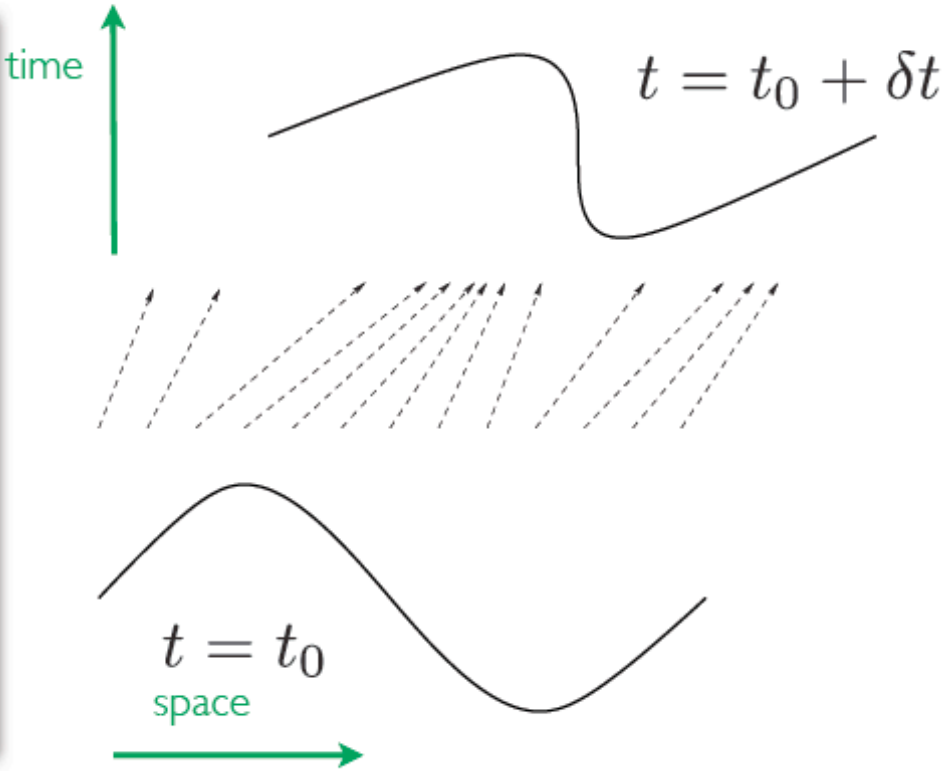
Some representative examples: Burgers equation

The simplest **nonlinear** hyperbolic equation is Burgers equation

$$\partial_t u(x,t) + u(x,t)\partial_x u(x,t) = \epsilon(x,t)\partial_x^2 u(x,t)$$

where the r.h.s. is zero in the inviscid limit. Despite the similarity with the advection equation, its solution is markedly different.





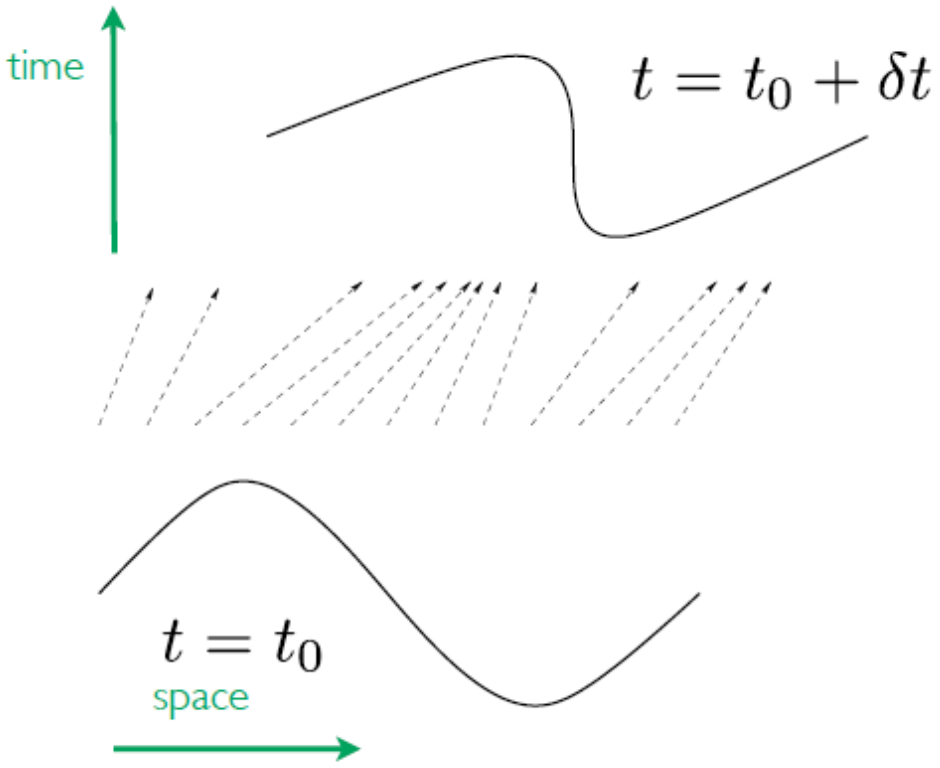
Some representative examples: Burgers equation

This behaviour is known as "shock steepening" and is a consequence of the propagation speeds not being constant, contrary to what happens with the advection equation, but are functions of space and time (nonlinear nature of the equation).

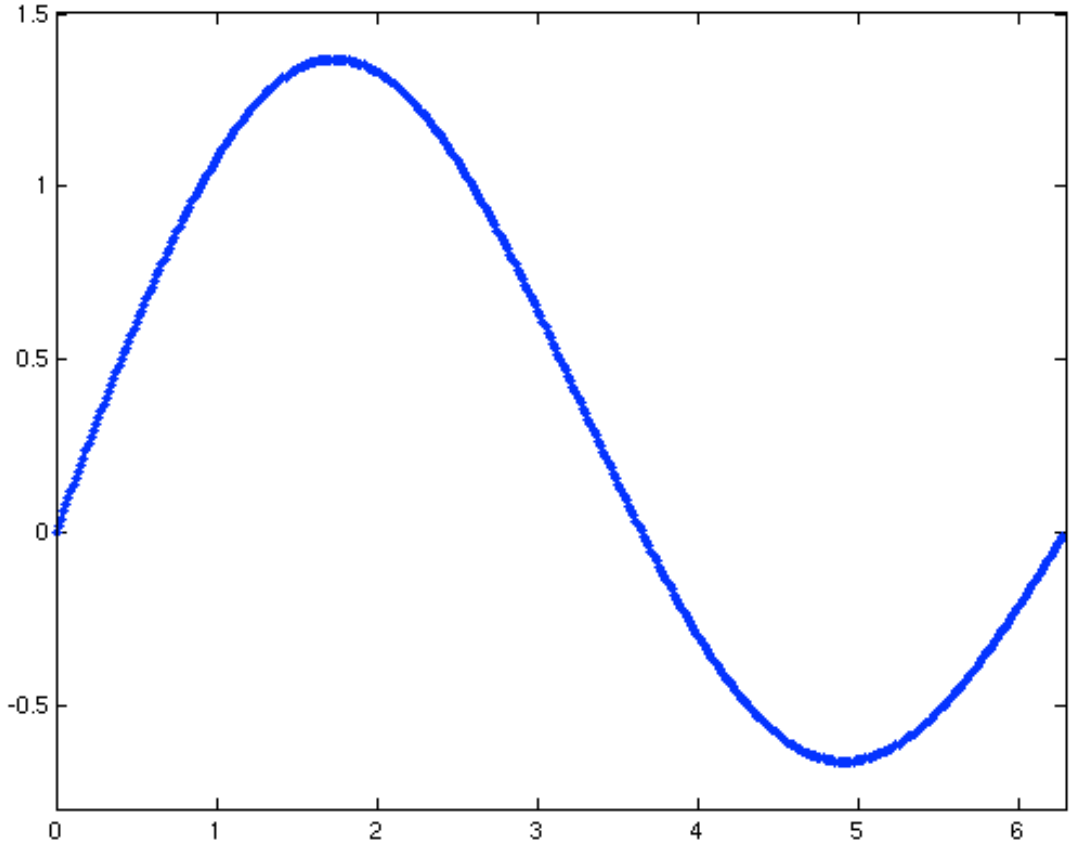
The maxima of the waves move faster than the minima and tend to reach them.

NOTE: this is a property of the equations and not of the initial data.

Even smooth initial data will lead to the appearance of shocks (in t>0) in the case of inviscid fluids.



$$\partial_t u + u\,\partial_x u = 0 \qquad u(x,0) = \sin x + \frac{1}{2}\sin\left(\frac{x}{2}\right)$$



Credit: Balbás & Tadmor.

`CentPack` (high-resolution central schemes)

`http://www.cscamm.umd.edu/centpack`

High-resolution methods: modified high-order finite-difference methods with appropriate amount of numerical dissipation in the vicinity of a discontinuity.

Quantity $\mathbf{u}_j^n$ is an approximation to $\mathbf{u}(x_j, t^n)$ but in the case of a conservation law it is preferable to view it as an approximation to the average within the numerical cell $[x_{j-1/2}, x_{j+1/2}]$

$$\mathbf{u}_j^n \sim \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{u}(x, t^n)\, dx$$

consistent with the integral form of the conservation law

For hyperbolic systems of conservation laws, schemes written in conservation form guarantee that the convergence (if it exists) is to one of the so-called weak solutions of the original system of equations (Lax-Wendroff theorem 1960).

A scheme written in conservation form reads:

$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \frac{\Delta t}{\Delta x}\left(\hat{\mathbf{f}}_{j+\frac{1}{2}}^n - \hat{\mathbf{f}}_{j-\frac{1}{2}}^n\right)$$

where $\hat{\mathbf{f}}$ is the numerical flux function

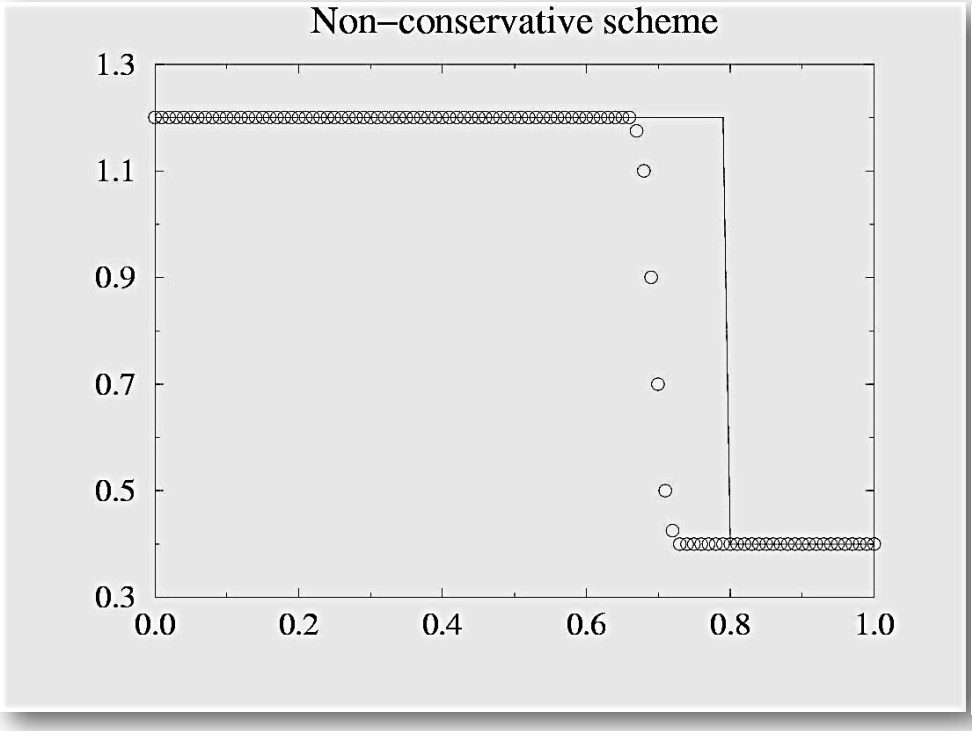
Example: Burgers equation with discontinuous initial data can be discretized using, e.g.

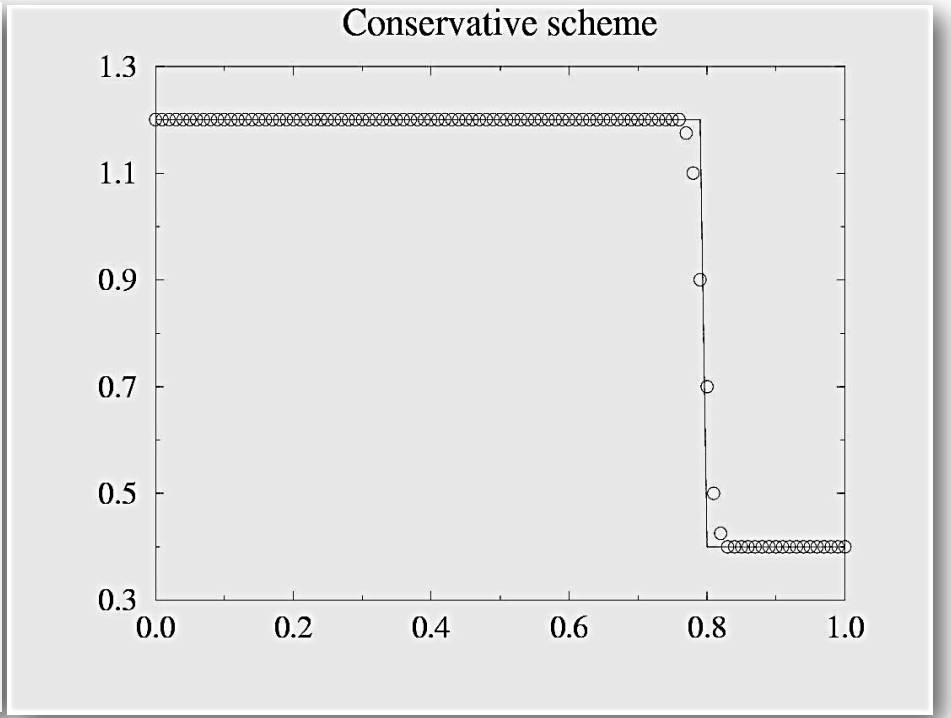
a **conservative upwind scheme:**

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(\frac{1}{2}(u_j^n)^2 - \frac{1}{2}(u_{j-1}^n)^2\right)$$

or using a **non-conservative upwind scheme:**

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} u_j^n (u_j^n - u_{j-1}^n)$$





The conservation form of the scheme is ensured by starting with the integral version of the PDE in conservation form. By integrating the PDE within a spacetime computational cell

$$[x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1}]$$

the numerical flux function is an approximation to the time-averaged flux across the interface:

$$\hat{\mathbf{f}}_{j+1/2} \sim \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{f}(\mathbf{u}(x_{j+1/2}, t))\, dt$$

The flux integral depends on the (unknown) solution at the numerical interfaces during the time step, $\mathbf{u}(x_{j+1/2}, t)$

Key idea (Godunov 1959): a possible procedure is to calculate this solution by solving Riemann problems at every cell interface.

$$\mathbf{u}(x_{j+1/2}, t) = \mathbf{u}(0, \mathbf{u}_j^n, \mathbf{u}_{j+1}^n)$$

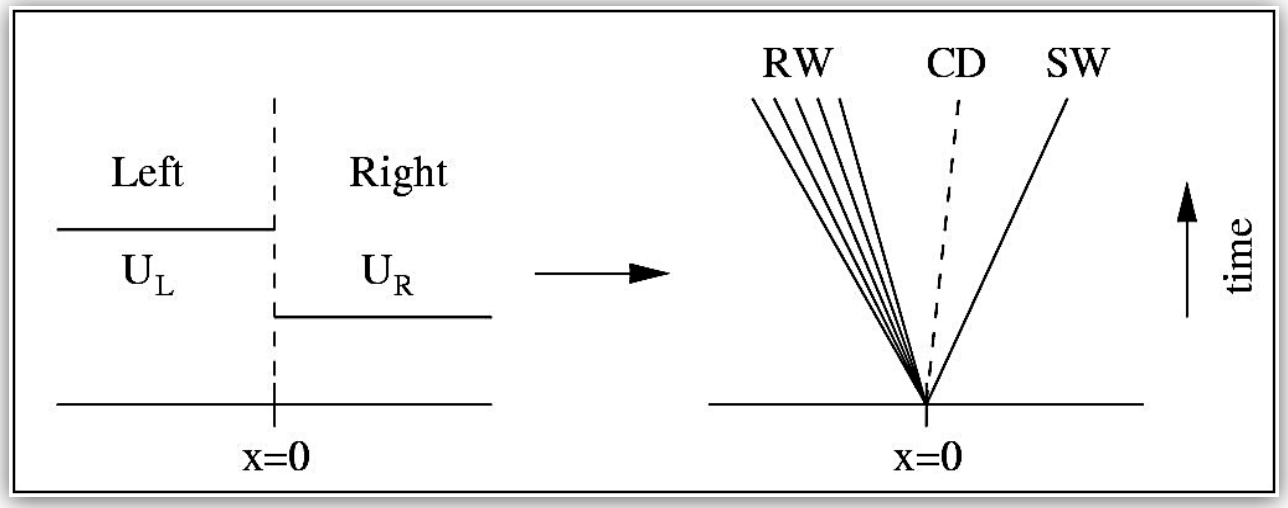
Riemann solution for the left and right states along the ray x/t=0.

The Riemann problem

A Riemann problem is an IVP with discontinuous initial data:

$$\mathbf{u}_0 = \begin{cases} \mathbf{u}_L & \text{if } x < 0 \\ \mathbf{u}_R & \text{if } x > 0 \end{cases}$$

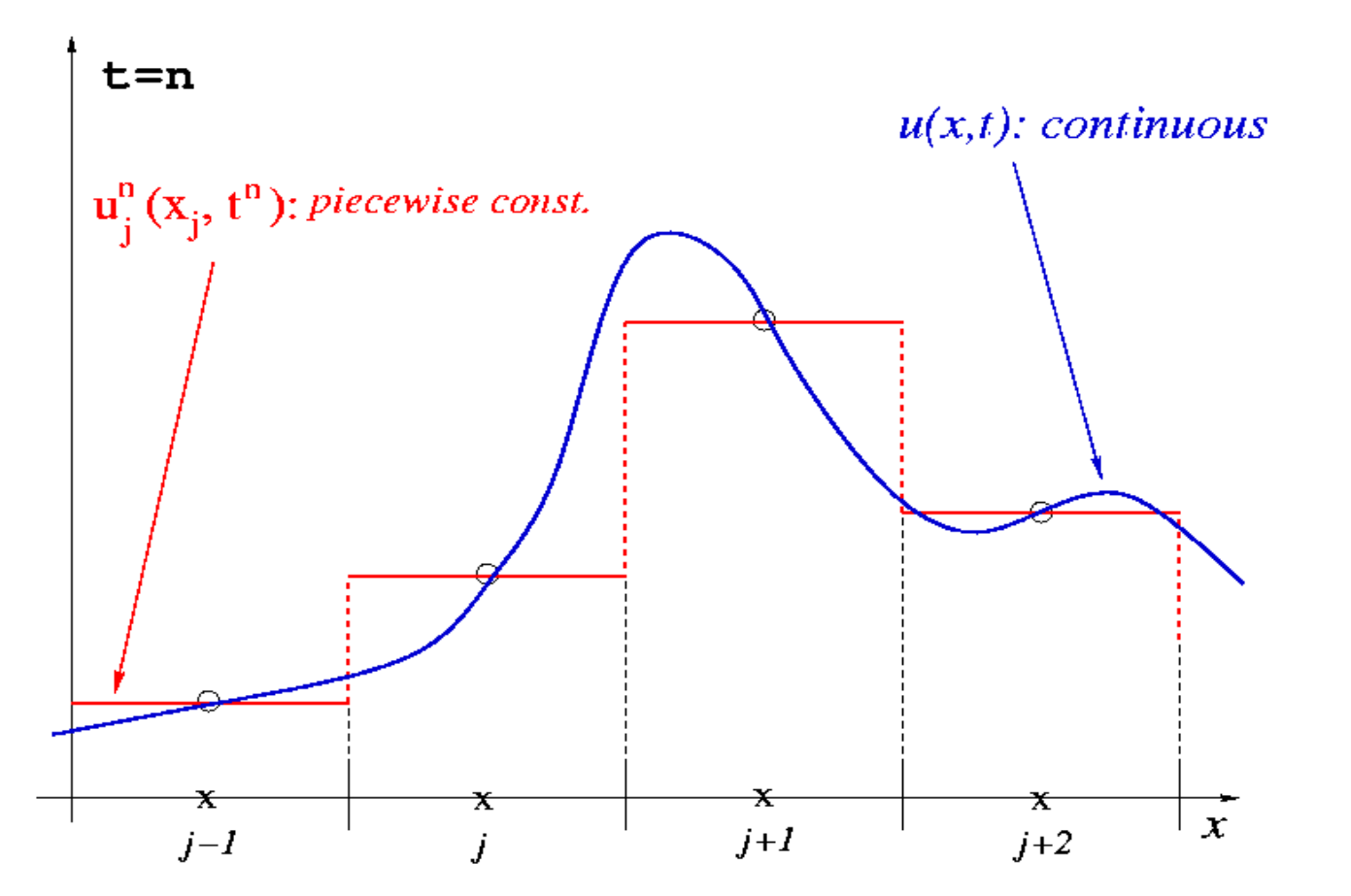
The solution is constant along the straight lines x/t=constant, and, hence, self-similar. It consists of **constant states separated by rarefaction waves** (continuous self-similar solutions of the differential equations), **shock waves, and contact discontinuities** (Lax 1972).



The incorporation of the **exact solution** of Riemann problems to compute the **numerical fluxes** of Euler's equations is due to **Godunov** (1959)

Why Riemann problems?

When a Cauchy problem described by a set of continuous PDEs is solved in a **discretized form** the numerical solution is **piecewise constant** (collection of local Riemann problems).

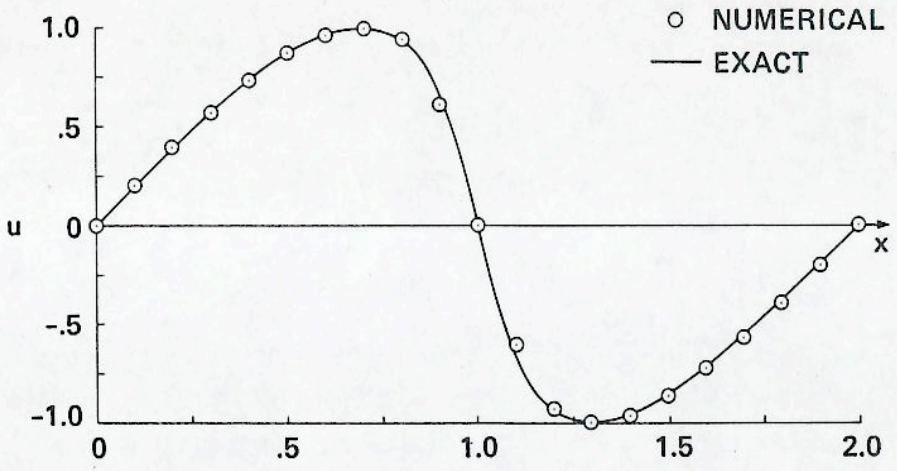


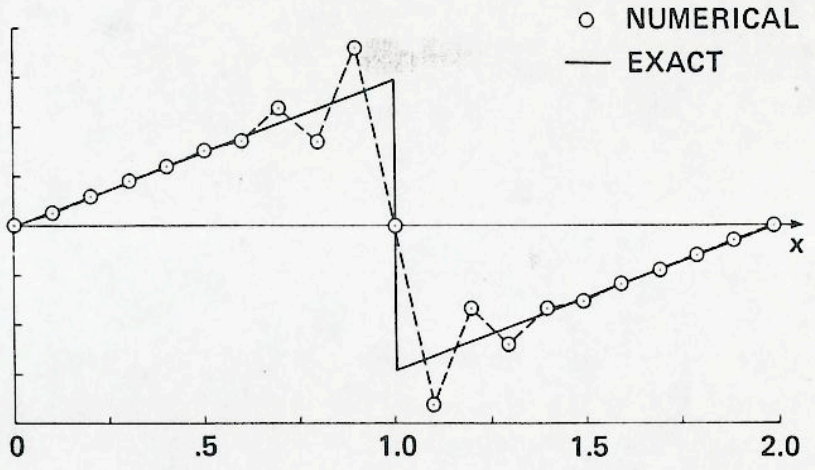
This is particularly problematic when solving the hydrodynamic equations (either Newtonian or relativistic) for compressible fluids.

Their hyperbolic, nonlinear character produces discontinuous solutions in a finite time (shock waves, contact discontinuities) even from smooth initial data!

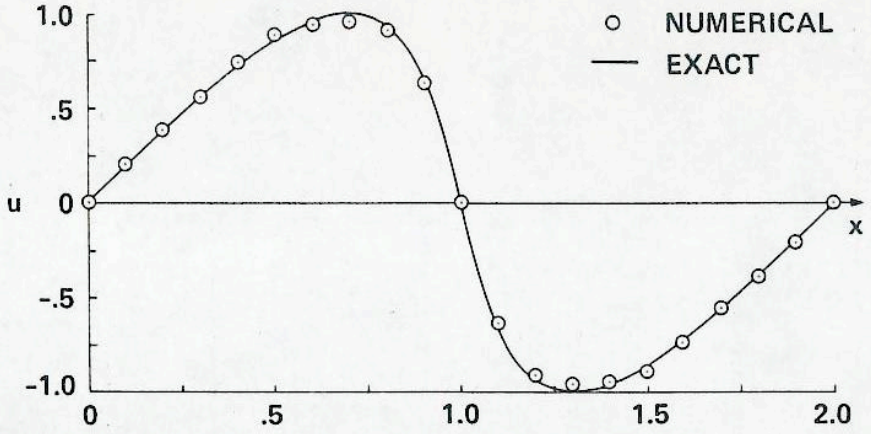
Any numerical scheme must be able to handle discontinuities in a satisfactory way.

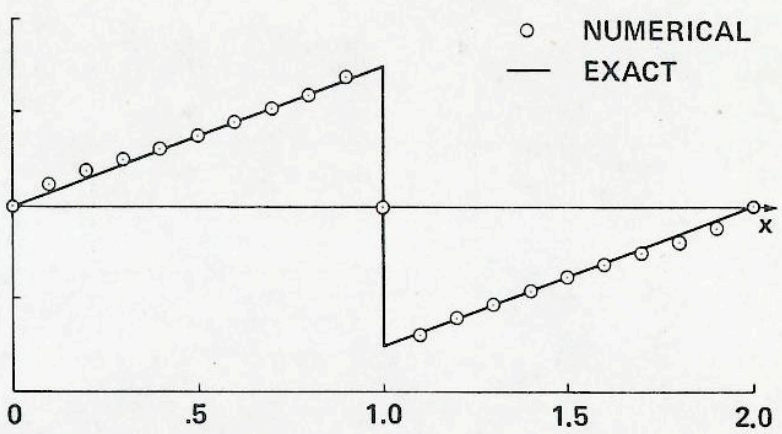
1. 1st order accurate schemes (Lax-Friedrich): Non-oscillatory but inaccurate across discontinuities (excessive diffusion)
2. (standard) 2nd order accurate schemes (Lax-Wendroff): Oscillatory across discontinuities
3. 2nd order accurate schemes with artificial viscosity
4. Godunov-type schemes (upwind High Resolution Shock Capturing schemes)





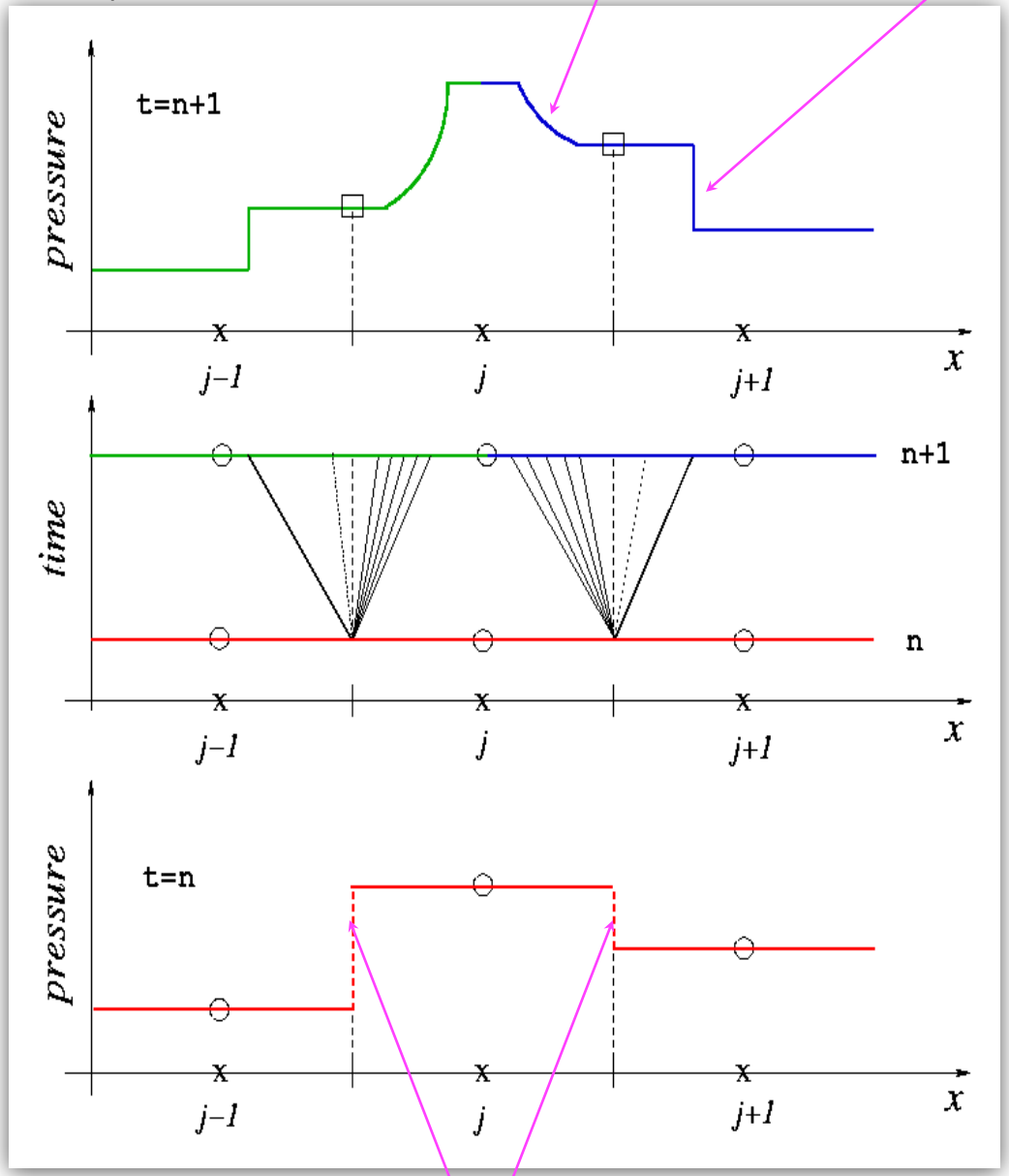
Lax-Wendroff numerical solution of Burger's equation at t=0.2 (left) and t=1.0 (right)





2nd order TVD numerical solution of Burger's equation at t=0.2 (left) and t=1.0 (right)

courtesy of L. Rezzolla



Solution at time n+1 of the two Riemann problems at the cell boundaries $x_{j+1/2}$ and $x_{j-1/2}$

Spacetime evolution of the two Riemann problems at the cell boundaries $x_{j+1/2}$ and $x_{j-1/2.}$ Each problem leads to a shock wave and a rarefaction wave moving in opposite directions

Initial data at time n for the two Riemann problems at the cell boundaries $x_{j+1/2}$ and $x_{j-1/2}$

$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \frac{\Delta t}{\Delta x}\left(\hat{\mathbf{f}}_{j+\frac{1}{2}}^n - \hat{\mathbf{f}}_{j-\frac{1}{2}}^n\right)$$

Approximate Riemann solvers

In Godunov's method the structure of the Riemann solution is "lost" in the cell averaging process (1st order in space).

The exact solution of a RP is computationally expensive, particularly in multidimensions and for complicated EoS.

This motivated development of approximate (linearized) Riemann solvers.

Based on the exact solution of RP corresponding to a new system of equations obtained by a linearization of the original one (quasi-linear form). The spectral decomposition of the Jacobian matrices is on the basis of all solvers ("extending" ideas for linear systems).

$$\frac{\partial \vec{u}}{\partial t} + \frac{\partial \vec{f}}{\partial x} = 0 \Rightarrow \frac{\partial \vec{u}}{\partial t} + A\frac{\partial \vec{u}}{\partial x} = 0\,, \quad A = \frac{\partial \vec{f}}{\partial \vec{u}}$$

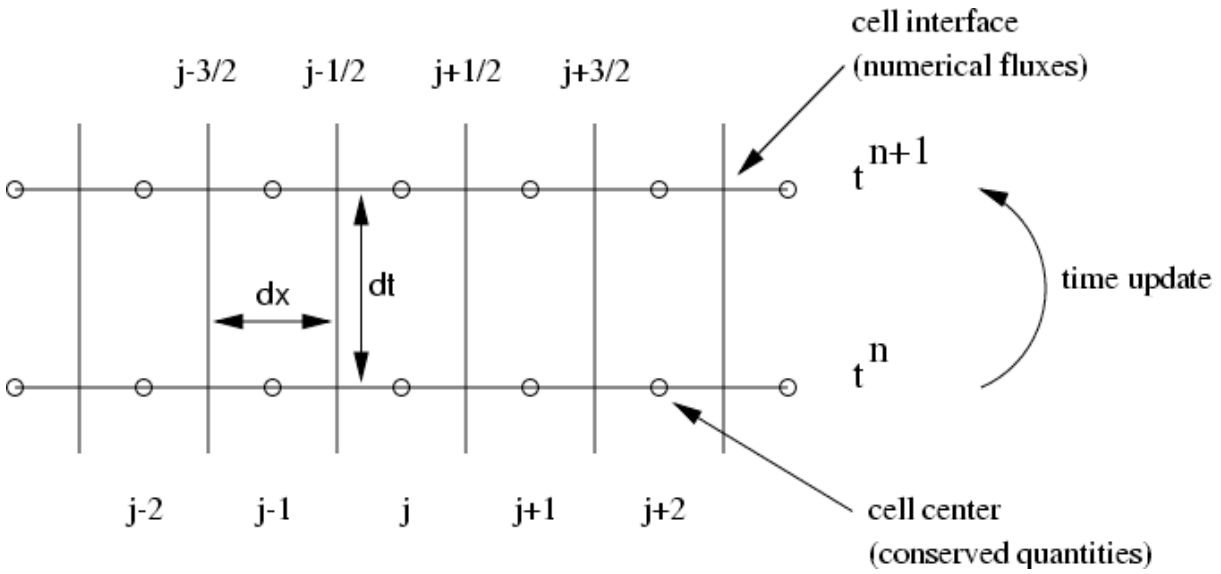
Approach followed by a subset of shock-capturing schemes, the so-called **Godunov-type methods** (Harten & Lax 1983; Einfeldt 1988).

See Martí & Müller (2003) for comprehensive discussion of approximate Riemann solvers for relativistic hydrodynamics.

Standard implementation of a HRSC scheme

1. Time update (MoL):

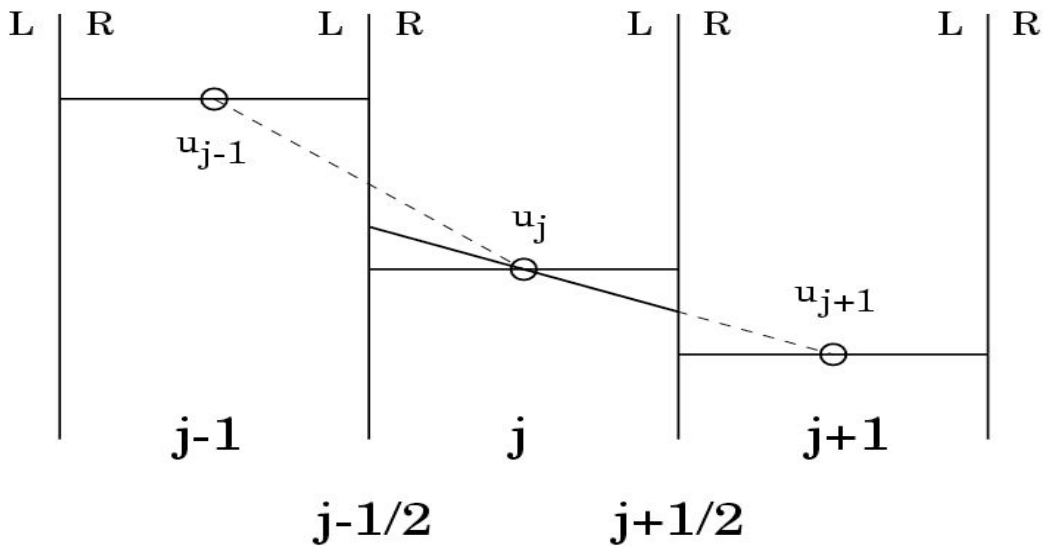
Algorithm in conservation form



$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \frac{\Delta t}{\Delta x}\left(\hat{\mathbf{f}}^n_{j+\frac{1}{2}} - \hat{\mathbf{f}}^n_{j-\frac{1}{2}}\right) + \Delta t\ \mathbf{S}_j^n$$

In practice: 2nd or 3rd order time accurate, conservative Runge-Kutta schemes (Shu & Osher 1989; MoL)

2. Cell reconstruction: Piecewise constant (Godunov), linear (MUSCL, MC, van Leer), parabolic (PPM, Colella & Woodward) interpolation procedures of state-vector variables from cell centers to cell interfaces.



3. Numerical fluxes: Approximate Riemann solvers (Roe, HLLE, Marquina). Explicit use of the spectral information of the system.

$$\hat{\mathbf{f}}_j = \frac{1}{2}\left[\mathbf{f}_j(\mathbf{w}_\mathrm{R}) + \mathbf{f}_j(\mathbf{w}_\mathrm{L}) - \sum_{i=1}^{5} |\tilde{\lambda}_i|\, \Delta\tilde{\mathbf{w}}_i\, \tilde{R}_i\right]$$

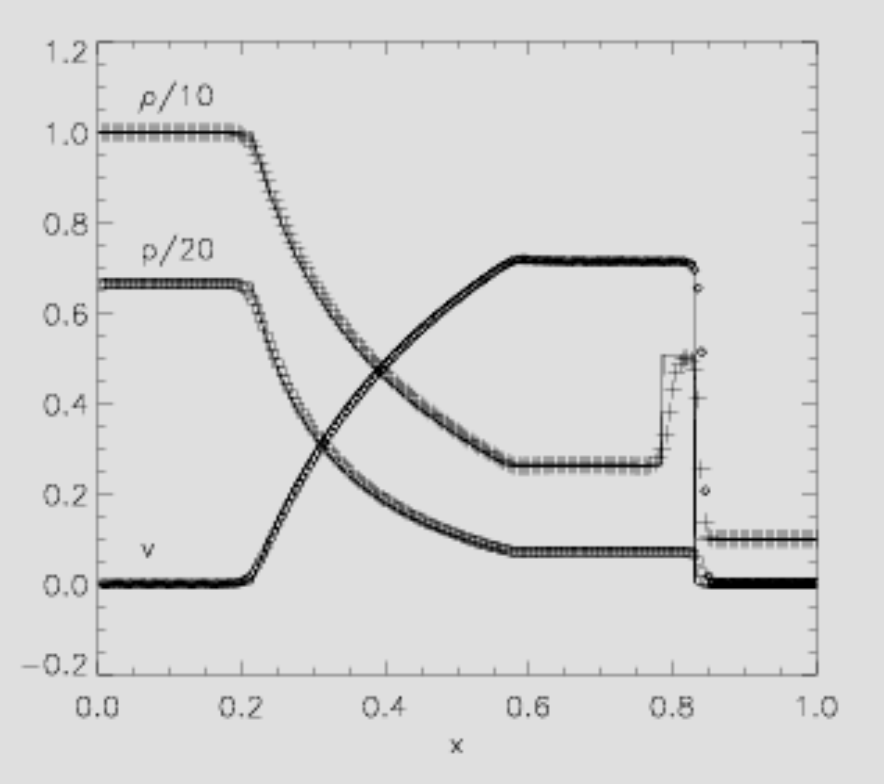
$$\Delta\mathbf{u} \equiv \mathbf{u}(\mathbf{w}_\mathrm{R}) - \mathbf{u}(\mathbf{w}_\mathrm{L}) = \sum_{i=1}^{5} \Delta\tilde{\mathbf{w}}_i\, \tilde{R}_i$$

High-resolution shock-capturing schemes

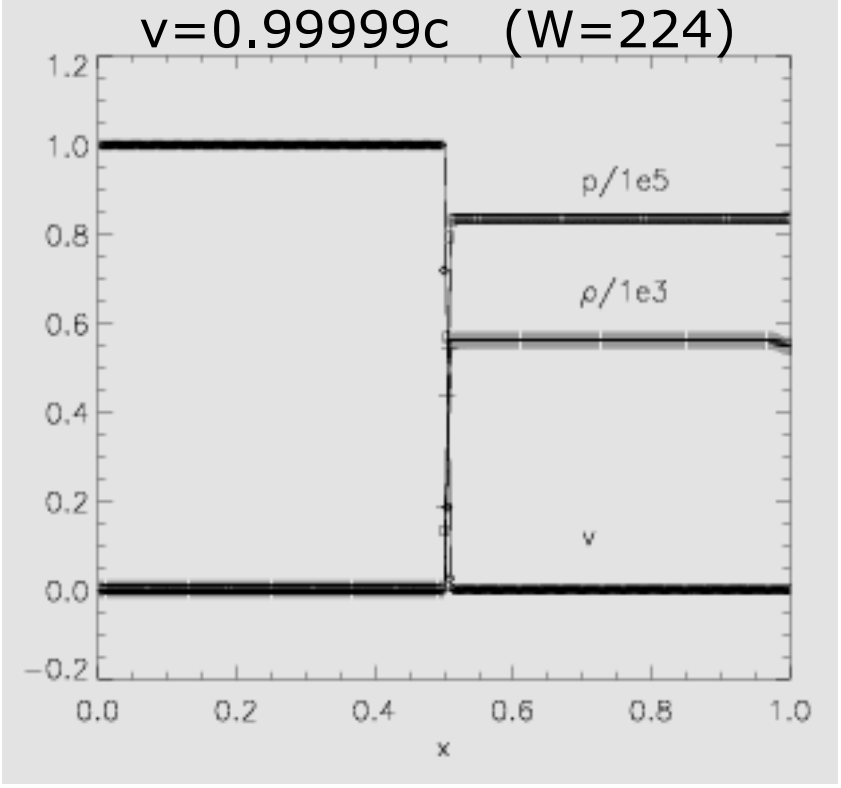
- **Stable and accurate shock profiles**
- **Accurate propagation speed of discontinuities**
- **Accurate numerical resolution of nonlinear features:**

discontinuites, rarefaction waves, vortices, turbulence, etc

Shock tube test



Relativistic shock reflection



Exercise: download the open source codes from the Living Review article by Martí & Müller (2003) and use it to solve various shock tube problems with the special relativistic hydrodynamics equations.

"Numerical Hydrodynamics in Special Relativity"
José Maria Martí and Ewald Müller
http://www.livingreviews.org/lrr-2003-7

Program RIEMANN

This program computes the solution of a 1D relativistic Riemann problem.

Program rPPM

This program simulates 1D relativistic flows in Cartesian coordinates using the exact Riemann solver and PPM reconstruction.