Basic Discretization Methods in Astrophysical Fluid Dynamics

ANDREA MIGNONE

DIPARTIMENTO DI FISICA
UNIVERSITÀ DI TORINO (ITALY)

Outline

1. ***Fundamental Equations and Motivations;***

2. ***Discretization Methods for Hyperbolic PDE:***

   a. *Finite difference and finite volume methods*
   b. *Scalar advection equation*
   c. *Systems of linear equations*
   d. *Nonlinear equation*
   e. *Extension to the Euler equations and MHD*
   f. *High-order spatial and temporal accuracy;*

1. INTRODUCTION

Plasma Description

- Most theoretical models are based on a fluid description ($L \gg \lambda_{mfp}$) requiring the solution of highly nonlinear hyperbolic / parabolic P.D.E., e.g.

Euler equations

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) &= 0 \\
\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{v} \boldsymbol{v}^T\right) + \nabla p &= \rho \boldsymbol{a} + \nabla \cdot \Pi \\
\frac{\partial E}{\partial t} + \nabla \cdot \left[(E + p)\, \boldsymbol{v}\right] &= \rho \boldsymbol{v} \cdot \boldsymbol{a} + \nabla \cdot (\boldsymbol{v} \cdot \Pi) + \nabla \cdot \boldsymbol{F}_c
\end{aligned}
$$

Single Fluid MHD equations

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) &= 0 \\
\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot \left[\rho \boldsymbol{v} \boldsymbol{v}^T - \boldsymbol{B}\boldsymbol{B}^T\right] + \nabla \left(p + \frac{\boldsymbol{B}^2}{2}\right) &= \rho \boldsymbol{a} + \nabla \cdot \Pi \\
\frac{\partial \boldsymbol{B}}{\partial t} - \nabla \times (\boldsymbol{v} \times \boldsymbol{B}) &= -\nabla \times (\eta \boldsymbol{J}) \\
\frac{\partial E}{\partial t} + \nabla \cdot \left[(E + p_T)\, \boldsymbol{v} - (\boldsymbol{B} \cdot \boldsymbol{v})\, \boldsymbol{B}\right] &= \rho \boldsymbol{v} \cdot \boldsymbol{a} - \nabla \cdot (\eta \boldsymbol{J} \times \boldsymbol{B}) + \nabla \cdot (\boldsymbol{v} \cdot \Pi) + \nabla \cdot \boldsymbol{F}_c
\end{aligned}
$$

Why Numerical Simulations ?

- Exact solutions possible under very restrictive assumptions, e.g. stationarity ($\partial/\partial t = 0$), self-similarity, spherically symmetry or similar.

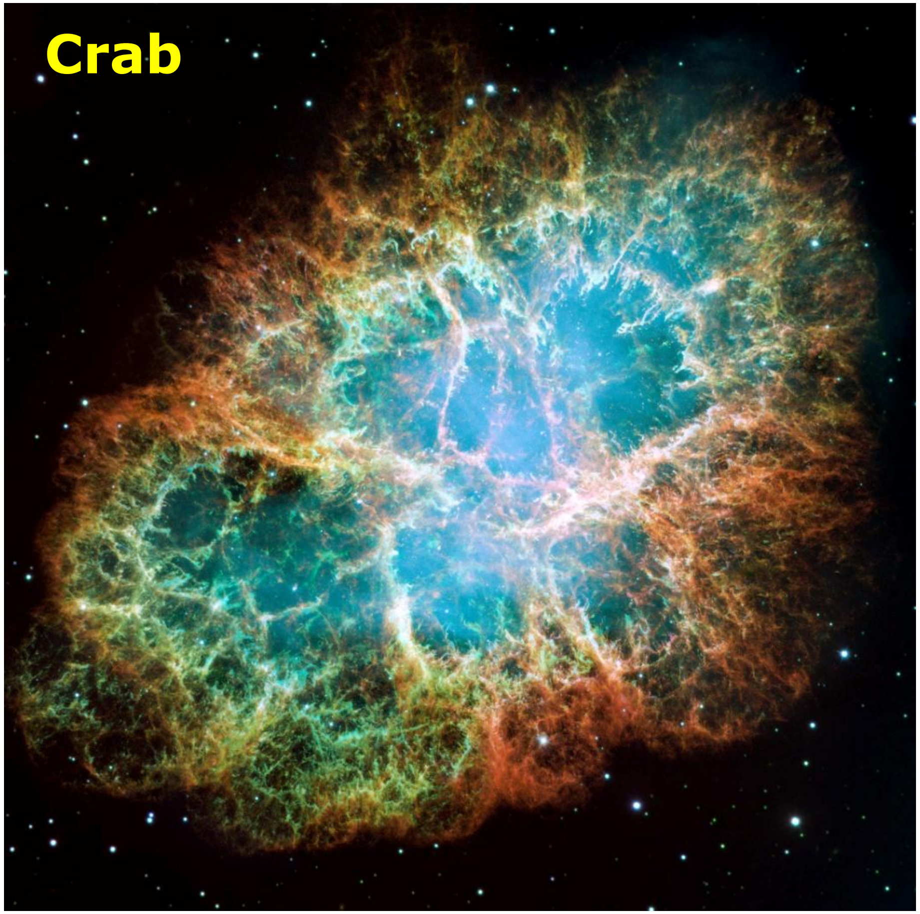
- Nonlinear, time-dependent systems can be studied only by means of numerical simulations.

- Grid-Based fluid approach via Finite Volume/Difference:
  - Fluid variables are discretized on a spatial grid (static or adaptive) and evolved in time.
  - Numerical solution of hyperbolic PDE in presence of discontinuous waves
  - Shock-Capturing (or Godunov-type) schemes.

A computational example:
Rayleigh-Taylor unstable flows

- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

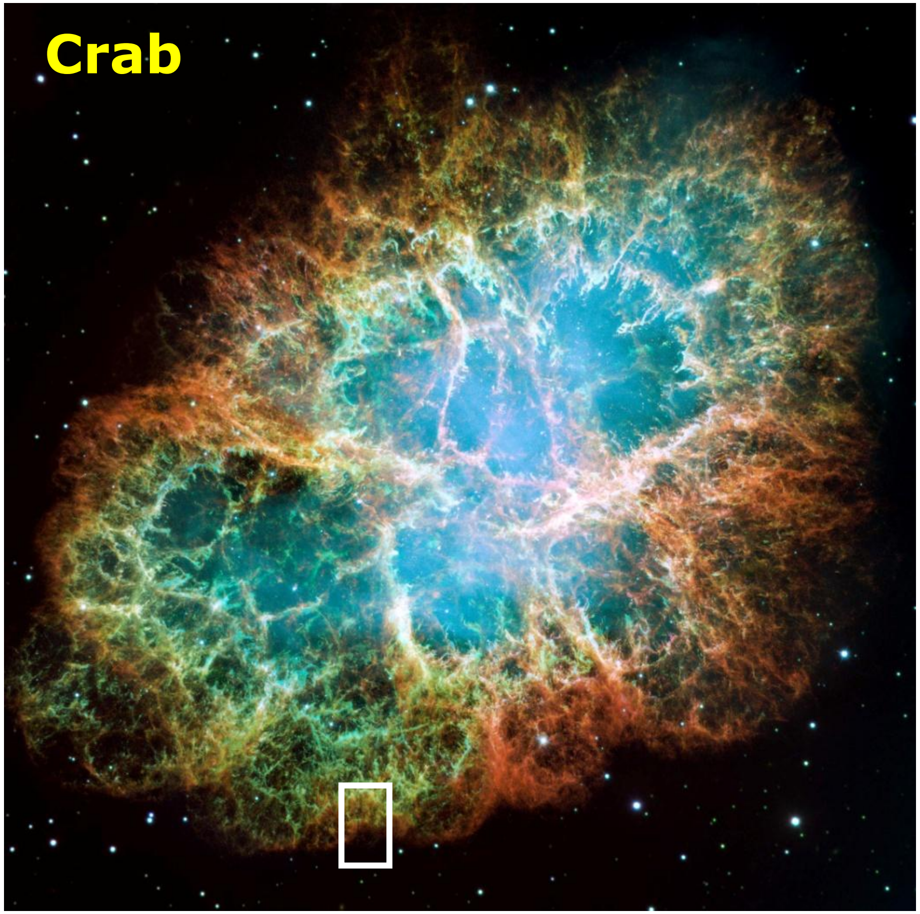


A computational example:
Rayleigh-Taylor unstable flows

- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

- Choose computational domain

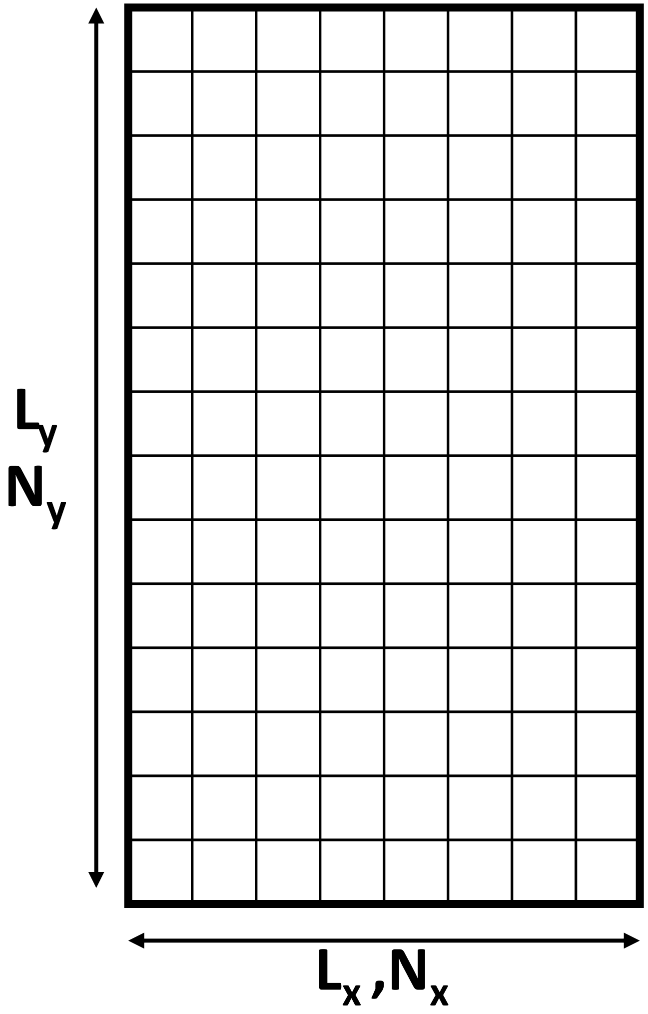


A computational example:
Rayleigh-Taylor unstable flows

- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

- Choose computational domain
- Set the number of zones



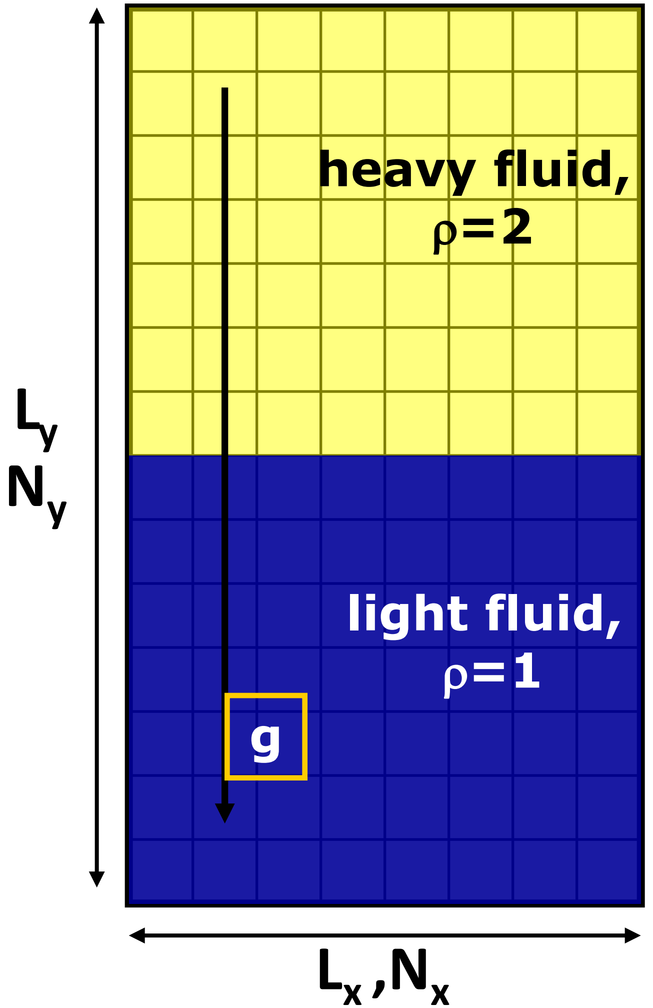
A computational example:
Rayleigh-Taylor unstable flows

- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

- Choose computational domain
- Set the number of zones
- Set initial conditions:

$$\rho = \begin{cases} 2 & \text{for} \quad y > 0 \\ 1 & \text{for} \quad y < 0 \end{cases} \qquad p = \frac{1}{\gamma} - \rho g y\,, \quad v_x = 0\,, \quad v_y = \epsilon R \left[1 + \cos\left(\frac{2\pi y}{L_y}\right)\right]$$



A computational example:
Rayleigh-Taylor unstable flows

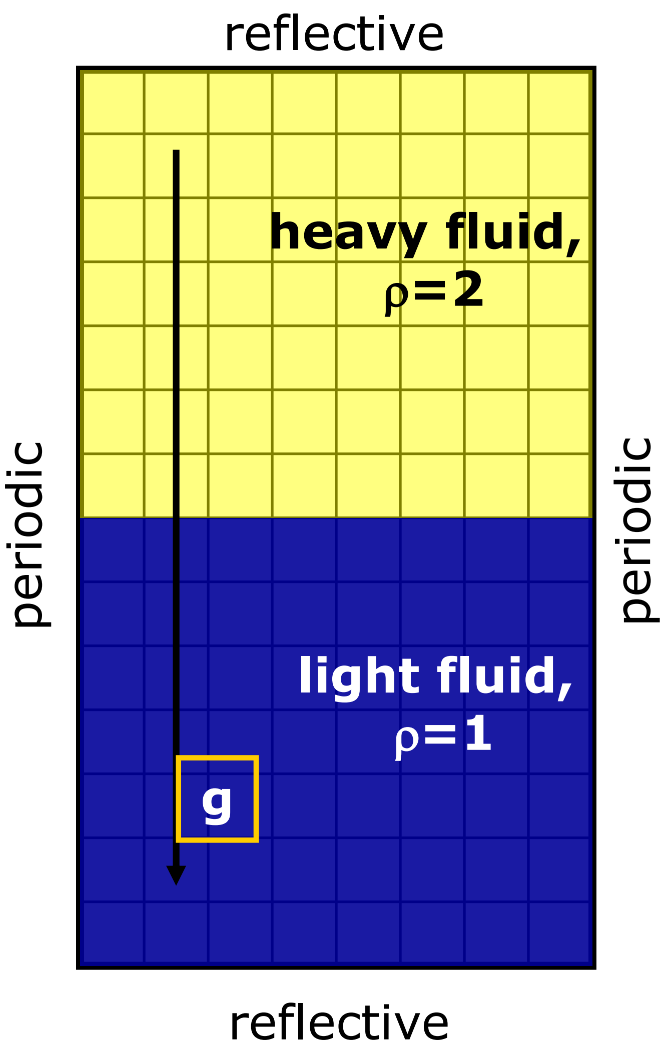
- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

- Choose computational domain
- Set the number of zones
- Set initial conditions:

$$\rho = \begin{cases} 2 & \text{for} \quad y > 0 \\ 1 & \text{for} \quad y < 0 \end{cases} \qquad p = \frac{1}{\gamma} - \rho g y\,, \quad v_x = 0\,, \quad v_y = \epsilon R\left[1 + \cos\left(\frac{2\pi y}{L_y}\right)\right]$$

- Set boundary conditions



A computational example:
Rayleigh-Taylor unstable flows

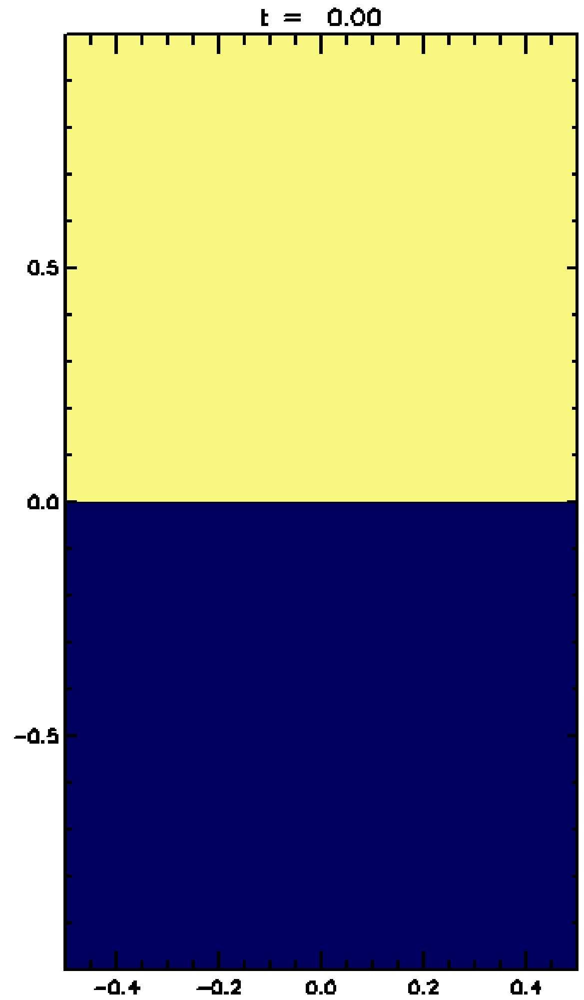
- ***Problem:***

  Supernova remnants morphology & Rayleigh Taylor Instability

- Choose computational domain
- Set the number of zones
- Set initial conditions:

$$\rho = \begin{cases} 2 & \text{for} \quad y > 0 \\ 1 & \text{for} \quad y < 0 \end{cases} \qquad p = \frac{1}{\gamma} - \rho g y\,, \quad v_x = 0\,, \quad v_y = \epsilon R\left[1 + \cos\left(\frac{2\pi y}{L_y}\right)\right]$$

- Set boundary conditions
- Set final integration time & Run!



2a. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE

FINITE DIFFERENCE

AND

FINITE VOLUME METHODS

Numerical Discretizations

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0$$

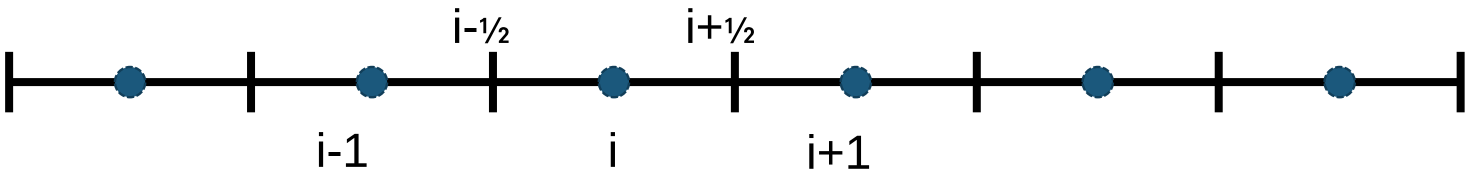
- Two popular methods for performing discretization:

  - *Finite Differences* (FD);
  - *Finite Volume* (FV);

- For some problems, the resulting discretizations look identical, but they are distinct approaches;
- We begin using finite-difference as it will allow to quickly learn some important concepts.

Finite Difference Approach

- A finite-difference method stores the solution at specific points in space and time;



- Associated with each grid point is a function value,

$$q_i^n = q(x_i, t^n)$$

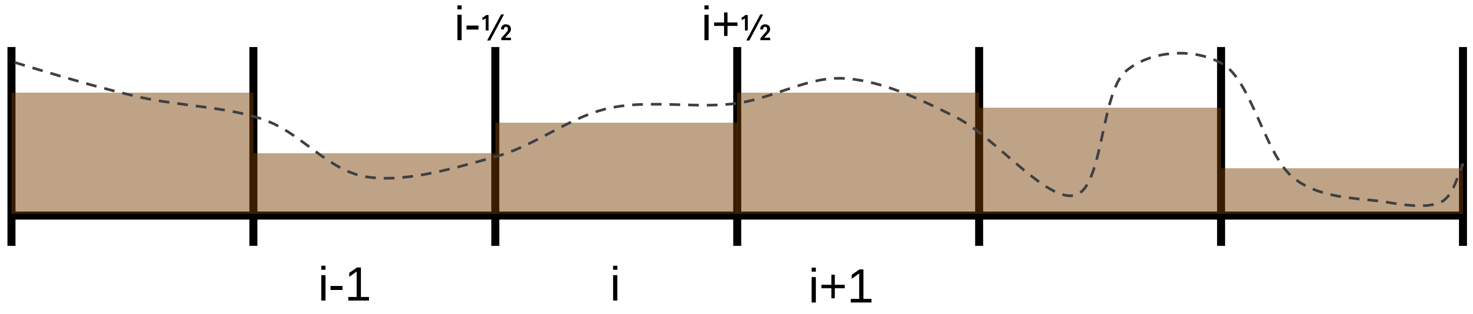
- We replace the derivatives in our PDE with differences between neighbor points

Finite Volume Approach

- In a finite volume discretization, the unknowns are the spatial averages of the function itself:

$$\langle \boldsymbol{q} \rangle_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \boldsymbol{q}(x,t)\, dx$$

where $x_{i-½}$ and $x_{i+½}$ denote the location of the cell interfaces.



- The solution to the conservation law involves computing fluxes through the boundary of the control volumes

Finite Volume Formulation

- The *conservative form* of the equations provides the link between the *differential* form of the equation,

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0$$

and the *integral* form, obtained by integrating the equations over a time interval $\Delta t = t^{n+1} - t^{n}$ and cell size $\Delta x = x_{i+1/2} - x_{i-1/2}$

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} \right) dt\, dx$$

Finite Volume Formulation

- Performing the spatial integration yields

$$\int_{t^n}^{t^{n+1}} \left[\Delta x \frac{d}{dt} \langle \boldsymbol{q} \rangle_i + \left(\boldsymbol{F}_{i+\frac{1}{2}} - \boldsymbol{F}_{i-\frac{1}{2}} \right) \right] dt = 0$$

with $\langle \boldsymbol{q} \rangle_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \boldsymbol{q}(x,t)\, dx$ being a spatial average.

- A second integration in time gives

$$\Delta x \left(\langle \boldsymbol{q} \rangle_i^{n+1} - \langle \boldsymbol{q} \rangle_i^{n} \right) + \Delta t \left(\tilde{\boldsymbol{F}}_{i+\frac{1}{2}}^{n} - \tilde{\boldsymbol{F}}_{i-\frac{1}{2}}^{n} \right) = 0$$

where $\tilde{\boldsymbol{F}}_{i\pm\frac{1}{2}}^{n} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \boldsymbol{F}\left(\boldsymbol{q}(x_{i\pm\frac{1}{2}}, t) \right) dt$ is a temporal average

Finite Volume Formulation

- Rearranging terms yields

$$\langle \boldsymbol{q} \rangle_i^{n+1} = \langle \boldsymbol{q} \rangle_i^n - \frac{\Delta t}{\Delta x} \left(\tilde{\boldsymbol{F}}_{i+\frac{1}{2}}^n - \tilde{\boldsymbol{F}}_{i-\frac{1}{2}}^n \right)$$

Integral form

with spatial and temporal averages given by

$$\langle \boldsymbol{q} \rangle_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \boldsymbol{q}(x,t)\, dx \qquad \tilde{\boldsymbol{F}}_{i\pm\frac{1}{2}}^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \boldsymbol{F}\Big(\boldsymbol{q}(x_{i\pm\frac{1}{2}}, t) \Big)\, dt$$

- This is an *EXACT* evolutionary equation for the spatial averages of *q*.
- This relation provides an *integral* representation of the original differential equation.
- The integral form does not make use of partial derivatives!

The Riemann Problem

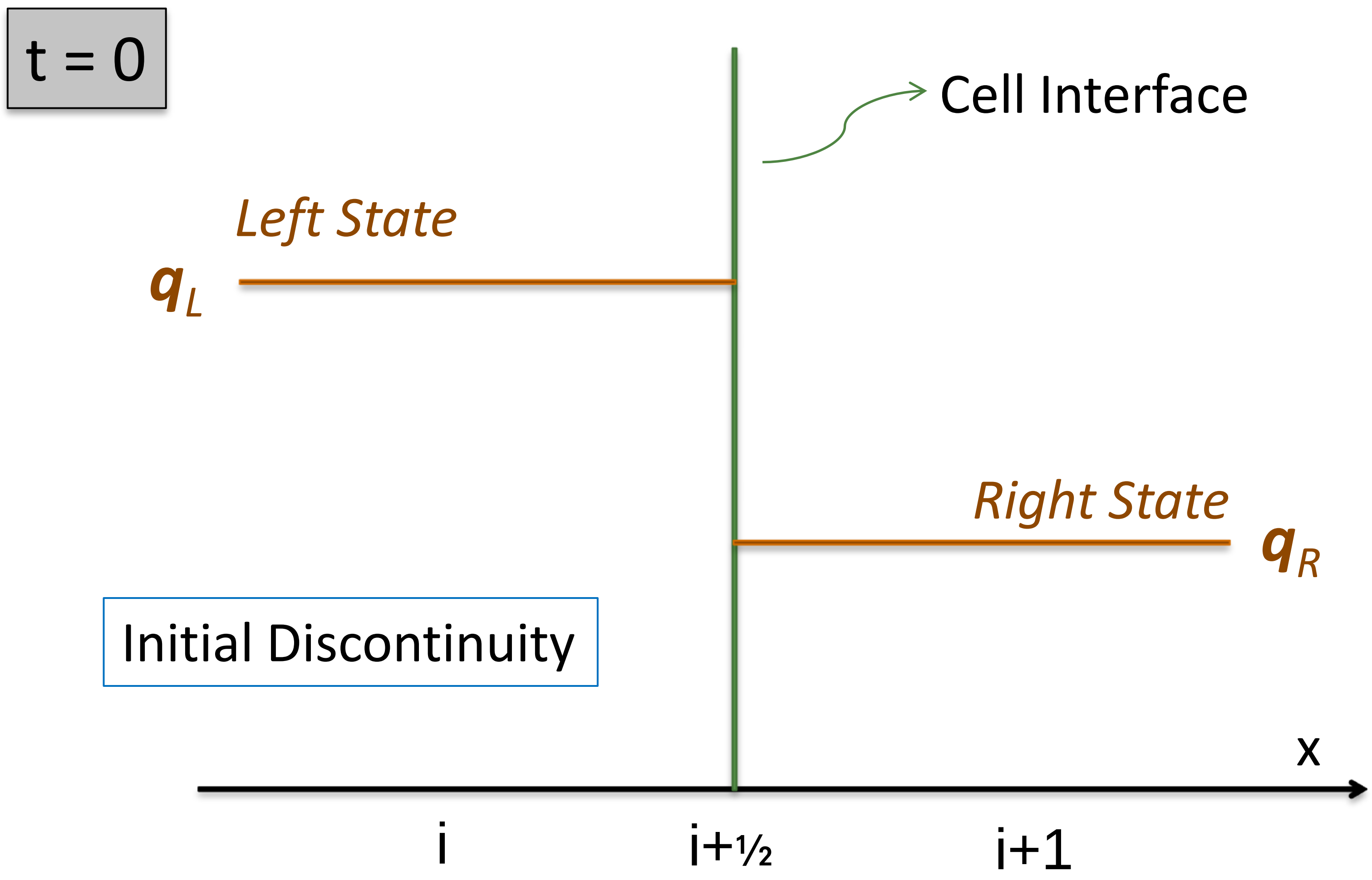
- The previous relations are exact.
- However, since the solution is known only at t^n, some kind of approximation is required in order to evaluate the flux through the boundary:

$$\tilde{\boldsymbol{F}}^n_{i\pm\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \boldsymbol{F}\Big(\boldsymbol{q}(x_{i\pm\frac{1}{2}}, t)\Big)\, dt$$

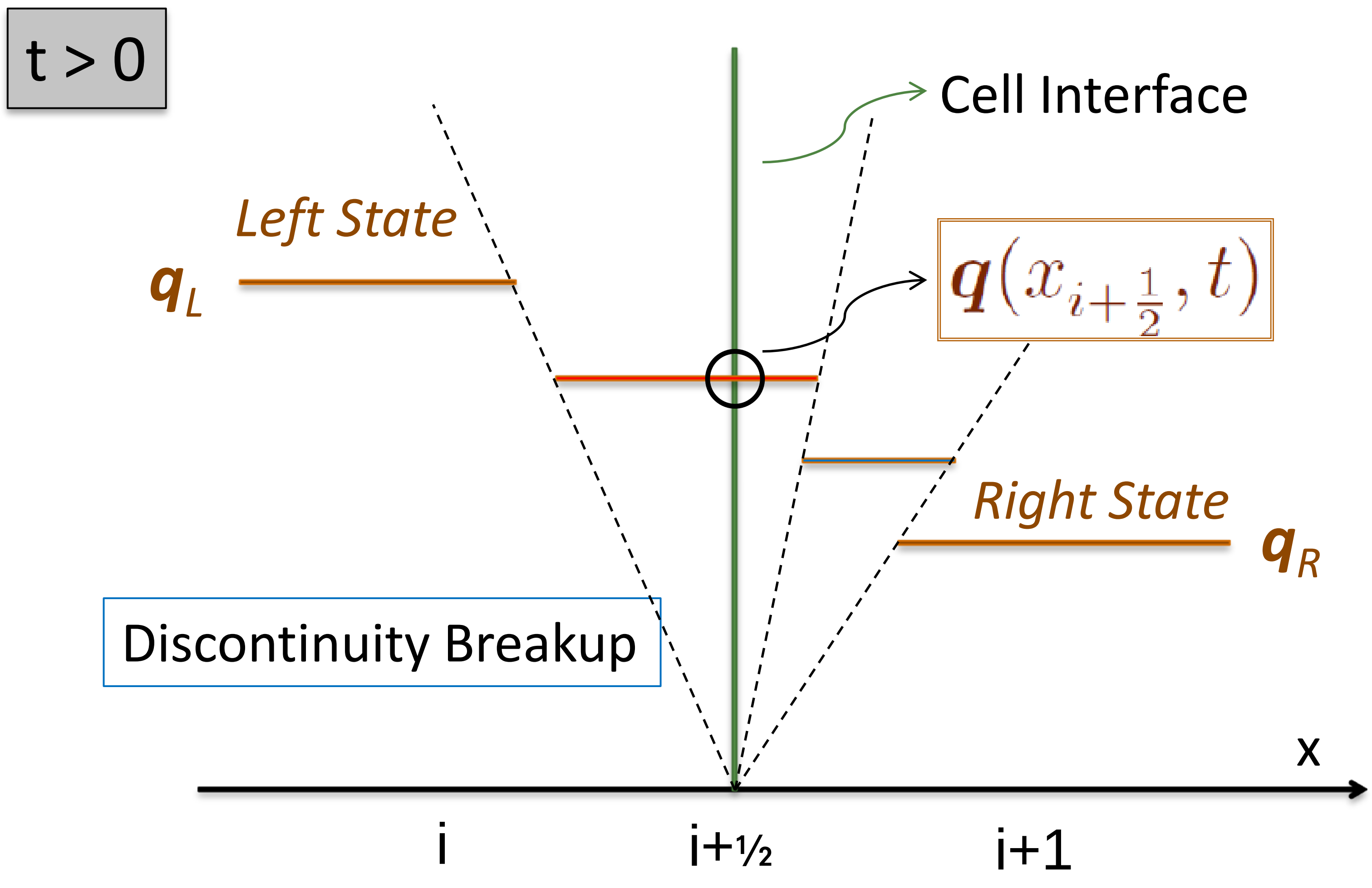
- This achieved by solving the so-called "*Riemann Problem*", i.e., the evolution of an inital discontinuity separating two *constant* states. The Riemann problem is defined by the initial condition:

$$\boldsymbol{q}(x,0) = \begin{cases} \boldsymbol{q}_L & \text{for} \quad x < x_{i+\frac{1}{2}} \\ \boldsymbol{q}_R & \text{for} \quad x > x_{i+\frac{1}{2}} \end{cases} \implies \boldsymbol{q}(x_{i+\frac{1}{2}}, t) = ??$$

The Riemann Problem



The Riemann Problem



2b. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE

THE LINEAR SCALAR ADVECTION EQUATION

The Advection Equation: Theory

- First order partial differential equation (PDE) in (x,t):

$$\frac{\partial q(x,t)}{\partial t} + a\frac{\partial q(x,t)}{\partial x} = 0$$

- Hyperbolic PDE: information propagates across domain at finite speed → method of characteristics
- Characteristic curves are the solutions of the equation

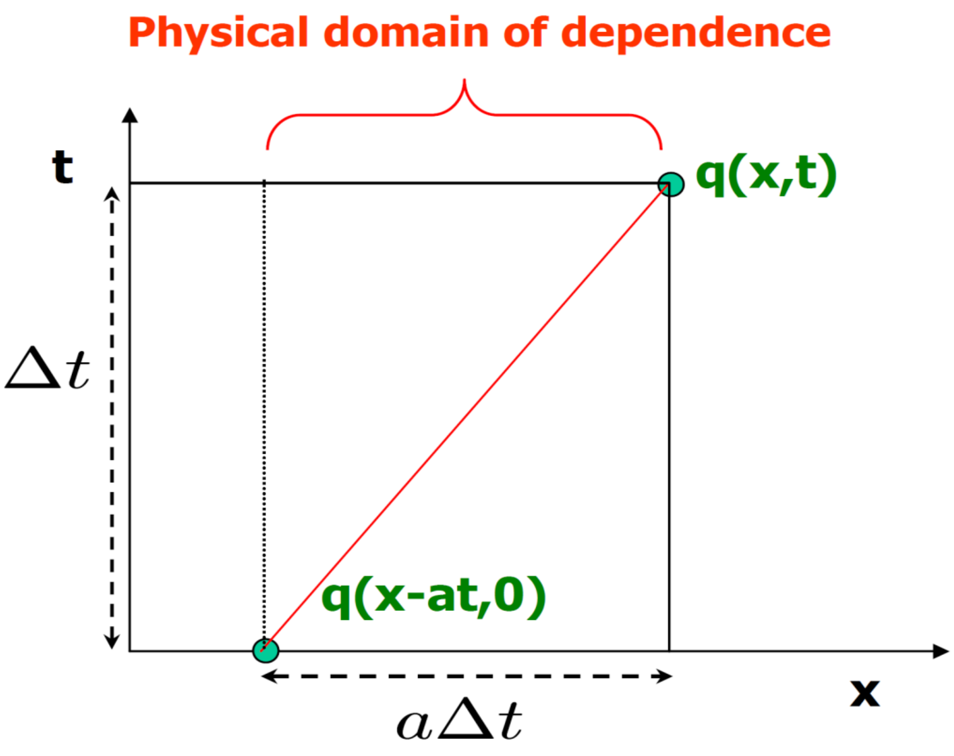
$$\frac{dx}{dt} = a$$

- So that, along each characteristic, the solution satisfies

$$\frac{dq}{dt} = \frac{\partial q}{\partial t} + \frac{dx}{dt}\frac{\partial q}{\partial x} = 0$$

The Advection Equation: Theory

- The solution is constant along the characteristic curves. At any point (x,t) we trace the characteristic back to the initial position.



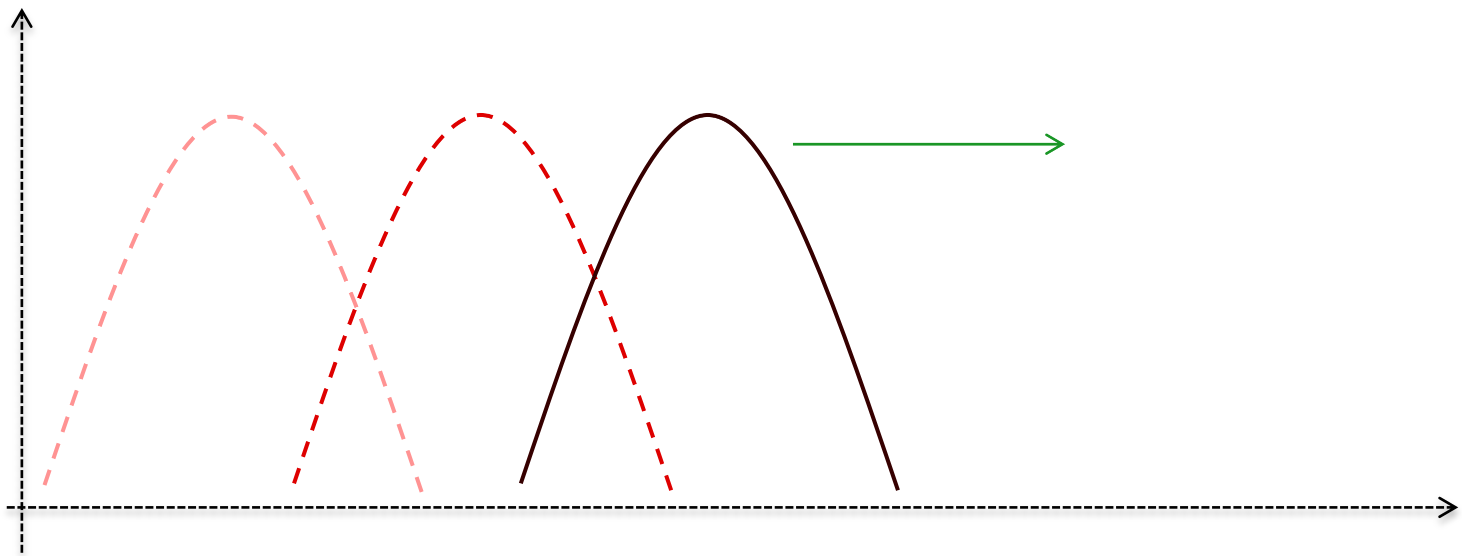
- This defines the physical domain of dependence.

The Advection Equation: Theory

- for constant *a*: the characteristics are straight parallel lines and the solution to the PDE is a uniform shift of the initial profile:

$$q(x,t) = \phi(x - at)$$

- Here $\phi(x) = q(x,0)$ is the initial condition



Discretization: the FTCS Scheme

- We need to approximate the derivatives in our PDE

$$\frac{\partial q(x,t)}{\partial t} + a\frac{\partial q(x,t)}{\partial x} = 0$$

- In time, use forward derivative, since we want to use information from the previous time level:

$$\frac{\partial q(x,t)}{\partial t} \approx \frac{q_i^{n+1} - q_i^n}{\Delta t} + O(\Delta t)$$

- In space, we use centered derivatives, since it is more accurate:

$$\frac{\partial q(x,t)}{\partial x} \approx \frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x} + O(\Delta x^2)$$

The FTCS Scheme

- Putting all together: $$\frac{q_i^{n+1} - q_i^n}{\Delta t} + a\left(\frac{q_{i+1}^n - q_{i-1}^n}{2\Delta x}\right) = 0$$
- and solving with respect to q_i^{n+1} gives

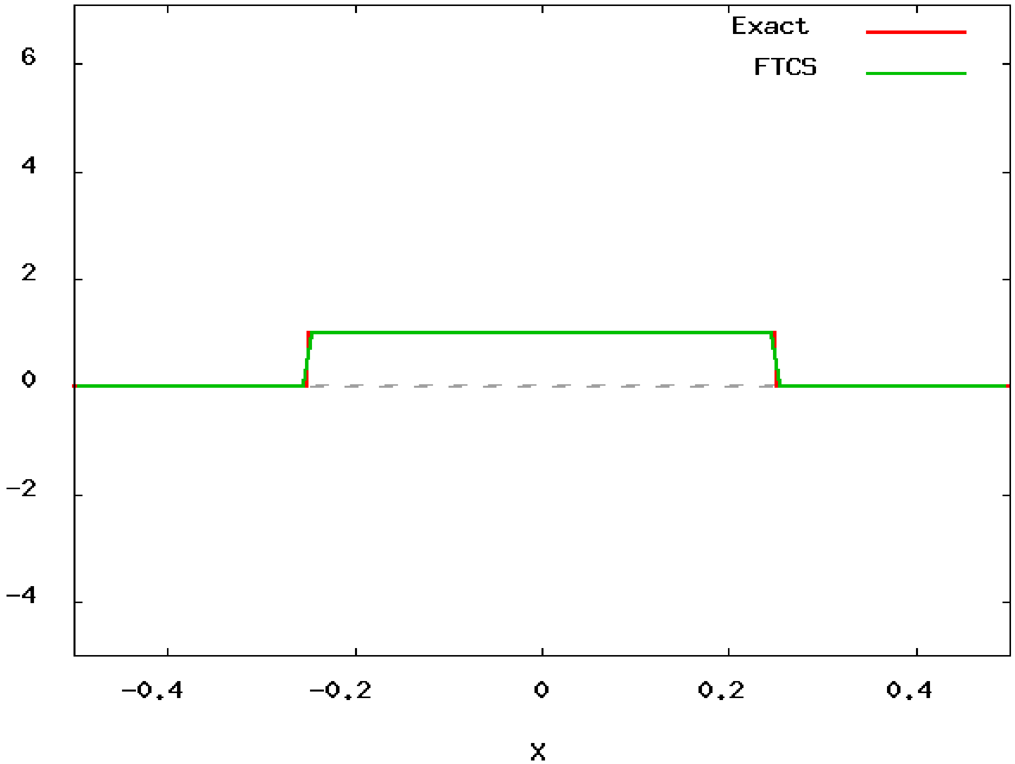
$$q_i^{n+1} = q_i^n - \frac{C}{2}\left(q_{i+1}^n - q_{i-1}^n\right)$$

where $C = a\dfrac{\Delta t}{\Delta x}$ is the Courant-Friedrichs-Lewy (CFL) number.

- We call this method *FTCS* for forward in time, centered in space.
- The value at the new time level depends only on quantities at the previous time steps → *explicit* method.

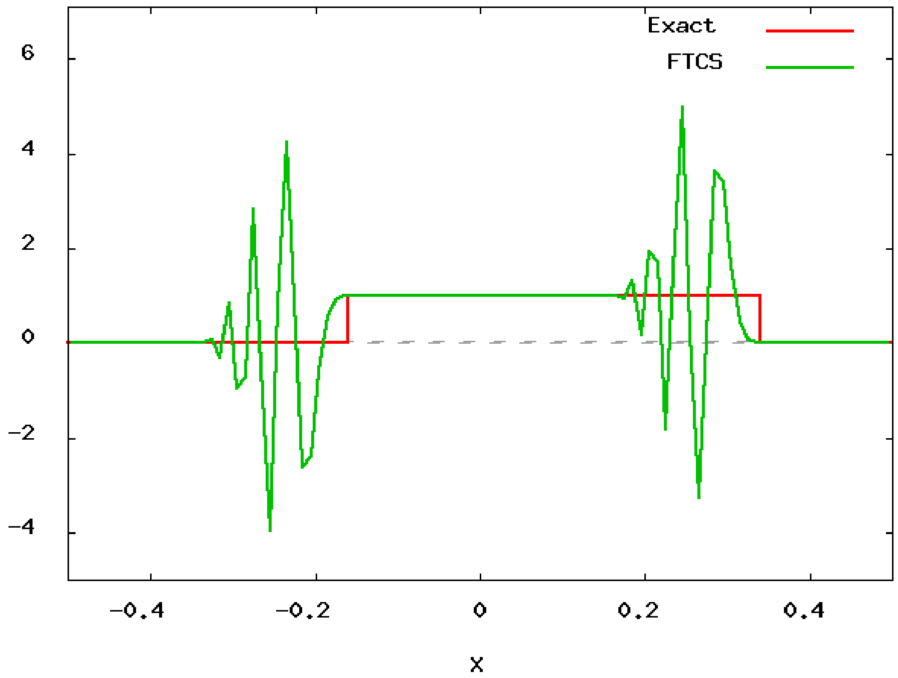
The FTCS Scheme

- At t=0, the *initial condition* is a square pulse with periodic boundary conditions:



The FTCS Scheme

- After some time, the solution looks like this:



- Something isn't right… why ?

von Neumann Stability Analysis

- Let's perform an analysis of *FTCS* by expressing the solution as a Fourier series.

- Since the equation is linear, we only examine the behavior of a single mode. Consider a trial solution of the form

$$q_i^n = A^n e^{Ii\theta}, \quad \theta = k\Delta x$$

- This is a spatial Fourier expansion. Plugging in the difference formula:

$$q_i^{n+1} = q_i^n - \frac{C}{2}\left(q_{i+1}^n - q_{i-1}^n\right) \implies A^{n+1} = A^n - \frac{C}{2}A^n\left(e^{I\theta} - e^{-I\theta}\right)$$

von Neumann Stability Analysis

- Defining the amplification factor $\frac{A^{n+1}}{A^n}$ one obtains

$$\frac{A^{n+1}}{A^n} = 1 - \frac{C}{2}\left(e^{I\theta} - e^{-I\theta}\right) = 1 - IC\sin\theta$$

- a method is well-behaved or stable when $\left|\frac{A^{n+1}}{A^n}\right| \leq 1$
- however, for FTCS, one gets $\left|\frac{A^{n+1}}{A^n}\right|^2 = 1 + C^2\sin^2\theta \geq 1$
- Indipendently of the CFL number, all Fourier modes increase in magnitude as time advances
- This method is unconditionally unstable!

Forward in Time, Backward in Space

- Let's try a difference approach. Consider the backward formula for the spatial derivative:

$$\frac{\partial q(x,t)}{\partial x} \approx \frac{q_i^n - q_{i-1}^n}{\Delta x} + O(\Delta x)$$

- Apply von Neumann stability analysis on the resulting discretized equation:

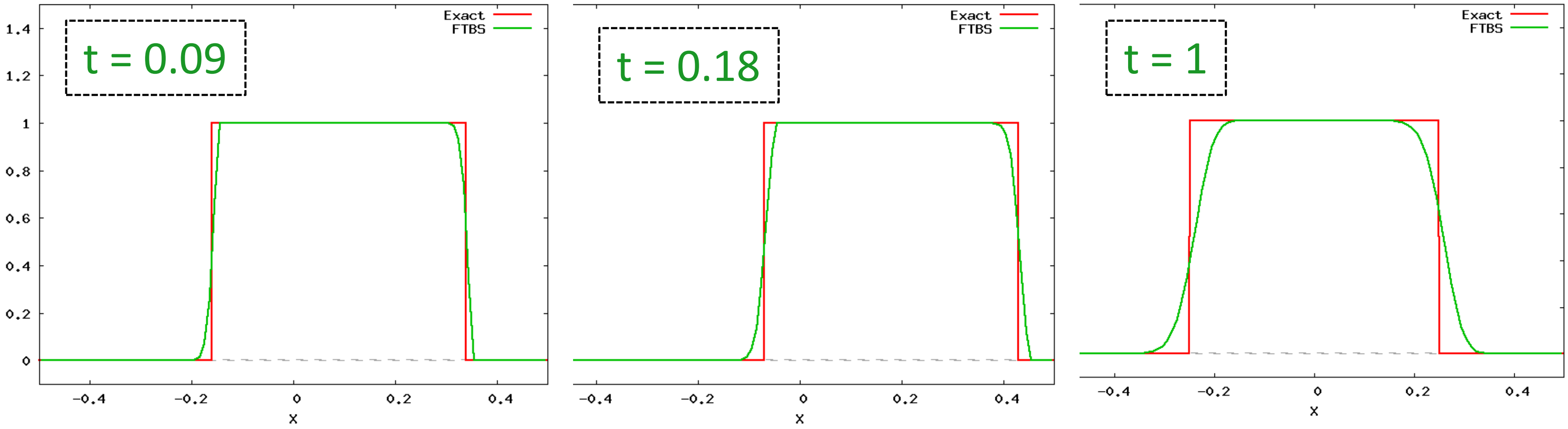
$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + a\left(\frac{q_i^n - q_{i-1}^n}{\Delta x}\right) = 0$$

- Solving for the amplification factor gives

$$\left|\frac{A^{n+1}}{A^n}\right|^2 = 1 - 2C(1-C)(1-\cos\theta)$$

Forward in Time, Backward in Space

- The method is stable when $\left|\frac{A^{n+1}}{A^n}\right| \leq 1 \rightarrow 2C(1-C) \geq 0$
- for a < 0 the method is *unstable*, but
- for a > 0 the method is stable when $0 \leq a\frac{\Delta t}{\Delta x} \leq 1$



Forward in Time, Forward in Space

- Repeating the same argument for the forward derivative

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + a\left(\frac{q_{i+1}^n - q_i^n}{\Delta x}\right) = 0$$

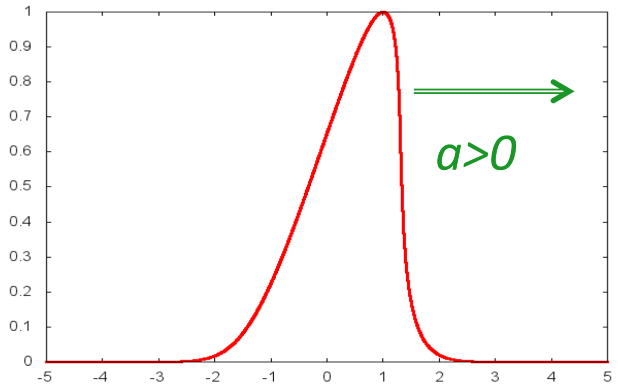
- Gives $$\left|\frac{A^{n+1}}{A^n}\right|^2 = 1 + 2C(1+C)(1-\cos\theta)$$

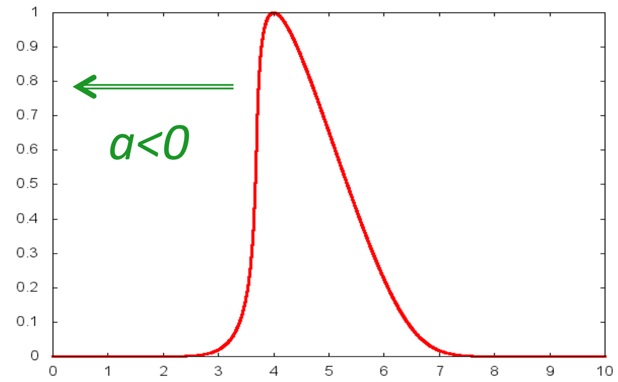
- If *a > 0* the method will always be unstable

- However, if $-1 \leq a\frac{\Delta t}{\Delta x} \leq 0$, then this method is stable;

The 1st Order Godunov Method

- Summarizing: the stable discretization makes use of the grid point where information is coming from:





- This is "upwind":

$$\begin{cases} q_i^{n+1} = q_i^n - \dfrac{a\Delta t}{\Delta x}\left(q_i^n - q_{i-1}^n\right) & \text{for} \quad a > 0 \\ q_i^{n+1} = q_i^n - \dfrac{a\Delta t}{\Delta x}\left(q_{i+1}^n - q_i^n\right) & \text{for} \quad a < 0 \end{cases}$$

- This is also called the first-order Godunov method;

Conservative Form

- We define the "flux" function $F^n_{i+\frac{1}{2}} = \frac{a}{2}\left(q^n_{i+1} + q^n_i\right) - \frac{|a|}{2}\left(q^n_{i+1} - q^n_i\right)$ so that Godunov method can be cast in *conservative* form

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\Delta x}\left(F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}\right)$$

a > 0

$$q_i^{n+1} = q_i^n - \frac{a\Delta t}{\Delta x}\left(q_i^n - q_{i-1}^n\right)$$

a < 0

$$q_i^{n+1} = q_i^n - \frac{a\Delta t}{\Delta x}\left(q_{i+1}^n - q_i^n\right)$$

- The conservative form ensures a correct description of *discontinuities* in nonlinear systems, ensures global conservation properties and is the main building block in the development of high-order finite volume schemes.

The CFL Condition

- Since the advection speed a is a parameter of the equation, Δx is fixed from the grid, the previous inequality is a stability constraint on the time step

$$\Delta t \leq \frac{\Delta x}{|a|}$$

- Δt cannot be arbitrarily large but, rather, less than the time taken to travel one grid cell (CFL) condition.

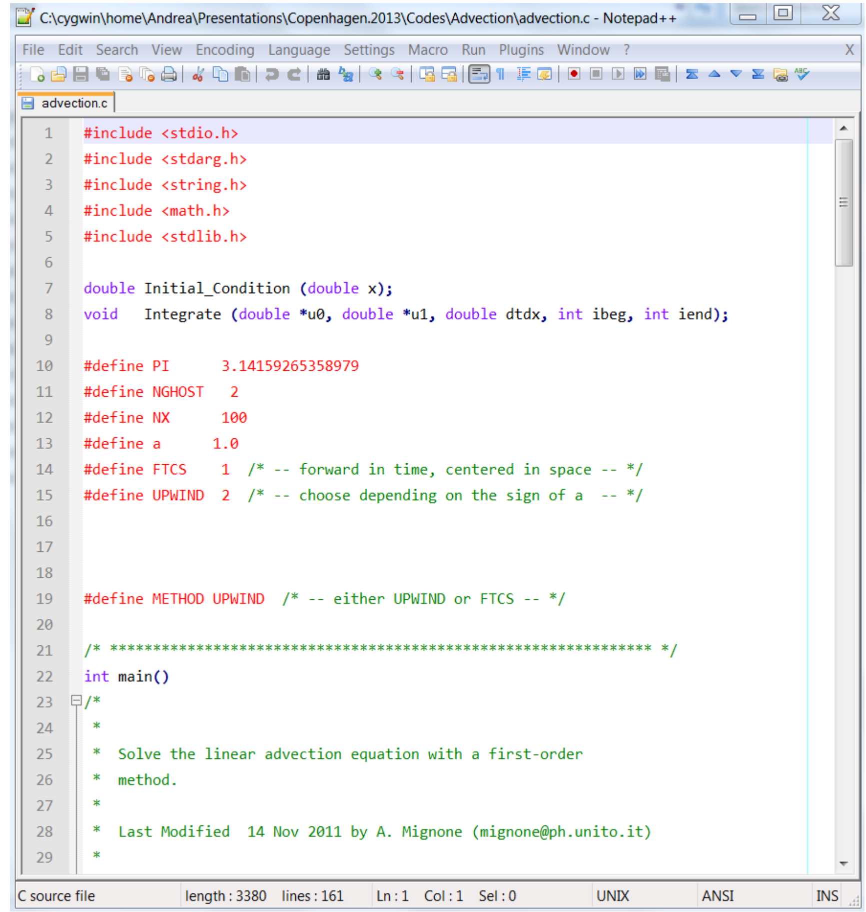
- In the case of nonlinear equations, the speed can vary in the domain and the maximum of a should be considered instead.

Code Example

- *File name*: `advection.c`
- *Purpose*: solve the linear advection equation using the 1st-order Godunov method.
- *Usage*:

```
> gcc –O advection.c –o advection
> ./advection
```

- *Output*: two-column ascii data file.




```c
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

double Initial_Condition (double x);
void   Integrate (double *u0, double *u1, double dtdx, int ibeg, int iend);

#define PI      3.14159265358979
#define NGHOST   2
#define NX      100
#define a      1.0
#define FTCS    1  /* -- forward in time, centered in space -- */
#define UPWIND  2  /* -- choose depending on the sign of a  -- */



#define METHOD UPWIND  /* -- either UPWIND or FTCS -- */

/* *************************************************************** */
int main()
/*
 *
 *  Solve the linear advection equation with a first-order
 *  method.
 *
 *  Last Modified  14 Nov 2011 by A. Mignone (mignone@ph.unito.it)
 *
```

2c. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE:

SYSTEM OF LINEAR EQUATIONS

System of Equations: Theory

- We turn our attention to the system of equations (PDE)

$$\frac{\partial \mathbf{q}}{\partial t} + A \cdot \frac{\partial \mathbf{q}}{\partial x} = 0$$

where $\mathbf{q} = \{q_1, q_2, \ldots q_m\}$ is the vector of unknowns. *A* is a $m \times m$ constant matrix.

- For example, for m=3, one has

$$\frac{\partial q_1}{\partial t} + A_{11}\frac{\partial q_1}{\partial x} + A_{12}\frac{\partial q_2}{\partial x} + A_{13}\frac{\partial q_3}{\partial x} = 0$$

$$\frac{\partial q_2}{\partial t} + A_{21}\frac{\partial q_1}{\partial x} + A_{22}\frac{\partial q_2}{\partial x} + A_{23}\frac{\partial q_3}{\partial x} = 0$$

$$\frac{\partial q_3}{\partial t} + A_{31}\frac{\partial q_1}{\partial x} + A_{32}\frac{\partial q_2}{\partial x} + A_{33}\frac{\partial q_3}{\partial x} = 0$$

System of Equations: Theory

- The system is hyperbolic if A has real eigenvalues, $\lambda^1 \leq \ldots \leq \lambda^m$ and a complete set of linearly independent right and left eigenvectors r^k and l^k ($r^j \cdot l^k = \delta_{jk}$) such that

$$\begin{cases} A \cdot \boldsymbol{r}^k = \lambda^k \boldsymbol{r}^k \\ \boldsymbol{l}^k \cdot A = \boldsymbol{l}^k \lambda^k \end{cases} \quad \text{for} \quad k = 1, \ldots, m$$

- For convenience we define the matrices $\Lambda = diag(\lambda^k)$, and

$$R = \left(\mathbf{r^1}|\mathbf{r^2}|\ldots|\mathbf{r^m}\right), \quad L = R^{-1} = \begin{pmatrix} \mathbf{l}^1 \\ \hline \mathbf{l}^2 \\ \hline \vdots \\ \hline \mathbf{l}^m \end{pmatrix}$$

- So that $A \cdot R = R \cdot \Lambda$, $L \cdot A = \Lambda \cdot L$, $L \cdot R = R \cdot L = I$, $L \cdot A \cdot R = \Lambda$.

System of Equations: Theory

- The linear system can be reduced to a set of decoupled linear advection equations.
- Multiply the original system of PDE's by *L* on the left:

$$L \cdot \left(\frac{\partial \boldsymbol{q}}{\partial t} + A \cdot \frac{\partial \boldsymbol{q}}{\partial x} \right) = L \cdot \frac{\partial \boldsymbol{q}}{\partial t} + L \cdot A \cdot R \cdot L \cdot \frac{\partial \boldsymbol{q}}{\partial x} = 0$$

- Define the *characteristic variables* $w = L \cdot q$ so that

$$\frac{\partial \boldsymbol{w}}{\partial t} + \Lambda \cdot \frac{\partial \boldsymbol{w}}{\partial x} = 0$$

- Since Λ is diagonal, these equations are not coupled anymore.

System of Equations: Theory

- In this form, the system decouples into *m* independent advection equations for the characteristic variables:

$$\frac{\partial \boldsymbol{w}}{\partial t} + \Lambda \cdot \frac{\partial \boldsymbol{w}}{\partial x} = 0 \quad \Longrightarrow \quad \frac{\partial w^k}{\partial t} + \lambda^k \cdot \frac{\partial w^k}{\partial x} = 0$$

where $w^k = \mathbf{l}^k \cdot \mathbf{q}$ (k=1,2,…,m) is a characteristic variable.

- When m=3 one has, for instance:

$$\frac{\partial w^1}{\partial t} + \lambda^1 \frac{\partial w^1}{\partial x} = 0$$

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

$$\frac{\partial w^3}{\partial t} + \lambda^3 \frac{\partial w^3}{\partial x} = 0$$

System of Equations: Theory

- The m advection equations can be solved independently by applying the standard solution techniques developed for the scalar equation.

- In particular, one can write the *exact analytical solution* for the k-th characteristic field as

$$w^k(x,t) = w^k(x - \lambda^k t, 0)$$

i.e., the initial profile of w^k shifts with uniform velocity λ^k, and

$$w^k(x - \lambda^k t, 0) = \mathbf{l}^k \cdot \mathbf{q}(x - \lambda^k t, 0)$$

is the initial profile.

- The characteristics are thus constant along the curves $dx/dt = \lambda^k$

System of Equations: Exact Solution

- Once the solution in characteristic space is known, we can solve the original system via the inverse transformation

$$\mathbf{q}(x,t) = R \cdot \mathbf{w}(x,t) = \sum_{k=1}^{k=m} w^k(x,t)\mathbf{r}^k = \sum_{k=1}^{k=m} w^k(x-\lambda^k t, 0)\mathbf{r}^k$$

- The characteristic variables are thus the coefficients of the right eigenvector expansion of q.
- The solution to the linear system reduces to a linear combination of m linear waves traveling with velocities λ^k .
- Expressing everything in terms of the original variables q,

$$\mathbf{q}(x,t) = \sum_{k=1}^{k=m} \mathbf{l}^k \cdot \mathbf{q}(x-\lambda^k t, 0)\mathbf{r}^k$$

Riemann Problem for Discontinuous Data

- If $\boldsymbol{q}$ is initially discontinuous, one or more characteristic variables will also have a discontinuity. Indeed, at $t = 0$,

$$w^k(x,0) = \boldsymbol{l}^k \cdot \boldsymbol{q}(x,0) = \begin{cases} w_L^k = \boldsymbol{l}^k \cdot \boldsymbol{q}_L & \text{if} \quad x < x_{i+\frac{1}{2}} \\ w_R^k = \boldsymbol{l}^k \cdot \boldsymbol{q}_R & \text{if} \quad x > x_{i+\frac{1}{2}} \end{cases}$$

- In other words, the initial jump $\boldsymbol{q}_R$ - $\boldsymbol{q}_L$ is decomposed in several waves each propagating at the constant speed λ^k and corresponding to the eigenvectors of the Jacobian A:

$$\boldsymbol{q}_R - \boldsymbol{q}_L = \alpha^1 \boldsymbol{r}^1 + \alpha^2 \boldsymbol{r}^2 + \cdots + \alpha^m \boldsymbol{r}^m$$

where $\alpha^k = \boldsymbol{l}^k \cdot (\boldsymbol{q}_R - \boldsymbol{q}_L)$ are the *wave strengths*

Riemann Problem for Discontinuous Data

- For the linear case, the *exact* solution for each wave at the cell interface is:

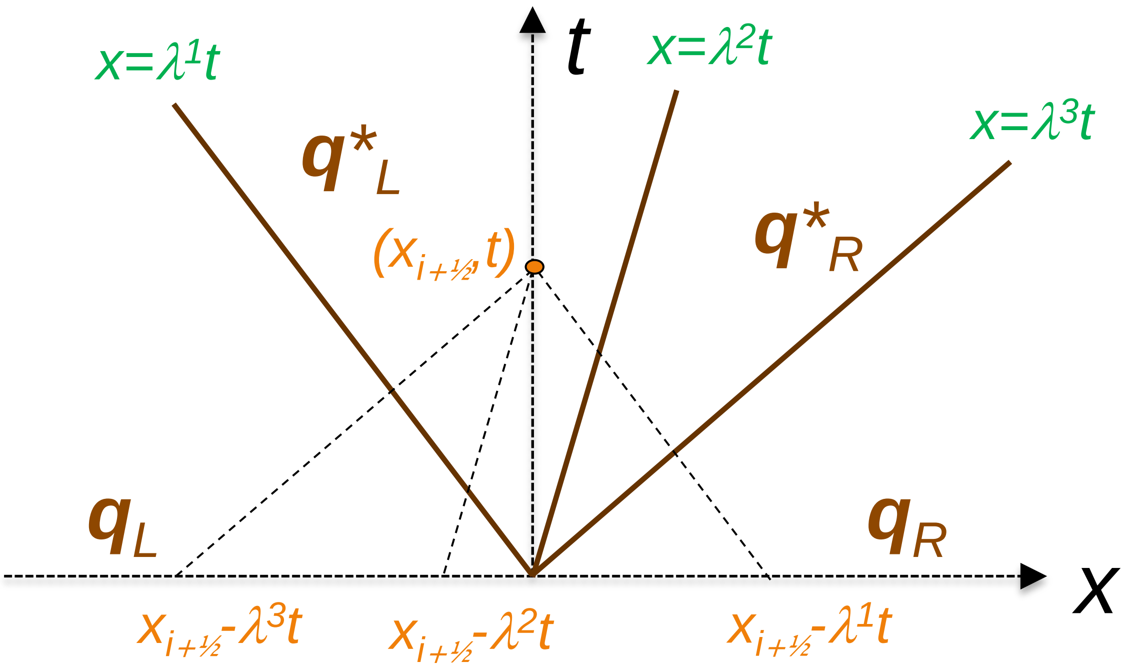
$$w^k\left(x_{i+\frac{1}{2}}, t\right) = w^k\left(x_{i+\frac{1}{2}} - \lambda^k t, 0\right) = \begin{cases} w_L^k & \text{if} & \lambda^k > 0 \\ w_R^k & \text{if} & \lambda^k < 0 \end{cases}$$

- The complete solution is found by adding all wave contributions:

$$\boldsymbol{q}\left(x_{i+\frac{1}{2}}, t\right) = \sum_{k:\lambda_k>0} w_L^k \boldsymbol{r}^k + \sum_{k:\lambda_k<0} w_R^k \boldsymbol{r}^k$$

- and the flux is finally computed as $\tilde{\boldsymbol{F}}_{i+\frac{1}{2}} = A \cdot \boldsymbol{q}\left(x_{i+\frac{1}{2}}, t\right)$

The Riemann Problem



Point (X_0,T) falls to the right of the λ^1 characteristic emanating from the initial jump, but to the left of the other 2, so the solution is:

$$\boldsymbol{q}\left(x_{i+\frac{1}{2}}, t\right) = w_R^1 \boldsymbol{r}^1 + w_L^2 \boldsymbol{r}^2 + w_L^3 \boldsymbol{r}^3$$

System of Equations: Numerics

- We suppose the solution at time level n is known as q^n and we wish to compute the solution q^{n+1} at the next time level $n+1$.

- Our numerical scheme can be derived by working in the characteristic space and then transforming back:

$$\boldsymbol{q}_i^{n+1} = \sum_k w_i^{k,n+1} \boldsymbol{r}^k = \boldsymbol{q}_i^n - \frac{\Delta t}{\Delta x} \left(\boldsymbol{F}_{i+\frac{1}{2}}^n - \boldsymbol{F}_{i-\frac{1}{2}}^n \right)$$

where

$$\boxed{\boldsymbol{F}_{i+\frac{1}{2}}^n = A \cdot \frac{\boldsymbol{q}_{i+1}^n + \boldsymbol{q}_i^n}{2} - \frac{1}{2} \sum_k |\lambda^k| \boldsymbol{l}^k \cdot (\boldsymbol{q}_{i+1}^n - \boldsymbol{q}_i^n) \boldsymbol{r}^k}$$

is the *Godunov flux* for a linear system of advection equations.

2d. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE:

NONLINEAR SCALAR EQUATION

Nonlinear Advection Equation

- We turn our attention to the scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0$$

- Where *f(u)* is, in general, a nonlinear function of *u*.

- To gain some insights on the role played by nonlinear effects, we start by considering the inviscid Burger's equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0$$

Nonlinear Advection Equation

- We can write Burger's equation also as $$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$
- In this form, Burger's equation resembles the linear advection equation, except that the velocity is no longer constant but it is equal to the solution itself.
- The characteristic curve for this equation is

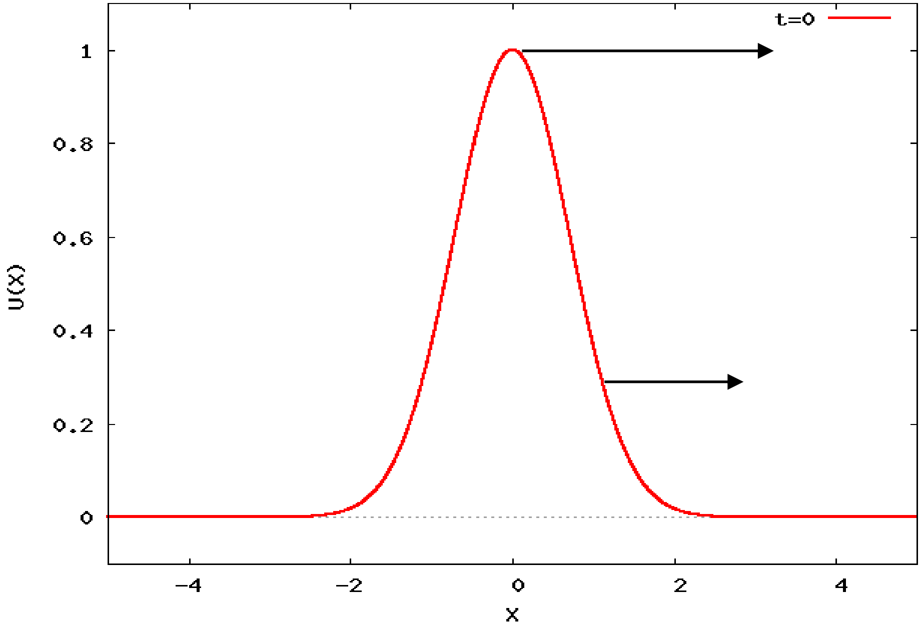
$$\frac{dx}{dt} = u(x,t) \quad \Longrightarrow \quad \frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{dx}{dt} = 0$$

- → *u* is constant along the curve *dx/dt=u(x,t)* → characteristics are again straight lines: values of *u* associated with some fluid element do not change as that element moves.

Nonlinear Advection Equation

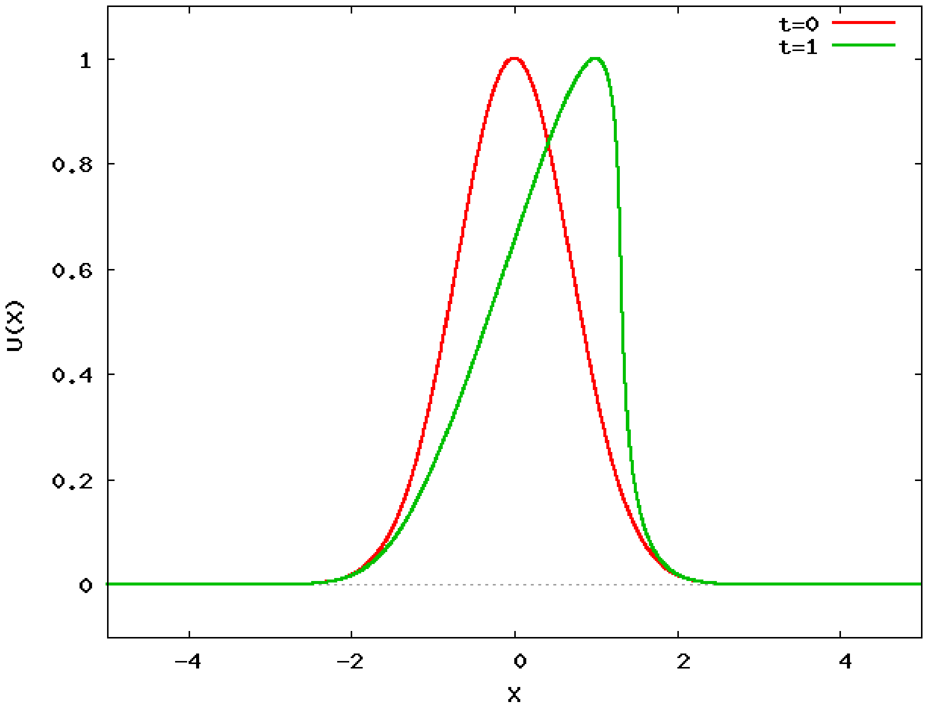
- From $$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

one can predict that, higher values of *u* will propagate faster than lower values: this leads to a wave steepening, since upstream values will advances faster than downstream values.



Nonlinear Advection Equation

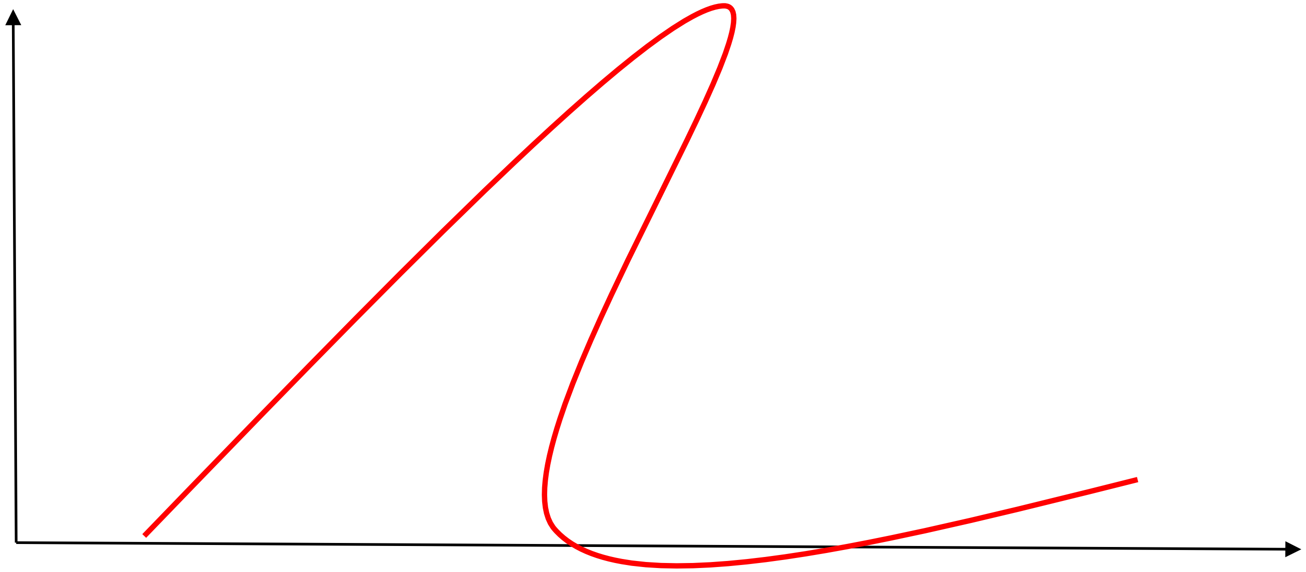
- Indeed, at t=1 the wave profile will look like:



- the wave steepens…

Nonlinear Advection Equation

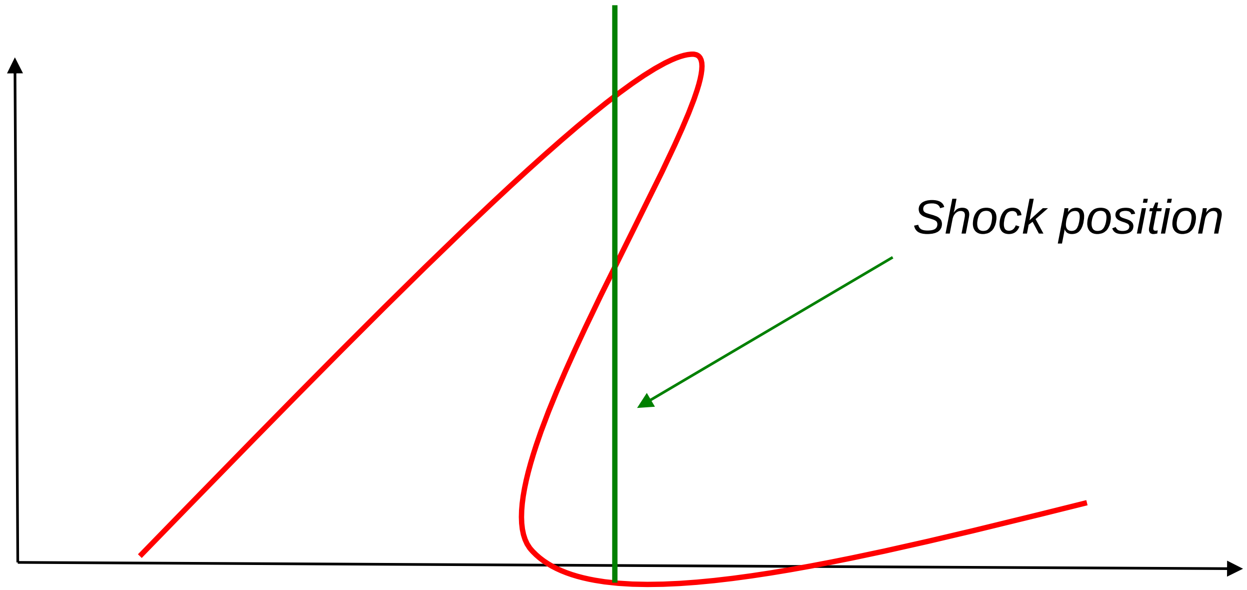
- If we wait more, we should get something like this:



- A multi-value functions ?! → Clearly ***NOT*** physical !

Nonlinear Advection Equation

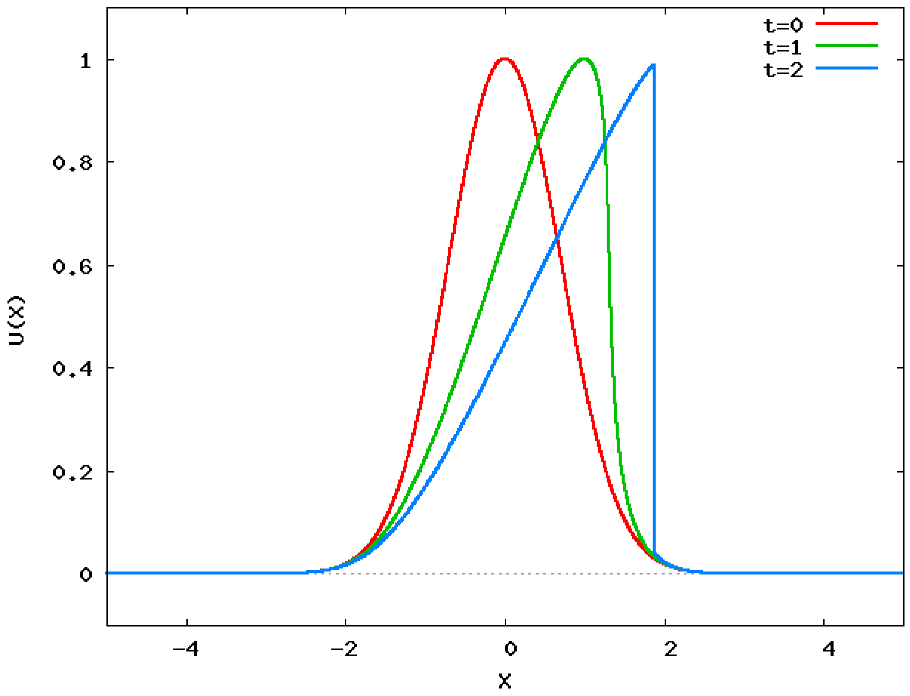
- The correct physical solution is to place a discontinuity there:
  a *shock wave*.



- Since the solution is no longer smooth, the differential form is not valid anymore and we need to consider the *integral form*.

Nonlinear Advection Equation

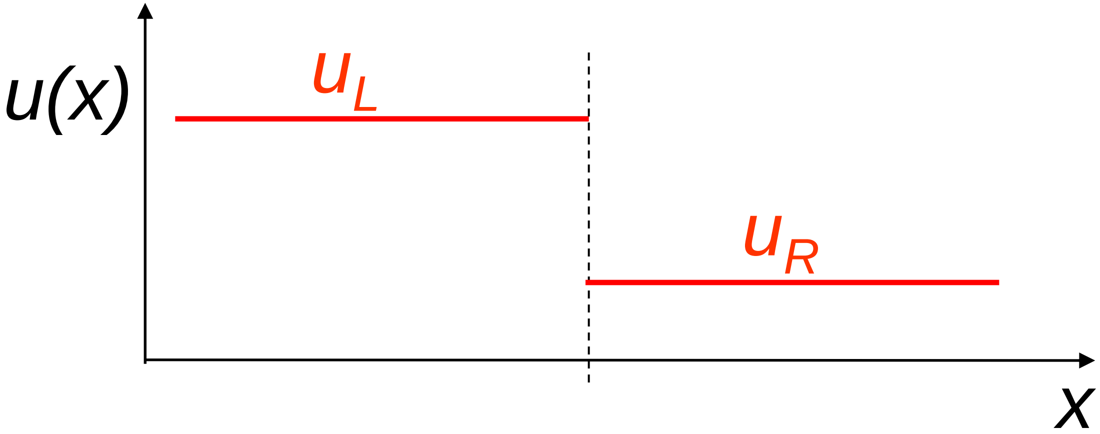
- This is how the solution should look like:



- Such solutions to the PDE are called *weak solutions*.

Nonlinear Advection Equation

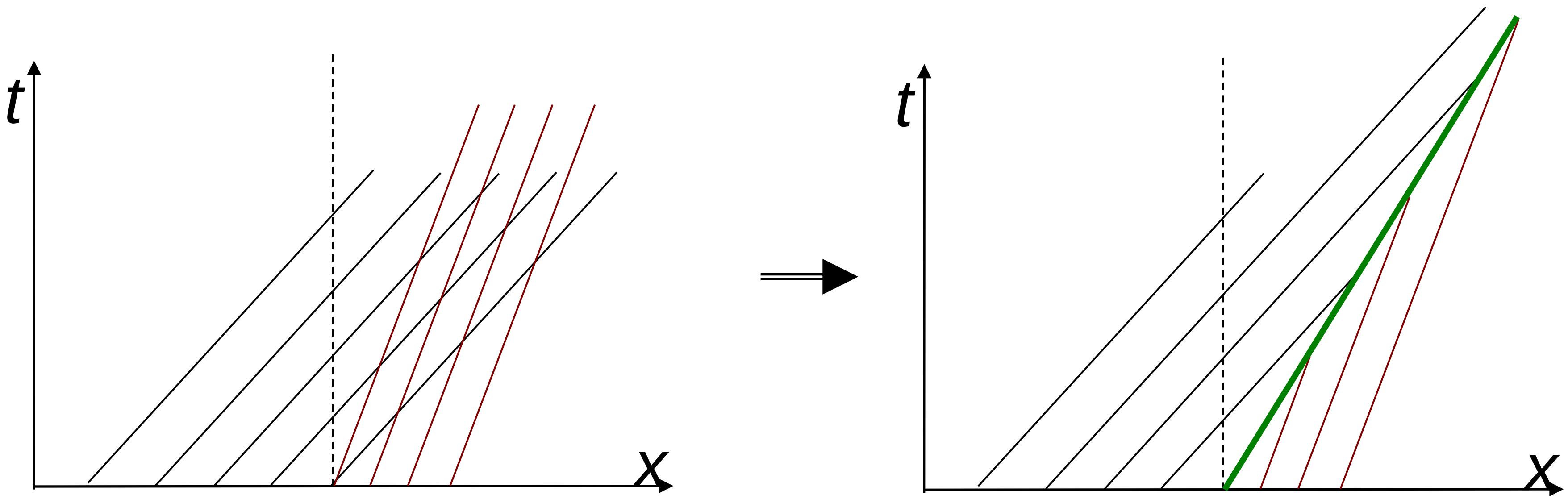
- Let's try to understand what happens by looking at the characteristics.
- Consider two states initially separated by a jump at an interface:



- Here, the characteristic velocities on the left are greater than those on the right.

Nonlinear Advection Equation

- The characteristic will intersect, creating a *shock wave*:



- The shock speed is such that $\lambda(u_L) > S > \lambda(u_R)$. This is called the *entropy condition*.

Nonlinear Advection Equation

- The shock speed *S* can be found using the Rankine-Hugoniot jump conditions, obtained from the integral form of the equation:

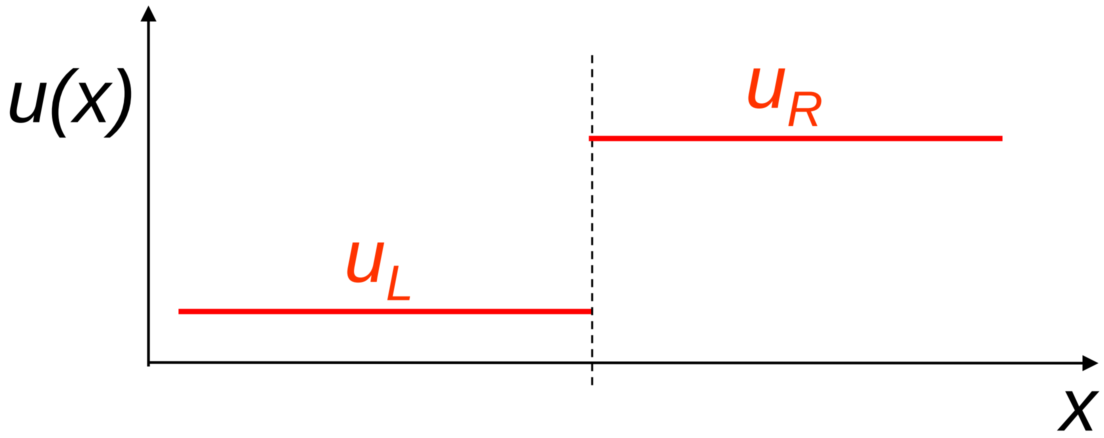
$$f(u_R) - f(u_L) = S(u_R - u_L)$$

- For Burger's equation $f(u) = u^2/2$, one finds the shock speed as

$$S = \frac{u_L + u_R}{2}$$

Nonlinear Advection Equation

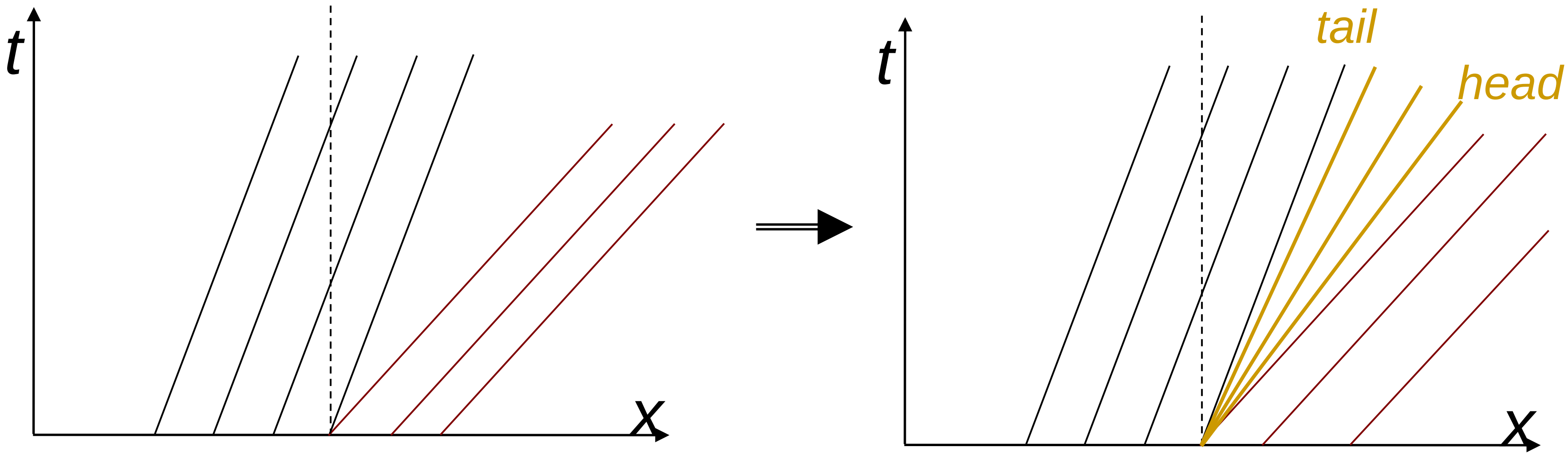
- Let's consider the opposite situation:



- Here, the characteristic velocities on the left are smaller than those on the right.

Nonlinear Advection Equation

- Now the characteristics will diverge:



- Putting a shock wave between the two states would be incorrect, since it would violate the entropy condition. Instead, the proper solution is a *rarefaction wave*.

Nonlinear Advection Equation

- A rarefaction wave is a nonlinear wave that smoothly connects the left and the right state. It is an expansion wave.
- The solution between the states can only be self-similar and takes on the range of values between u_L and u_R
- The head of the rarefaction moves at the speed $\lambda(u_R)$, whereas the tail moves at the speed $\lambda(u_L)$.
- The general condition for a rarefaction wave is $\lambda(u_L)<\lambda(u_R)$
- Both rarefactions and shocks are present in the solutions to the Euler equation. Both waves are nonlinear.

Nonlinear Advection Equation

- These results can be used to write the general solution to the Riemann problem for Burger's equation:
  - If $u_L > u_R$ the solution is a discontinuity (*shock wave)*. In this case

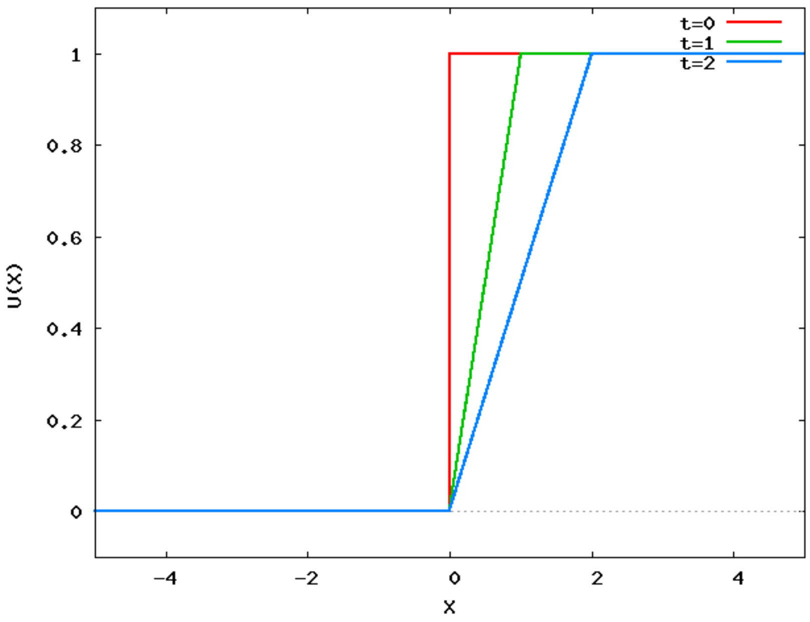
$$u(x,t) = \begin{cases} u_L & \text{if} \quad x - St < 0 \\ u_R & \text{if} \quad x - St > 0 \end{cases} , \qquad S = \frac{u_L + u_R}{2}$$

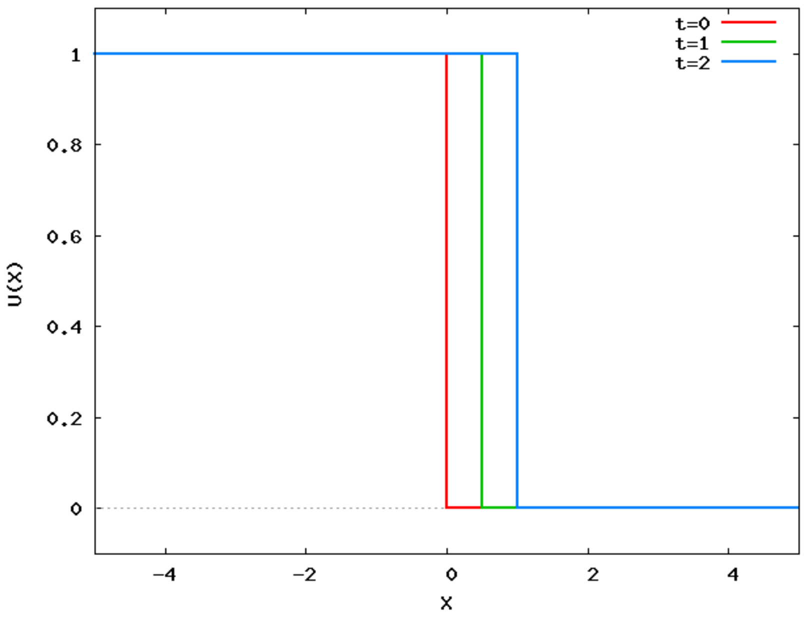
  - If $u_L < u_R$ the solution is a *rarefaction wave*. In this case

$$u(x,t) = \begin{cases} u_L & \text{if} \quad x/t \le u_L \\ x/t & \text{if} \quad u_L < x/t < u_R \\ u_R & \text{if} \quad x/t > u_R \end{cases}$$

Nonlinear Advection Equation

- Solutions look like





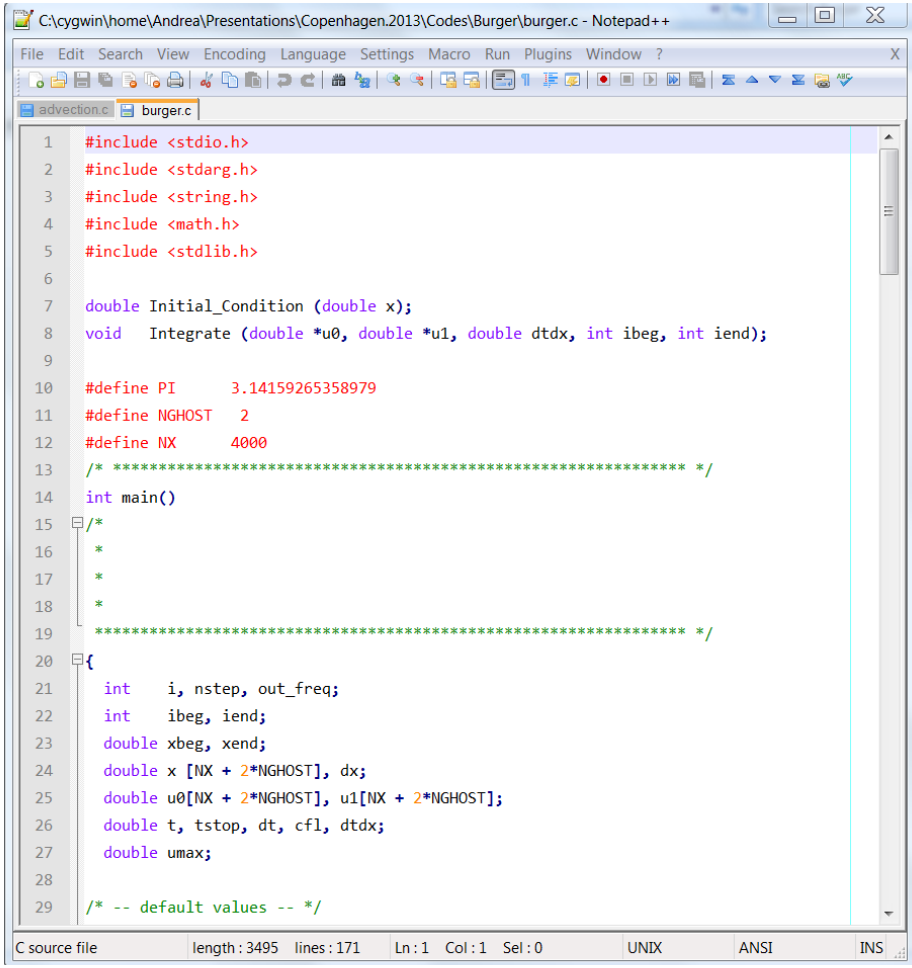
- for a rarefaction and a shock, respectively.

Code Example

- *File name*: `burger.c`
- *Purpose*: solve Burger's equation using 1st-order Godunov method.
- *Usage*:

```
> gcc –O burger.c –o burger
> ./burger
```

- *Output*: two-column ascii data files "data.nnnn.out"




```c
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

double Initial_Condition (double x);
void   Integrate (double *u0, double *u1, double dtdx, int ibeg, int iend);

#define PI      3.14159265358979
#define NGHOST   2
#define NX      4000
/* ************************************************************** */
int main()
/*
 *
 *
 *
 ************************************************************** */
{
  int    i, nstep, out_freq;
  int    ibeg, iend;
  double xbeg, xend;
  double x [NX + 2*NGHOST], dx;
  double u0[NX + 2*NGHOST], u1[NX + 2*NGHOST];
  double t, tstop, dt, cfl, dtdx;
  double umax;

/* -- default values -- */
```

2e. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE:

NONLINEAR SYSTEMS

Nonlinear Systems

- Much of what is known about the numerical solution of hyperbolic systems of nonlinear equations comes from the results obtained in the linear case or simple nonlinear scalar equations.

- The key idea is to exploit the conservative form and assume the system can be locally "frozen" at each grid interface.

- However, this still requires the solution of the Riemann problem, which becomes increasingly difficult for complicated set of hyperbolic P.D.E.

Euler Equations

- System of conservation laws describing conservation of mass, momentum and energy:

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0 && \text{(mass)}\\
&\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot[\rho\mathbf{v}\mathbf{v} + \mathbf{I}p] = 0 && \text{(momentum)}\\
&\frac{\partial E}{\partial t} + \nabla\cdot[(E+p)\,\mathbf{v}] = 0 && \text{(energy)}
\end{aligned}
$$

- Total energy density E is the sum of thermal + Kinetic terms:

$$E = \rho\epsilon + \rho\frac{\mathbf{v}^2}{2}$$

- Closure requires an Equation of State (EoS). For an ideal gas one has

$$\rho\epsilon = \frac{p}{\Gamma - 1}$$

Euler Equations: Characteristic Structure

- The equations of gasdynamics can also be written in "quasi-linear" or primitive form. In 1D:

$$\frac{\partial \mathbf{V}}{\partial t} + A \cdot \frac{\partial \mathbf{V}}{\partial x} = 0\,, \qquad A = \left(\begin{array}{ccc} v_x & \rho & 0 \\ 0 & v_x & 1/\rho \\ 0 & \rho c_s^2 & v_x \end{array}\right)$$

where $V = [\rho, v_x, p]$ is a vector of primitive variable, $c_s = (\gamma p/\rho)^{1/2}$ is the adiabatic speed of sound.

- It is called "quasi-linear" since, differently from the linear case where we had $A=const$, here $A = A(V)$.

Euler Equations: Characteristic Structure

- The quasi-linear form can be used to find the eigenvector decomposition of the matrix *A*:

$$\mathbf{r}^1 = \begin{pmatrix} 1 \\ -c_s/\rho \\ c_s^2 \end{pmatrix}, \quad \mathbf{r}^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{r}^3 = \begin{pmatrix} 1 \\ c_s/\rho \\ c_s^2 \end{pmatrix}$$

- Associated to the eigenvalues:

$$\lambda^1 = v_x - c_s\,, \quad \lambda^2 = v_x\,, \quad \lambda^3 = v_x + c_s$$

- These are the characteristic speeds of the system, i.e., the speeds at which information propagates. They tell us a lot about the structure of the solution.

Euler Equations: Riemann Problem

- By looking at the expressions for the right eigenvectors,

$$\mathbf{r}^1 = \begin{pmatrix} 1 \\ -c_s/\rho \\ c_s^2 \end{pmatrix}, \quad \mathbf{r}^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{r}^3 = \begin{pmatrix} 1 \\ c_s/\rho \\ c_s^2 \end{pmatrix}$$

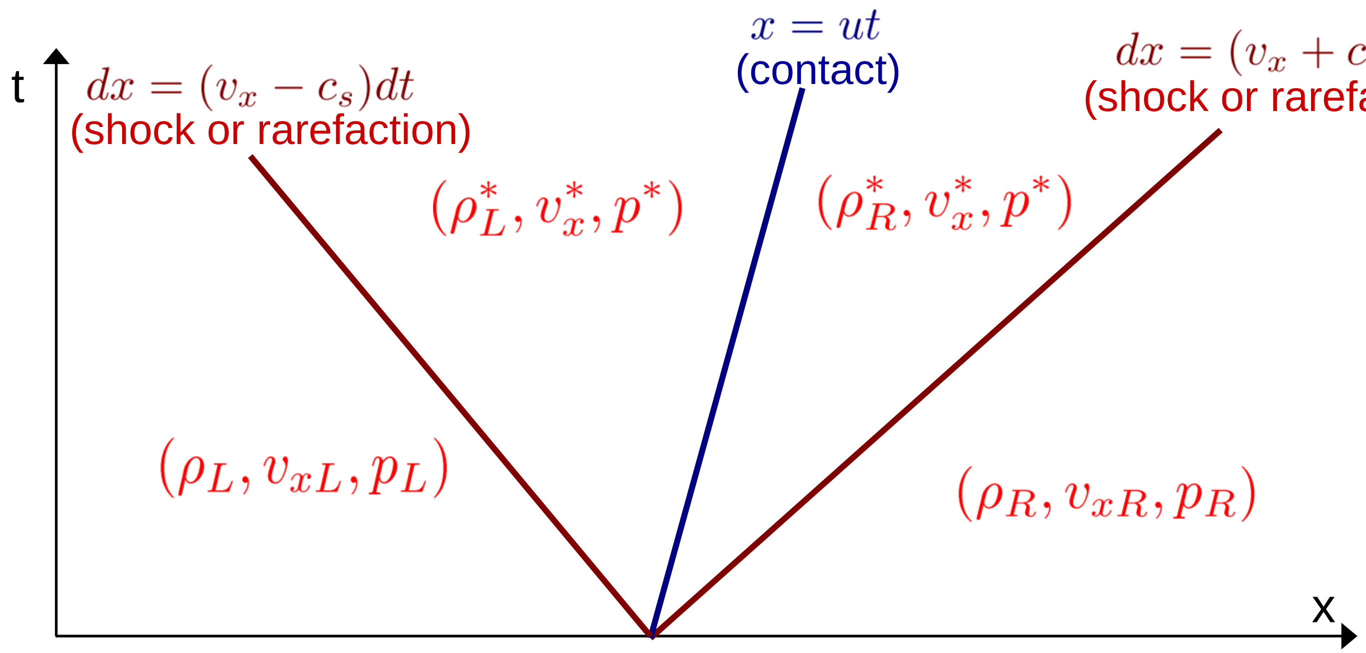
- we see that across waves 1 and 3, all variables jump. These are nonlinear waves, either shocks or rarefactions waves.

- Across wave 2, only density jumps. Velocity and pressure are constant. This defines the *contact discontinuity*.

- The characteristic curve associated with this linear wave is $dx/dt = u$, and it is a straight line. Since v_x is constant across this wave, the flow is neither converging or diverging.

Euler Equations: Riemann Problem

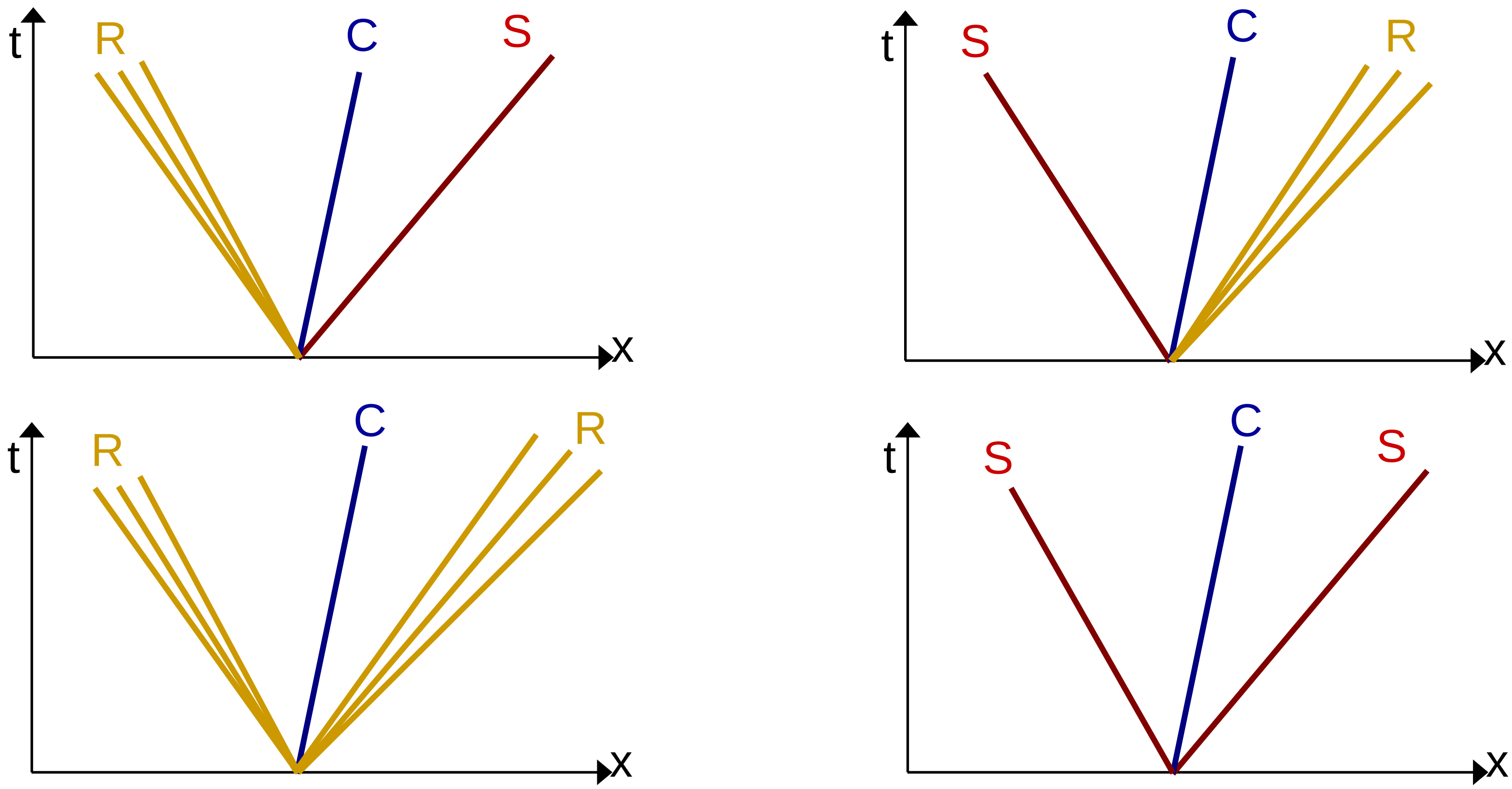
- The solution to the Riemann problem looks like



- The outer waves can be either shocks or rarefactions.
- The middle wave is always a contact discontinuity.
- In total one has 4 unknowns: $\rho_L^*, \rho_R^*, v_x^*, p^*$ since only density jumps across the contact discontinuity.

Euler Equations: Riemann Problem

- Depending on the initial discontinuity, a total of 4 patterns can emerge from the solution:



Euler Equations: Shock Tube Problem

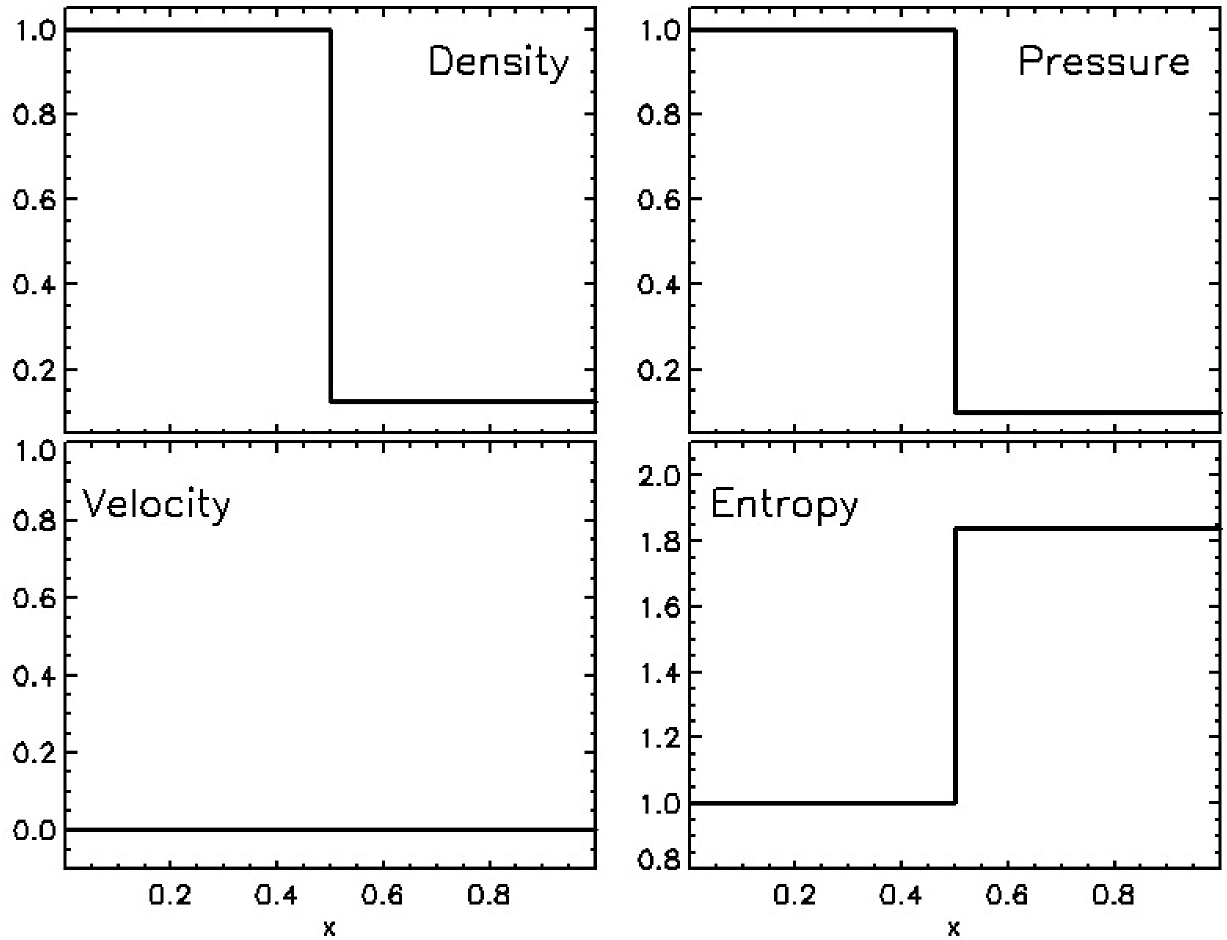
- The decay of the discontinuity defines what is usually called the "shock tube problem",

- Left Values:

$$(\rho_L, v_{xL}, p_L) = (1, 0, 1)$$

- Right Values:

$$(\rho_R, v_{xR}, p_R) = \left(\frac{1}{8}, 0, \frac{1}{10}\right)$$



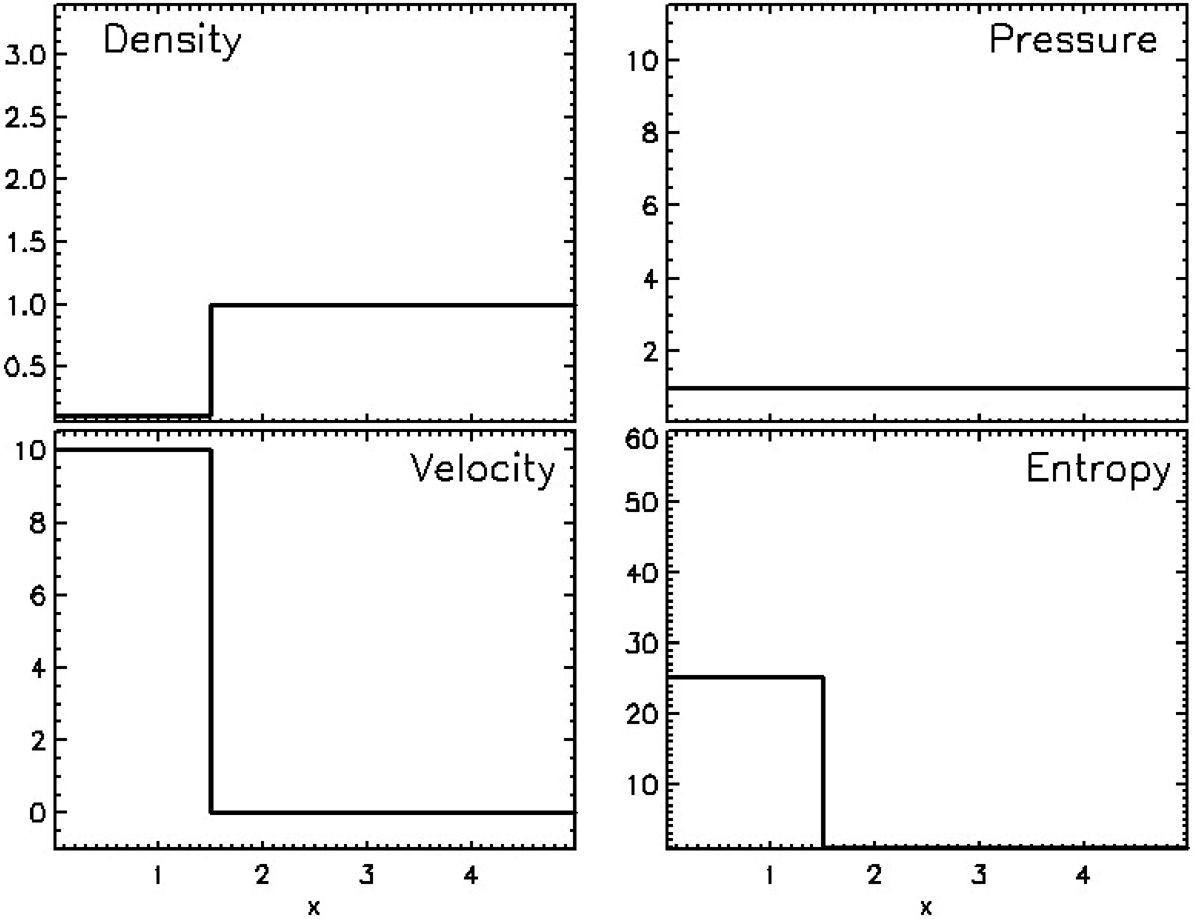
Euler Equations: Shock Tube Problem

- The one dimensional jet problem reduces to a shock-tube with a S-C-S structure:

- Left Values:

$$(\rho_L, v_{xL}, p_L) = (0.1, 10, 1)$$

$$(\rho_R, v_{xR}, p_R) = (1, 0, 1)$$



Code Example

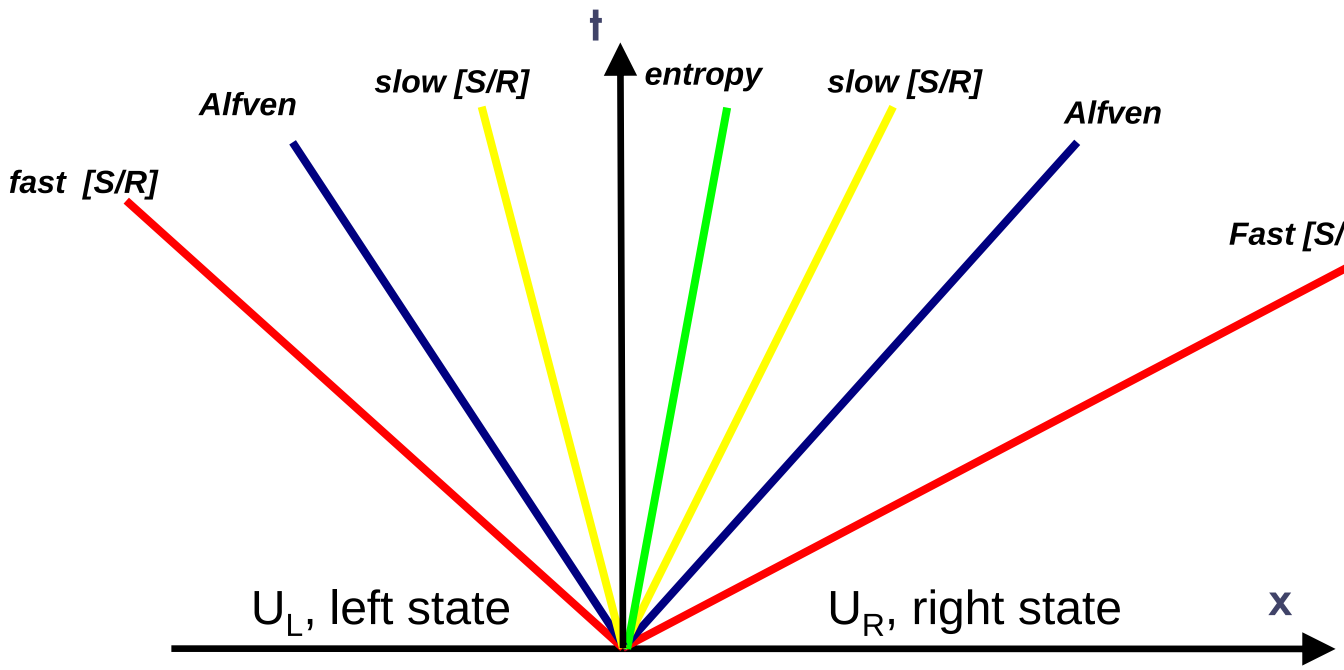
- ***File name***: `euler.f`
- ***Purpose***: solve 1D Euler's equation using a 1st-order Lax-Friedrichs method.
- ***Usage***:

```
> gfortran –O euler.f –o euler
> ./euler
```

- ***Output***: 4-column ascii data files "data.out"

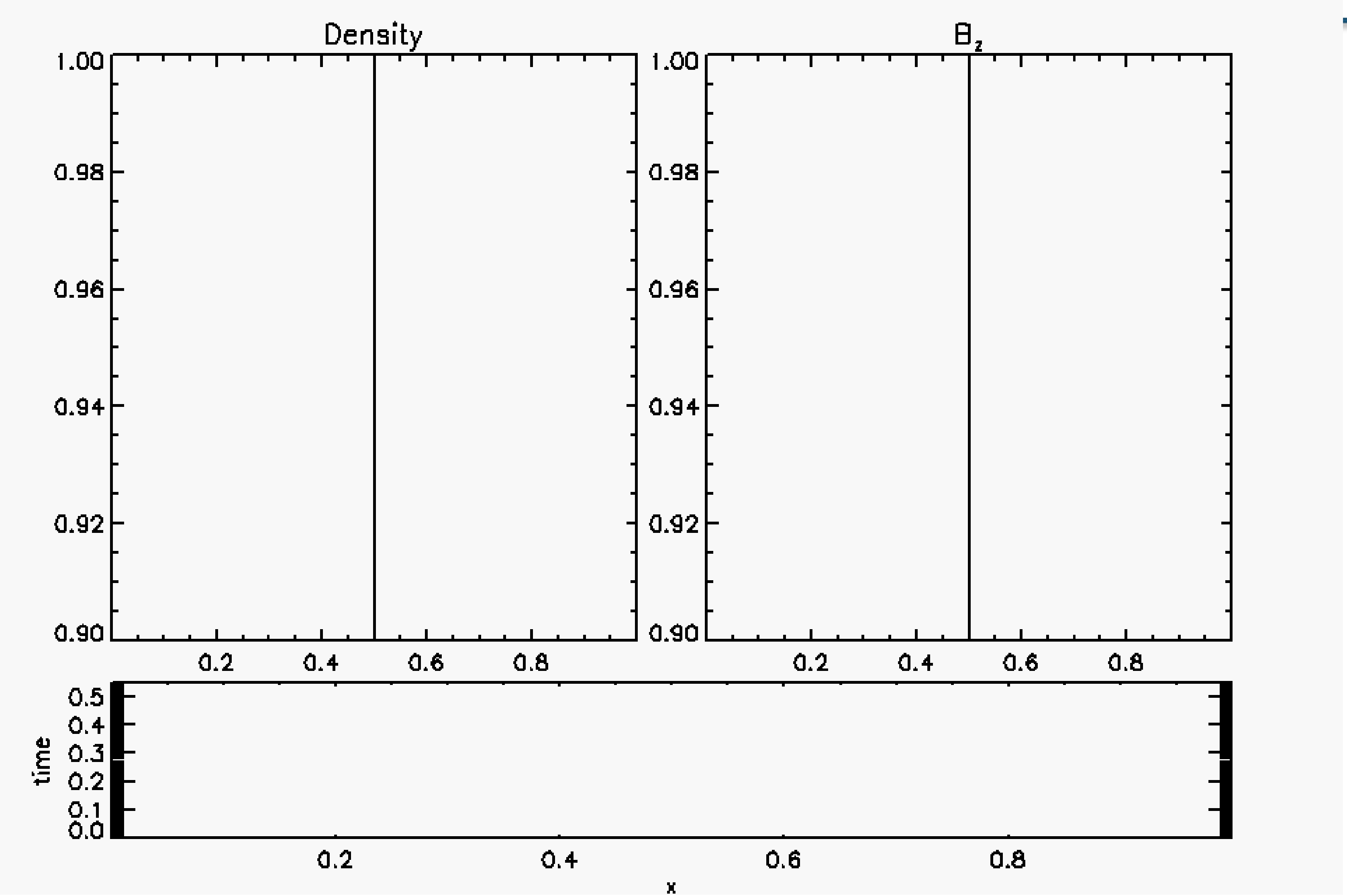
```fortran
      program euler

      include 'common.h'

      integer  i, nt, nv
      integer  ibeg, iend
      real*8   u(nvar, nx),v(nvar, nx), flux(nvar, nx)
      real*8   x(nx)
      real*8   t, dt, cmax, cfl, tstop
      real*8   tfreq, df, dx

c ** generate grid **

      call grid (x, dx)
      ibeg = nghost + 1
      iend = nx - nghost

      call init (v, x)
      call primtocon (v, u, ibeg, iend)

      dt    = 1.d-4
      cfl   = 0.8d0
      tstop = 0.2
      t     = 0.d0

c ** begin computation **

      do nt = 1, 9999

```

Riemann Problem in MHD



- 7 wave pattern, $\lambda^{(\kappa)}\left(\boldsymbol{U}_L^{(\kappa)}-\boldsymbol{U}_R^{(\kappa)}\right)=\boldsymbol{F}\left(\boldsymbol{U}_L^{(\kappa)}\right)-\boldsymbol{F}\left(\boldsymbol{U}_R^{(\kappa)}\right)$
- across the contact wave, for $B_n \neq 0$, only density has a jump;
- across Alfven waves, $[\rho]=[p_{gas}]=[v_x]=0$

An example



Density
1.00
0.98
0.96
0.94
0.92
0.90
0.2
0.4
0.6
0.8
B_z
1.00
0.98
0.96
0.94
0.92
0.90
0.2
0.4
0.6
0.8
time
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
x

Solving the Riemann Problem

- The full analytical solution to the Riemann problem for the Euler equation can be found, but this is a rather complicated task (see the book by Toro).

- In general, approximate methods of solution are preferred.

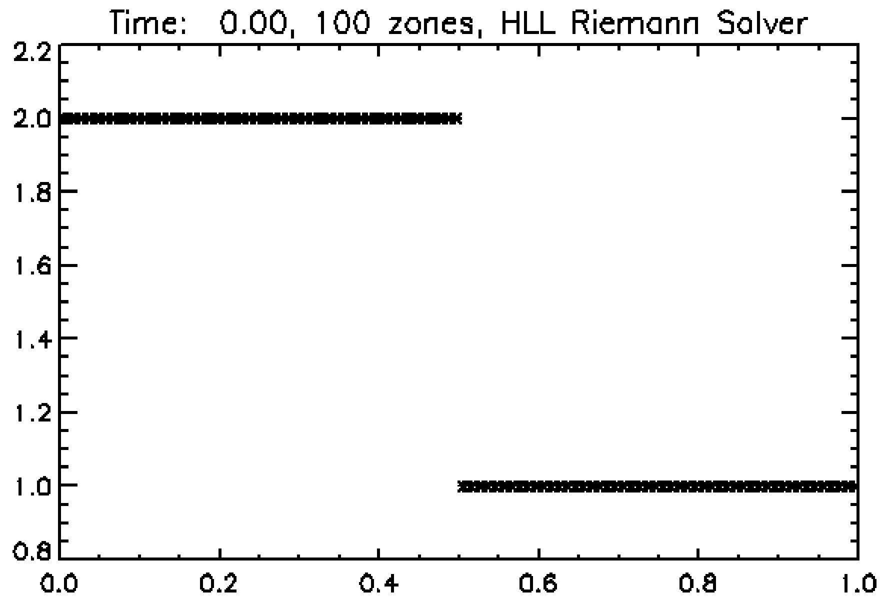
- The advantage of using approximate solvers is the reduced computational costs and the ease of implementation.

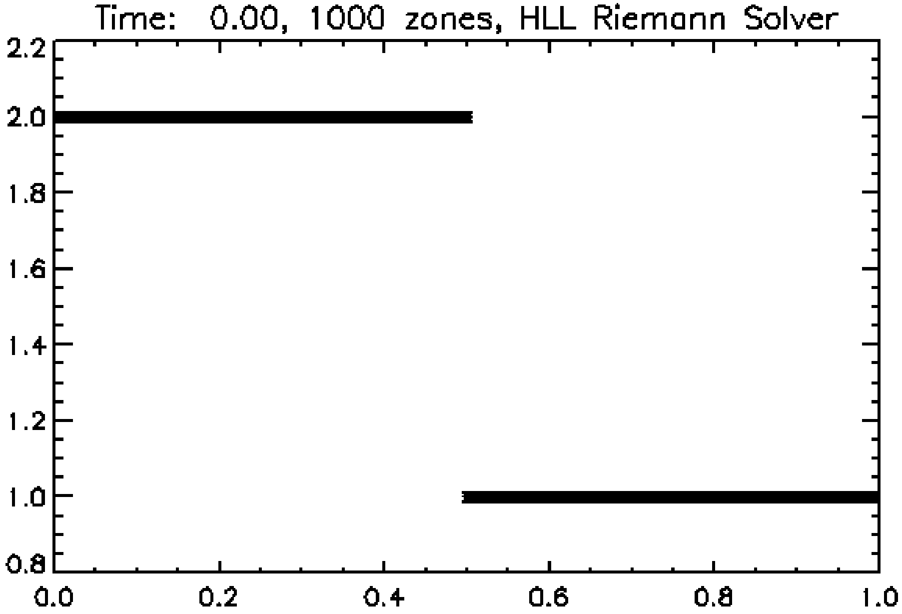
- The degree of approximation reflects on the ability to "capture" and spread discontinuities over few or more computational zones.

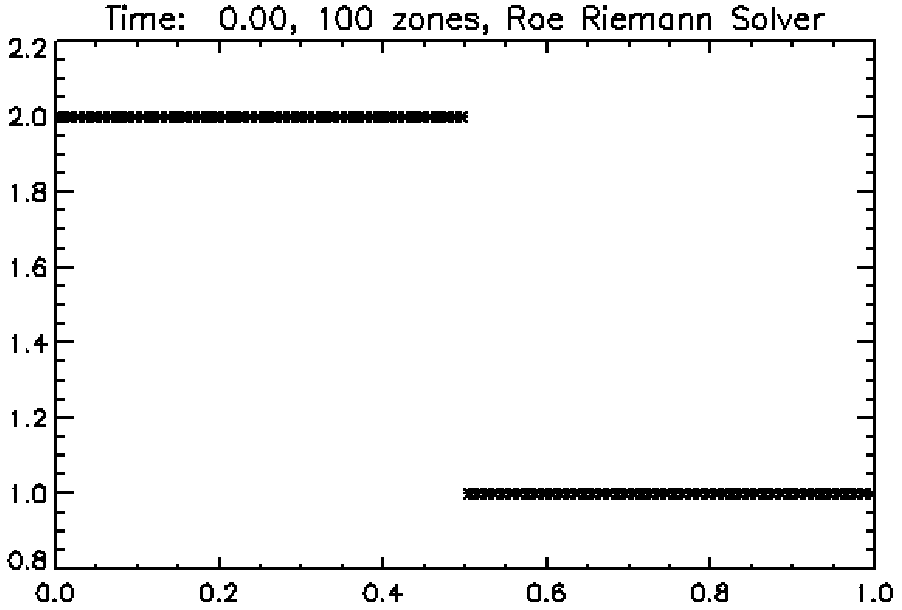
Solving the Riemann Problem

- *Exact Riemann solvers* (nonlinear)
  - Full nonlinear solution:
  - Expensive / impracticable for heavily usage in upwind codes;
- *Linearized Riemann solvers* (Roe type)
  - require characteristic decomposition in eigenvectors
  - may be prone to numerical pathologies
- *HLL-type Riemann solvers* (guess-based)
  - based on guess to the signal speeds and on the integral average of the solution over the Riemann Fan;
  - fewer waves are considered in the solution;
  - preserve positivity;

Resolution of Contact Discontinuity







2f. BASIC DISCRETIZATION METHODS FOR HYPERBOLIC PDE:

HIGH-ORDER SCHEMES

High Order Integration in Time

- A simple and effective way to achieve 2nd or 3rd order accuracy in time is to treat the PDE in semi-discrete form:

$$\int \left(\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{F} \right) dV = 0 \quad \Longrightarrow \quad \frac{d\bar{\boldsymbol{q}}}{dt} = - \oint \tilde{\boldsymbol{F}} \cdot d\boldsymbol{S}$$

- In such a way the PDE becomes a regular ordinary differential equation (ODE) in time;

$$\frac{d\bar{\boldsymbol{q}}}{dt} = \boldsymbol{R}(\boldsymbol{q}, t) = \boldsymbol{R} \quad \Longrightarrow \quad \bar{\boldsymbol{q}}^{n+1} - \bar{\boldsymbol{q}}^{n} = \int_{n}^{n+1} \boldsymbol{R}\, dt$$

- Standard integration based on predictor/corrector schemes can then be used to solve ODEs.

Second-Order Runge-Kutta

- Using the trapezoidal method, the solution of our ODE writes:

$$\bar{\boldsymbol{q}}^{n+1} = \bar{\boldsymbol{q}}^{n} + \frac{\Delta t}{2}\left(\boldsymbol{R}^{n} + \boldsymbol{R}^{n+1}\right) + O(\Delta t^3)$$

- Problem: the unknown $\bar{\boldsymbol{q}}^{n+1}$ appears on both side of the equation!!!
- Solution: use an estimate (predictor) for $\boldsymbol{R}^{n+1}$ with Euler method:

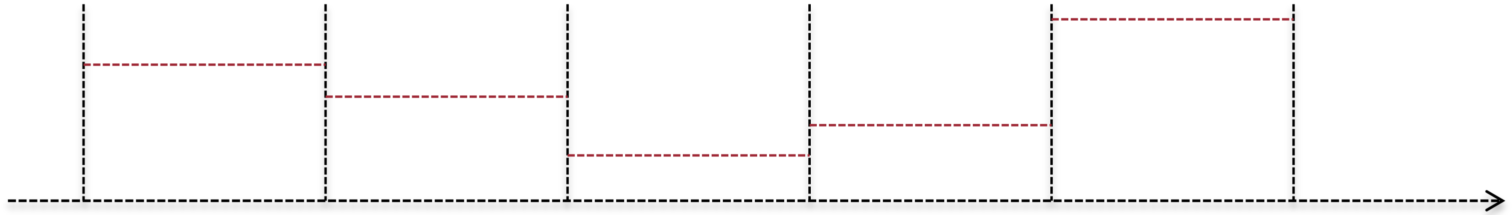
$$\begin{aligned} \bar{\boldsymbol{q}}^{*} &= \bar{\boldsymbol{q}}^{n} + \Delta t \boldsymbol{R}^{n} + O(\Delta t^2) \\ \bar{\boldsymbol{q}}^{n+1} &= \bar{\boldsymbol{q}}^{n} + \frac{\Delta t}{2}\left(\boldsymbol{R}^{n} + \boldsymbol{R}^{*}\right) + O(\Delta t^3) \end{aligned}$$

- This is the second-order explicit Runge-Kutta method (or Heun's method) It is 2nd order accurate.

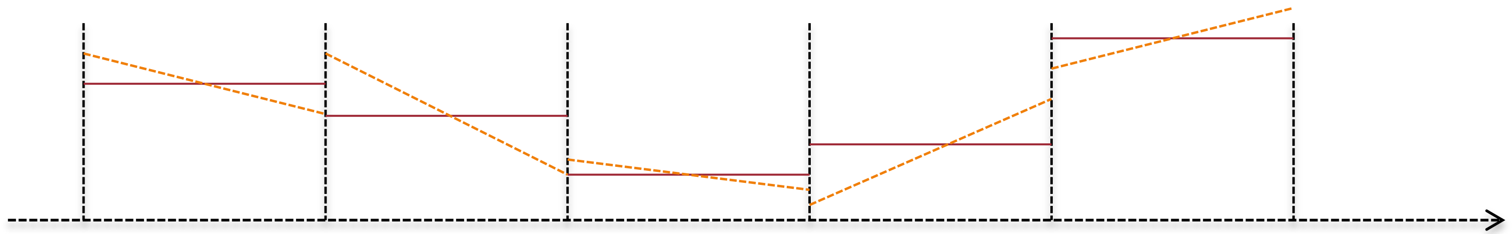
Improving spatial accuracy

➢ High order reconstruction can be carried inside each cell by suitable oscillation-free polynomial interpolation:

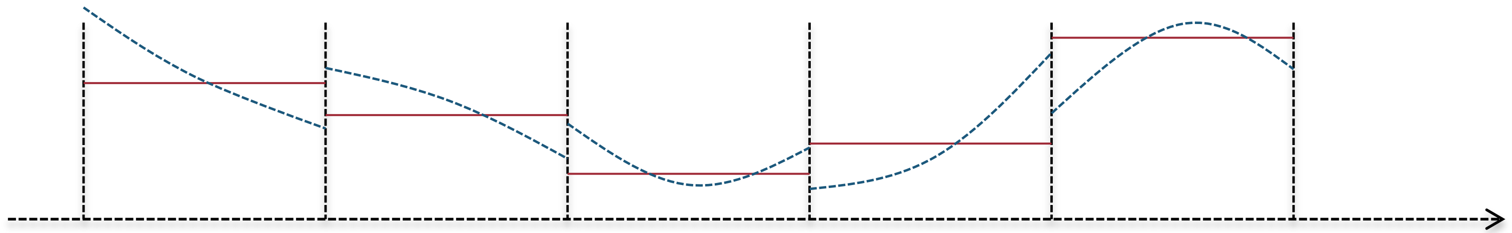
Piecewise constant



Piecewise Linear



Piecewise parabolic



Reconstruction Constraints

- Must be consistent with data representation

$$\frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} P_i(x)dx = \bar{u}_i$$

- Satisfy monotonicity constraints:

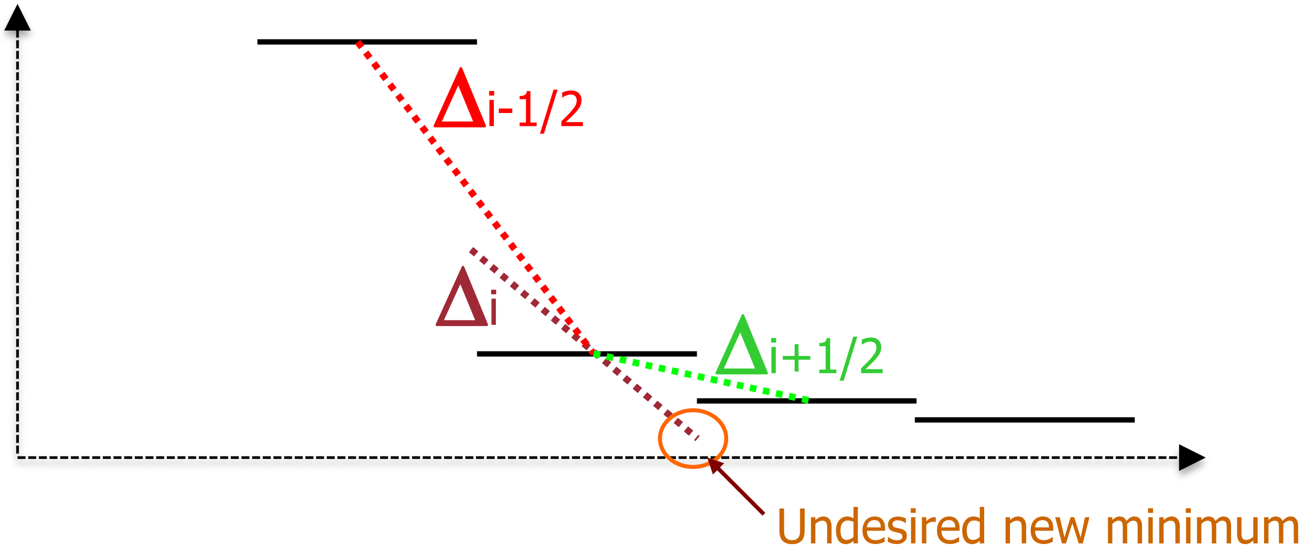
$$\min(P_i(x)) \geq \min\left(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}\right)$$
$$\max(P_i(x)) \leq \max\left(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}\right)$$

- no new extrema allowed (Total Variation Diminishing (TVD) schemes)
- Oscillation free solution

Example: 2nd order linear reconstruction

- For 2nd-order interpolant, we use

$$V(x) = V_i + \frac{\delta V}{\Delta x}(x - x_i)$$



- Use slope limiters to avoid introducing new extrema:

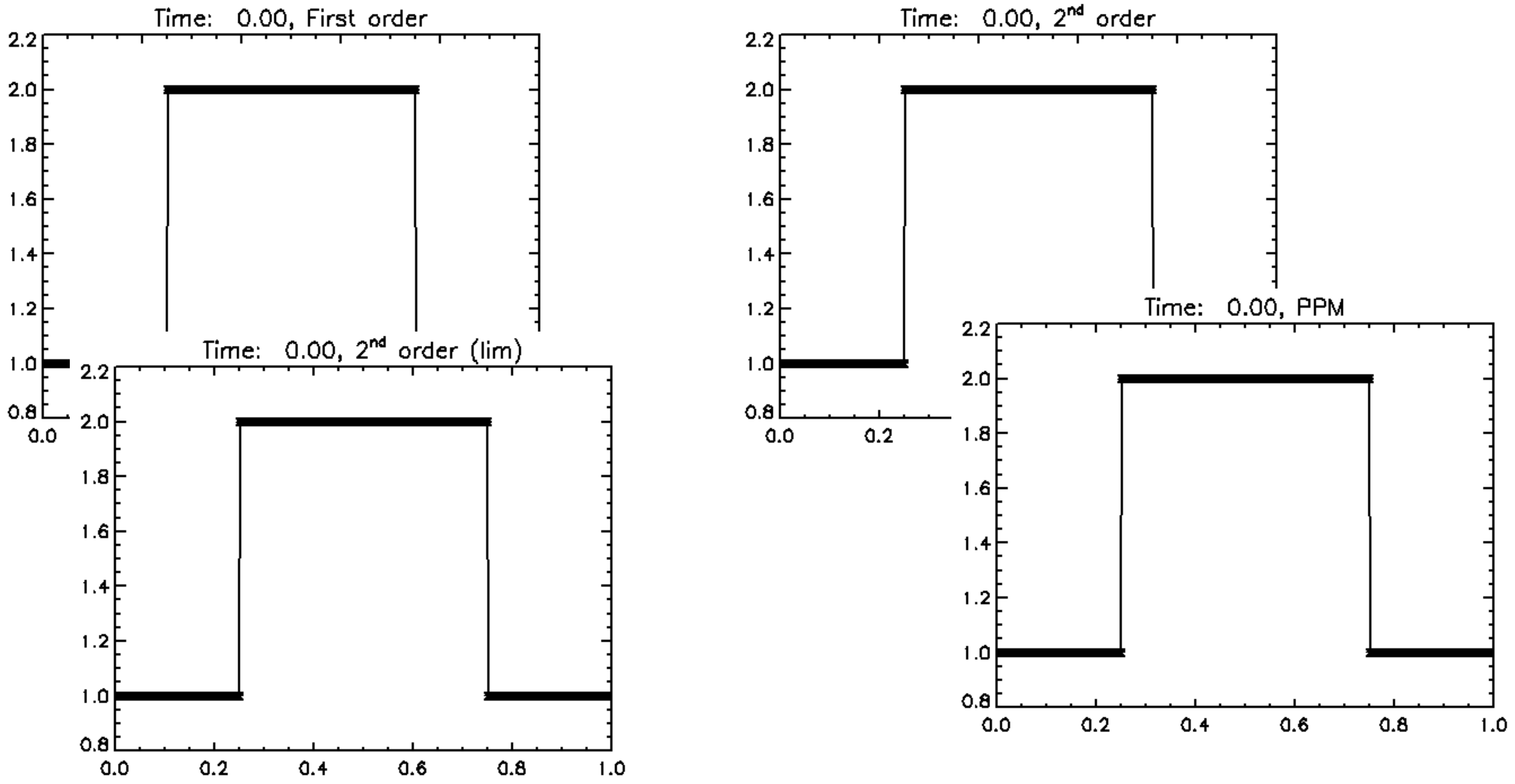
$$\delta V_i = \lim\left(\Delta_{\mathrm{i}-1/2}, \Delta_{\mathrm{i}+1/2}\right)$$

- Example

$$\mathrm{minmod}(x, y) = \begin{cases} x & \text{if} \quad |x| < |y|, xy > 0 \\ y & \text{if} \quad |y| < |x|, xy > 0 \\ 0 & \text{if} \quad xy < 0 \end{cases}$$

Comparison

- Improving reconstruction decreases the amount of numerical dissipation:



Equivalent Advection/Diffusion Equation

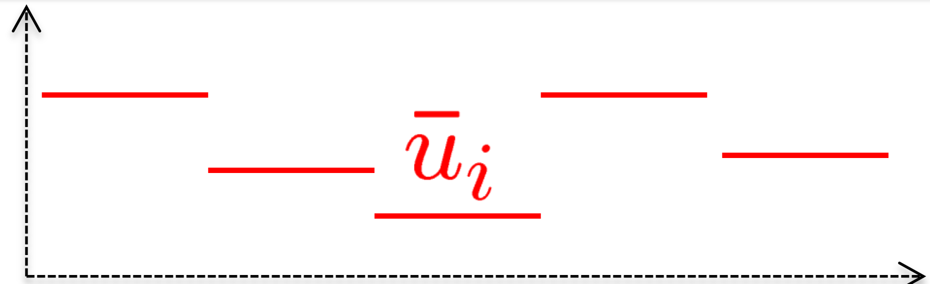
- A discretized PDE gives the exact solution to an equivalent equation with a diffusion term;

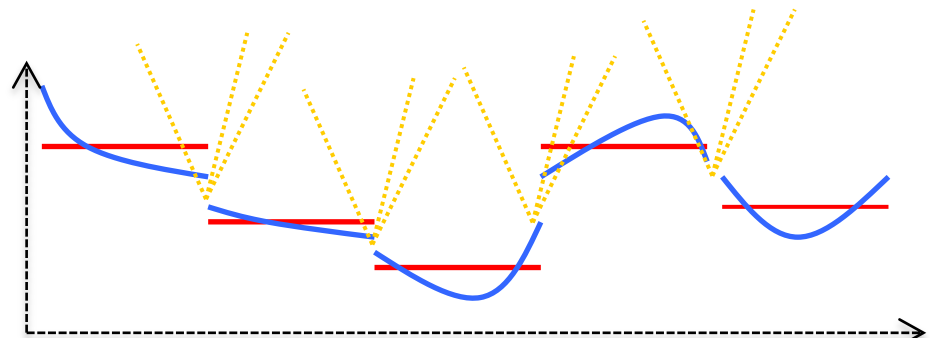
- Consider $$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0\,, \qquad a > 0$$
  - Use upwind discretization: $$\frac{q_i^{n+1} - q_i^n}{\Delta t} + a\frac{q_i^n - q_{i-1}^n}{\Delta x} = 0$$
  - Do Taylor expansion on q_i^{n+1} and q_{i-1}^n
  - The solution to the discretized equation satisfies *exactly*

$$\frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = \frac{a\Delta x}{2}\left(1 - a\frac{\Delta t}{\Delta x}\right)\frac{\partial^2 q}{\partial x^2} + H.O.T.$$

Algorithm Summary: Reconstruct-Solve-Average (RSA)

- Start from zone averages, break the problem into 3 pieces:





1. Piecewise polynomial reconstruction

$$u_i(x) = P_i(x)\,, \text{ for } x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}}$$

2. Solve Riemann problem between left and right states

$$\begin{cases} u_L = P_i\left(x_{i+\frac{1}{2}}\right) \\ u_R = P_{i+1}\left(x_{i+\frac{1}{2}}\right) \end{cases} \longrightarrow f_{i+\frac{1}{2}}$$

3. Form new averages (evolve)

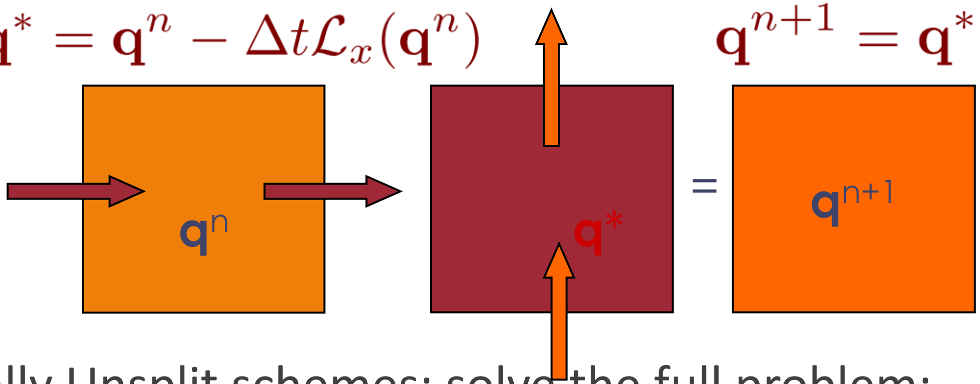
$$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\Delta t}{\Delta x}\left[f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right]$$

Multi Dimensional Integration

- Integration in more than one dimensions can be achieved using two distinct approaches:

  - Dimensionally Split schemes: solve the PDE as a sequence of 1-D sub-problems.

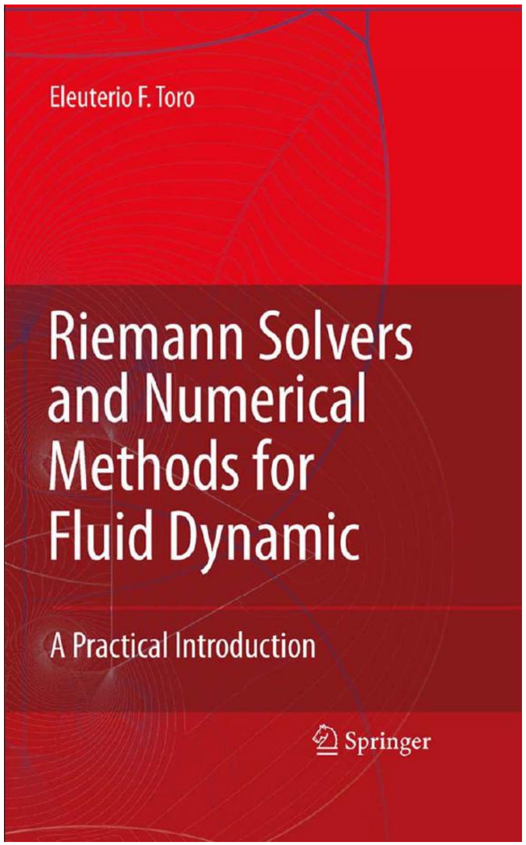
$$\mathbf{q}^* = \mathbf{q}^n - \Delta t \mathcal{L}_x(\mathbf{q}^n) \qquad \mathbf{q}^{n+1} = \mathbf{q}^* - \Delta t \mathcal{L}_y(\mathbf{q}^*)$$

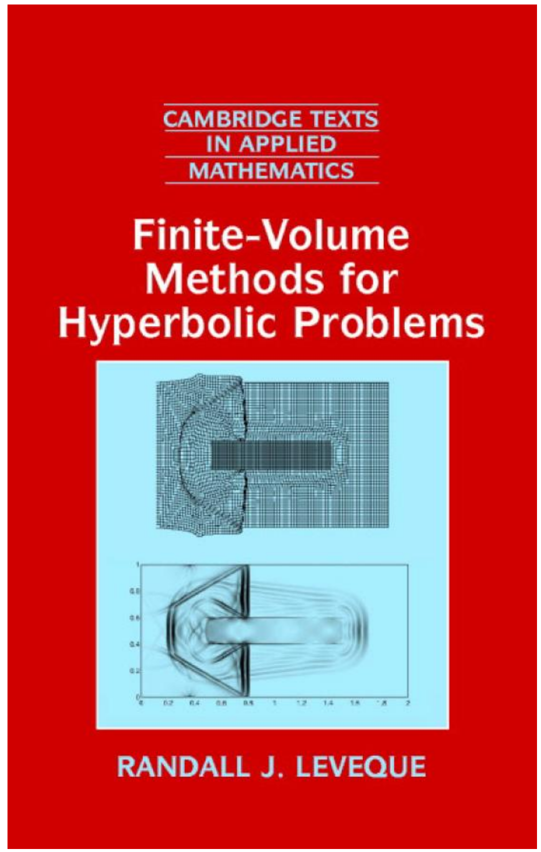


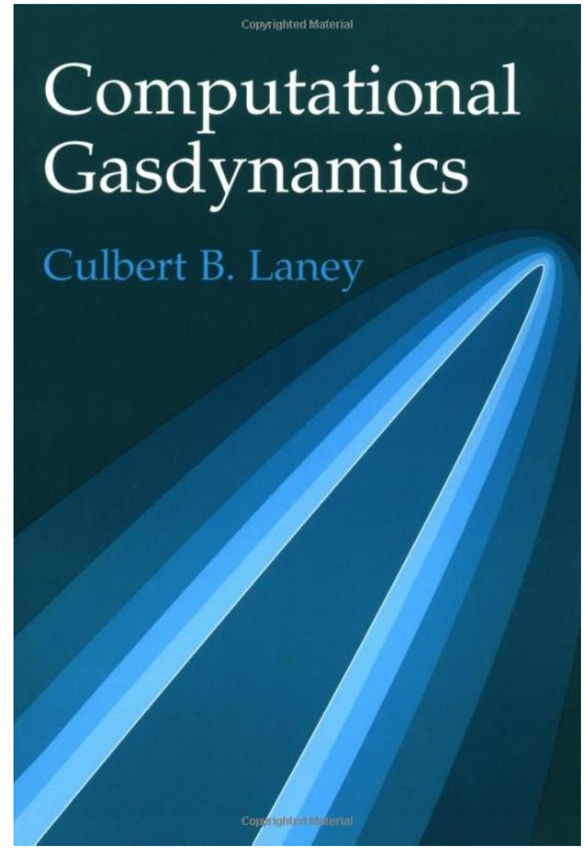
  - Dimensionally Unsplit schemes: solve the full problem:

$$\mathbf{q}^{n+1} = \mathbf{q}^n - \Delta t \mathcal{L}_x(\mathbf{q}^n) - \Delta t \mathcal{L}_y(\mathbf{q}^n)$$

Useful Books







The End