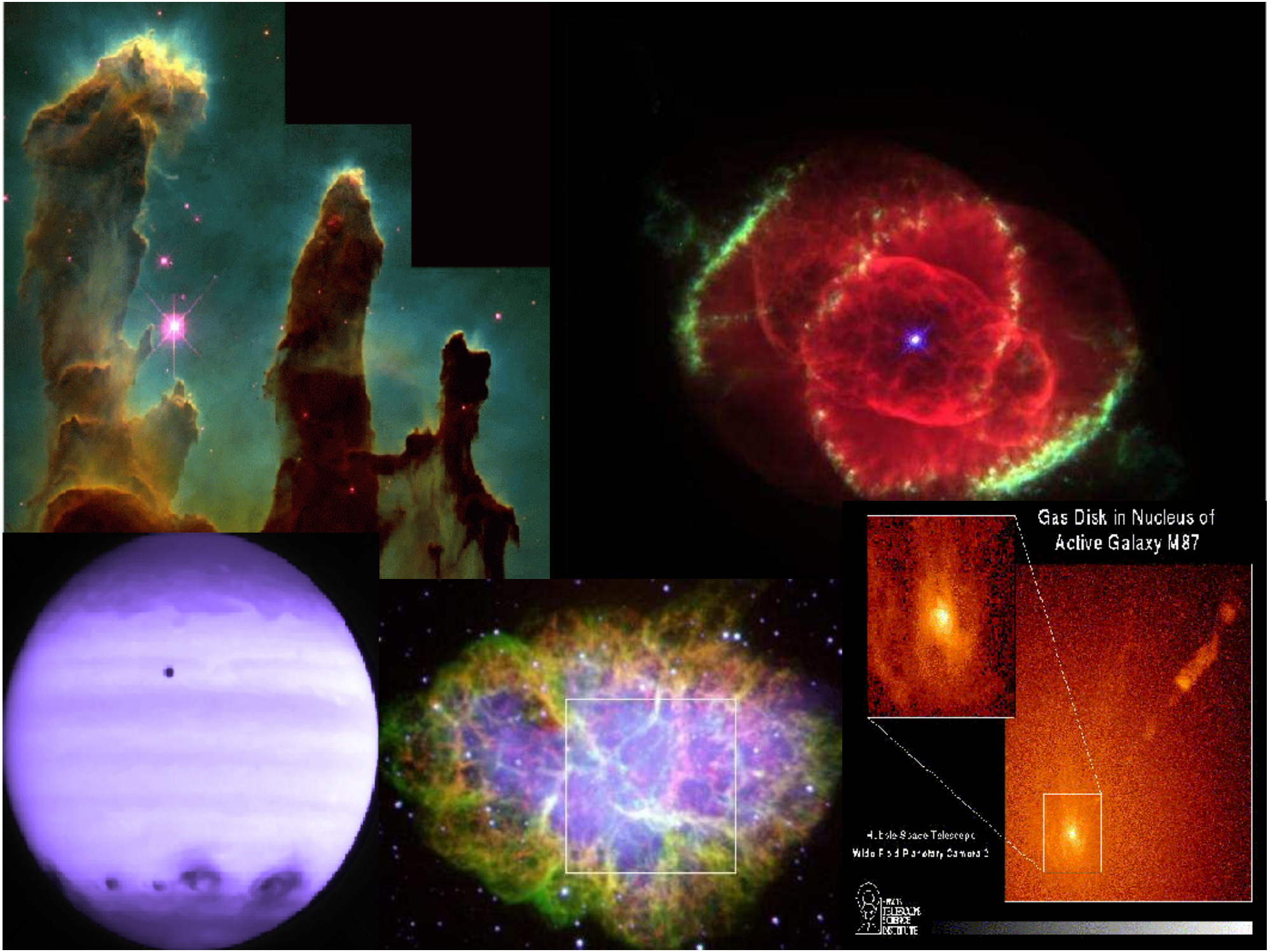
Algorithms to solve the magneto- hydrodynamical equations including source terms

Contents

- The non- ideal (M) H D equations
- Splitting into different parts
- Algorithms for the ideal equations (hyperbolic part)
- A note on div (B) = 0
- Algorithms for the viscous, diffusive and resistive equations (parabolic part)


Gas Disk in Nucleus of
Active Galaxy M87

Modeling results in a complex system of equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}(\rho \boldsymbol{v}) = 0$$

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}\left[\rho \boldsymbol{v} \cdot \boldsymbol{v} + p \boldsymbol{I} - \theta_{\nu_v}\right] = \boldsymbol{f}(\boldsymbol{x}, t)$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{v}(E + p) - \theta_{\nu_v} \cdot \boldsymbol{v} - \kappa_C T^{5/2} \frac{\partial T}{\partial \boldsymbol{x}}\right] = \boldsymbol{v} \cdot \boldsymbol{f} - C + H$$

$$\frac{1}{c}\frac{\partial I^{\nu_f}}{\partial t} + (\underline{n} \cdot \nabla) I^{\nu_f} = \eta^{\nu_f} - (\kappa^{\nu_f} + \lambda^{\nu_f}) I^{\nu_f} + \lambda^{\nu_f} \oint_{S_2} d\Omega\, I^{\nu_f}$$

$$\frac{\partial Z_i}{\partial t} = R(Z^i, I, \boldsymbol{x}, t) \qquad \Rightarrow f_R, C_R, H_R$$

$$\Delta \Phi = \rho \quad \Rightarrow f_G = \nabla \Phi$$

Compressible Navier- Stokes equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}(\rho \boldsymbol{v}) = 0$$
$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}[\rho \boldsymbol{v} \cdot \boldsymbol{v} + p \boldsymbol{I} - \theta_{\nu_v}] = \boldsymbol{f}(\boldsymbol{x}, t)$$
$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}[\boldsymbol{v}(E + p) - \theta_{\nu_v} \cdot \boldsymbol{v} - \kappa_C T^{5/2} \frac{\partial T}{\partial \boldsymbol{x}}] = \boldsymbol{v} \cdot \boldsymbol{f} - C + H$$

Non- ideal MHD equations

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho \boldsymbol{v} \\ \boldsymbol{B} \\ E \end{bmatrix} + \frac{\partial}{\partial \boldsymbol{x}}\begin{bmatrix} \rho \boldsymbol{v} \\ \rho \boldsymbol{v}\boldsymbol{v} + \left(p + \frac{1}{2\mu_0} B^2\right)\boldsymbol{I} - \frac{1}{\mu_0}\boldsymbol{B}\boldsymbol{B} \\ \boldsymbol{v}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{v} \\ \boldsymbol{v}\left(E + p + \frac{1}{2\mu_0} B^2\right) - \frac{1}{\mu_0}\boldsymbol{B}\left(\boldsymbol{v} \cdot \boldsymbol{B}\right) \end{bmatrix} = \nabla \boldsymbol{B}\begin{bmatrix} 0 \\ \boldsymbol{B} \\ \boldsymbol{v} \\ \boldsymbol{v} \cdot \boldsymbol{B} \end{bmatrix} + \begin{matrix} non \\ ideal \\ terms \end{matrix}$$

Non- linear advection- diffusion equation with an inhomogeneity

$$\frac{\partial \boldsymbol{q}(\boldsymbol{x},t)}{\partial t}+\frac{\partial}{\partial \boldsymbol{x}}\left(\boldsymbol{f}^{\boldsymbol{adv}}(\boldsymbol{q},\boldsymbol{x},t)+\boldsymbol{f}^{\boldsymbol{diff}}(\boldsymbol{q},\boldsymbol{x},t)\right)=S(\boldsymbol{q},\boldsymbol{x},\boldsymbol{t})$$

We split into three parts

- $\frac{\partial \boldsymbol{q}(\boldsymbol{x},t)}{\partial t}+\frac{\partial}{\partial \boldsymbol{x}}\left(\boldsymbol{f}^{\boldsymbol{adv}}(\boldsymbol{q},\boldsymbol{x},t)\right)=0$ hyperbolic equation
- $\frac{\partial \boldsymbol{q}(\boldsymbol{x},t)}{\partial t}+\frac{\partial}{\partial \boldsymbol{x}}\left(\boldsymbol{f}^{\boldsymbol{diff}}(\boldsymbol{q},\boldsymbol{x},t)\right)=0$ parabolic equation
- $\frac{\partial \boldsymbol{q}(\boldsymbol{x},t)}{\partial t}=S(\boldsymbol{q},\boldsymbol{x},\boldsymbol{t})$ ODE

fractional steps for evolving the system for one timestep: $t_n \longmapsto t_{n+1} = t_n + \Delta t$

solve ODE for the sources, $\partial q/\partial t = S$, with initial data $\underline{q}^n$ for time-step $\Delta t/2$ to obain $q^{n+1/2}$

solve homogeneous advection equations with initial data $\underline{q}^{n+1/2}$ = for a time- step $\Delta t/2$ to obtain $(\underline{q'})^{n+1/2}$

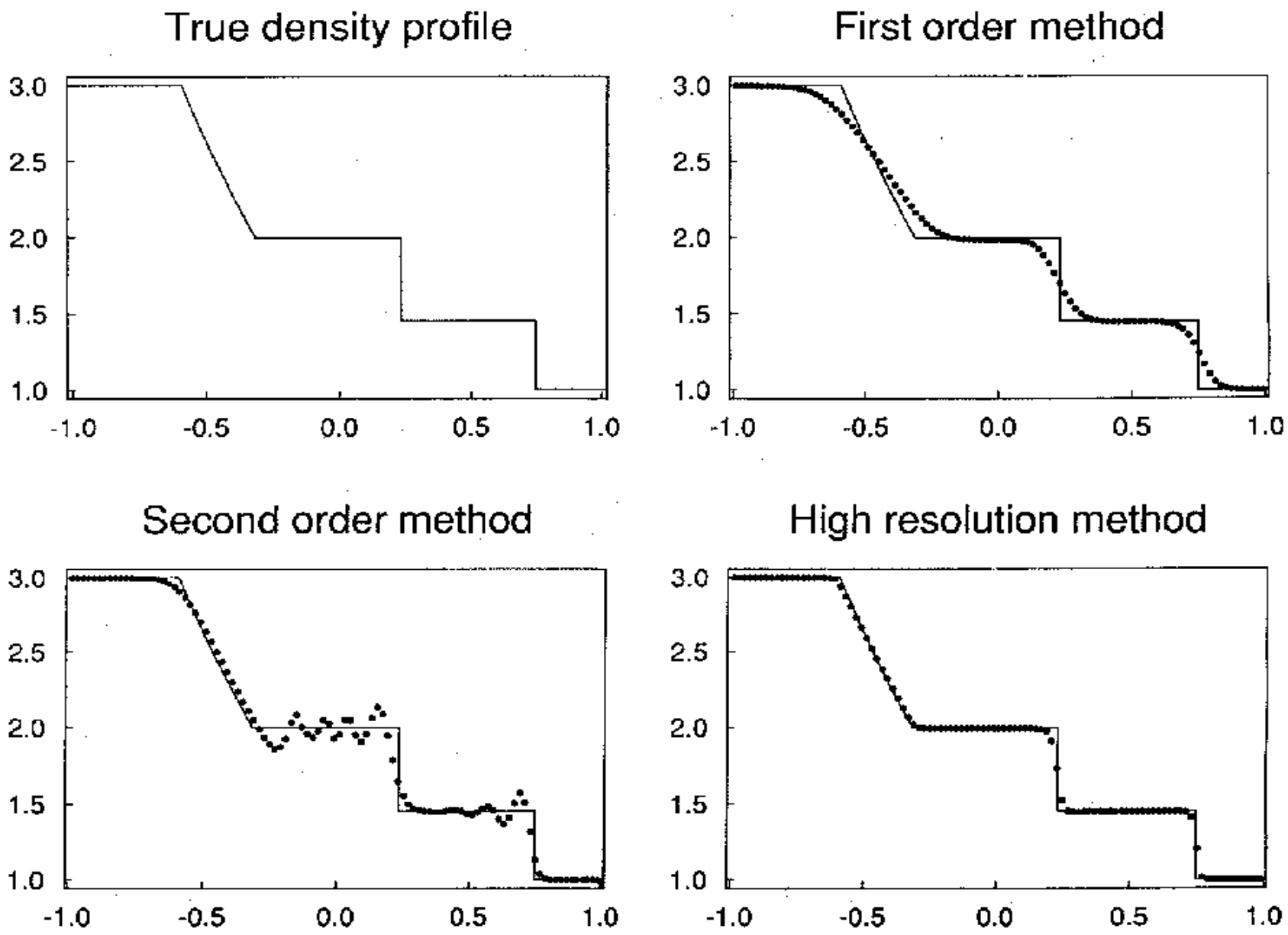
solve homogeneous diffusion equations with initial data $(\underline{q'})^{n+1/2}$ = for a time- step Δt to obtain $(\underline{q''})^{n+1/2}$

solve homogeneous advection equations with initial data $(q'')^{n+1/2}$ for time- step $\Delta t/2$ to obtain $(\underline{q'''})^{n+1/2}$

solve ODE for the sources, $\partial q/\partial t = S$, with initial data $(\underline{q'''})^{n+1/2}$ for time- step $\Delta t/2$ to obain q^{n+1}

Algorithms for the ideal equations (hyperbolic part)

$$\frac{\partial \boldsymbol{q}(\boldsymbol{x}, t)}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{f}^{\boldsymbol{adv}} \left(\boldsymbol{q}, \boldsymbol{x}, t \right) \right) = 0$$



Lax- Wendroff, Lax- Friedrichs

non- stable methods:
central differences ---> Lax-Wendroff, Lax- Friedrichs
- saves:
  artificial viscosity stabileizes ---> will come back

pectral methods: osciallatory at discontinuities

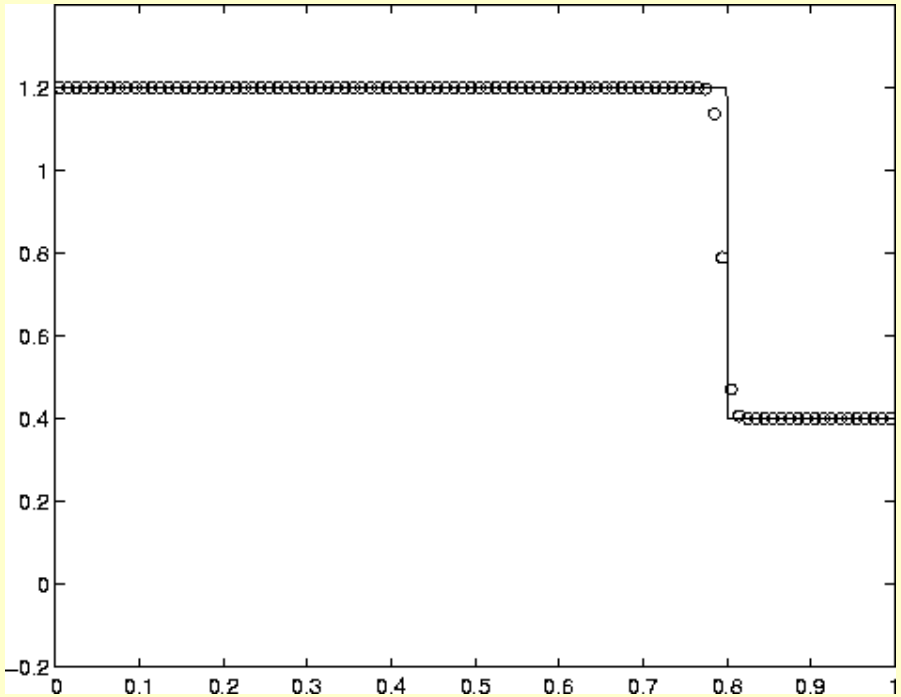
finite elements: stable methods are under development

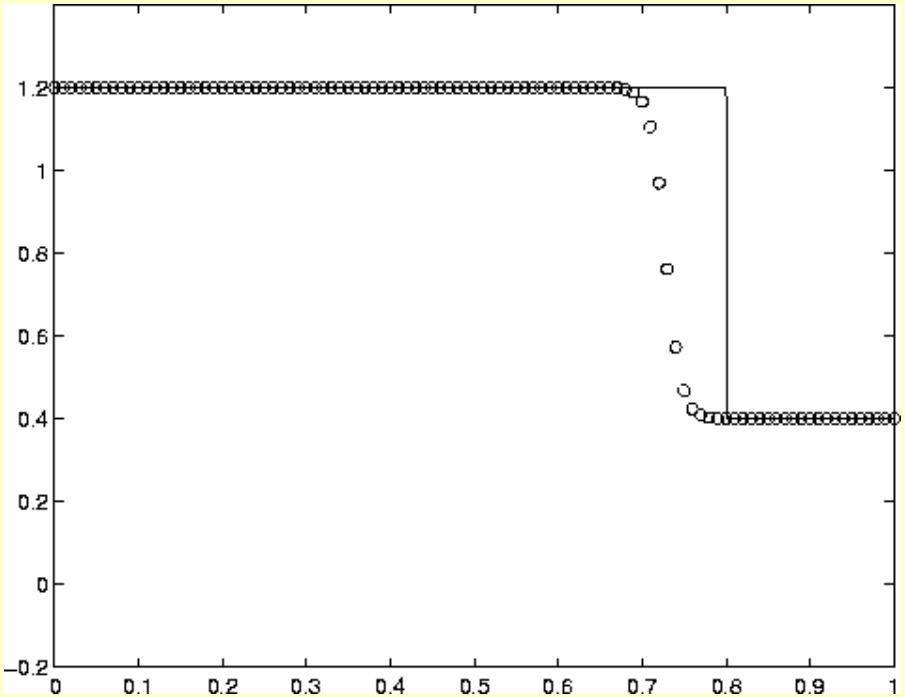
requirements:

- upwinding
- total variation diminishing (TVD) ---> non- oscillatory
- conservativity ----> finite volumes ---> shock capturing

The importance of a conservative discretization

Burgers equations

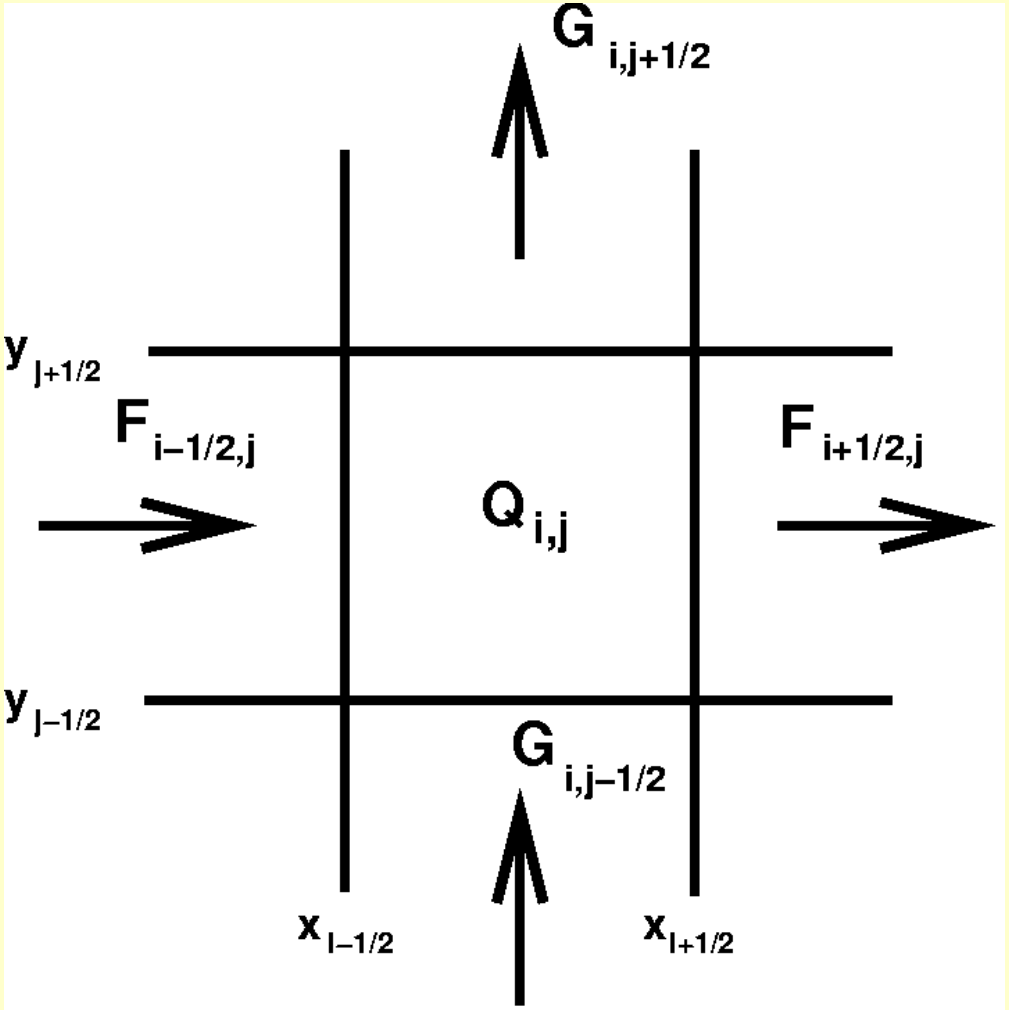




$$\frac{\partial q}{\partial t}+\frac{1}{2}\frac{\partial\left(q^2\right)}{\partial x}=0$$

$$\frac{\partial q}{\partial t}+q\frac{\partial\left(q\right)}{\partial x}=0$$

non- classical solution ----> at discontinuity there exists only a weak solution which is different for the two formulations



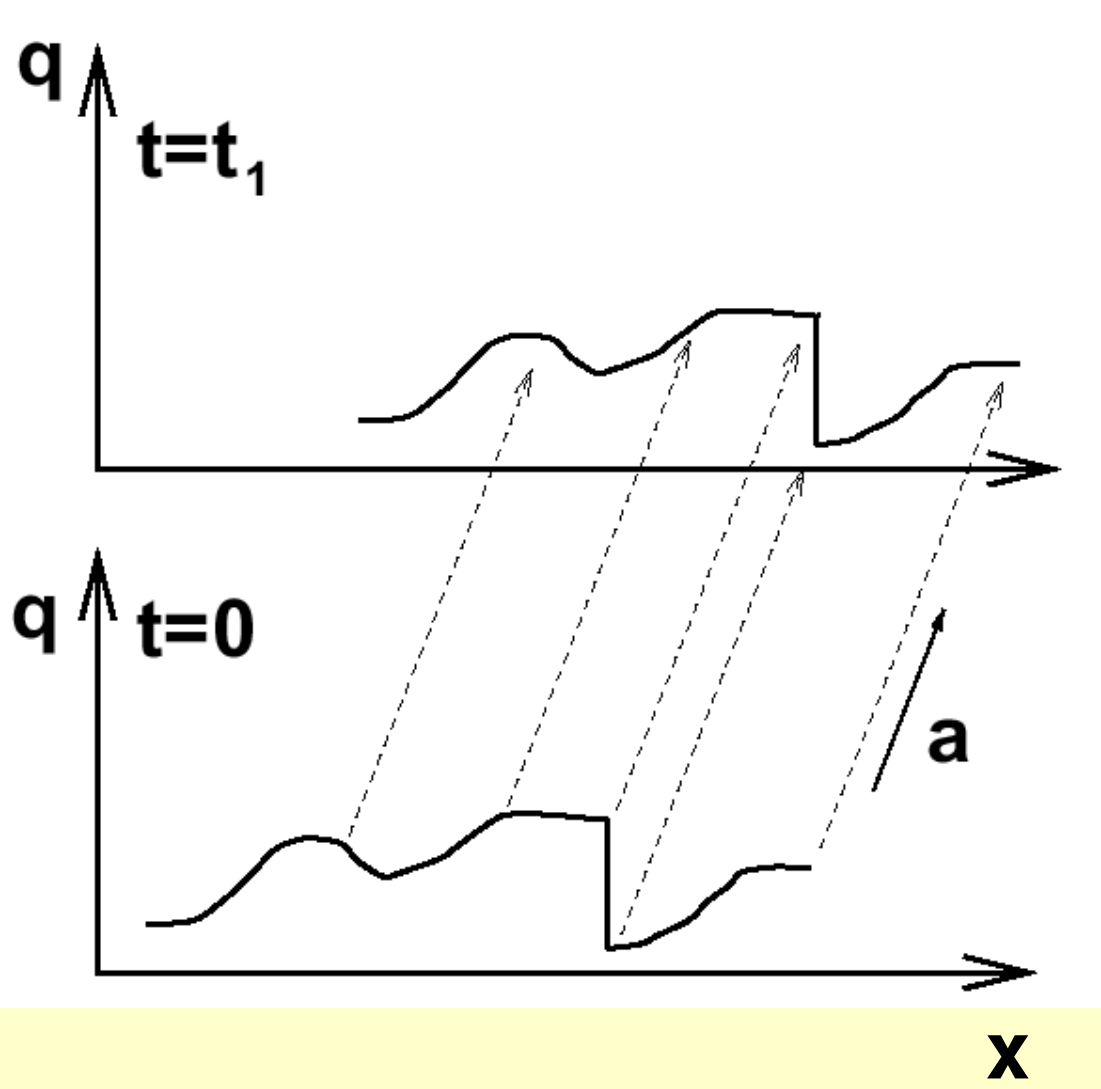
Finite volumes:

'natural' conservativity

But how to compute the fluxes?

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \frac{dt}{A}\left[\left(F_{i+1/2,j} - F_{i-1/2,j}\right) + \left(G_{i,j+1/2} - G_{i,j-1/2}\right)\right]$$

Linear, scalar advection equation



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

$$q(x, o) = q_0(x)$$

$$f(q) = a\,q, \quad a = const$$

simple solution!

$$q(x, t) = q_0(x - at, 0)$$

Linear hyperbolic (advection) systems

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

$$\boldsymbol{q}(x, o) = \boldsymbol{q_0}(x)$$

$$\boldsymbol{f}(\boldsymbol{q}) = A\,\boldsymbol{q}, \quad A = \begin{bmatrix} n & x & n \end{bmatrix} - matrix = const$$

Definition: This system is called hyperbolic <-- >
A is diagonizable with real eigenvalues

$$A\,r_p = \lambda_p\, r_p$$

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

$$\boldsymbol{q}(x, o) = \boldsymbol{q_0}(x)$$

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

$$\boldsymbol{v}(x, o) = \boldsymbol{v_0}(x)$$

equation in its characteristic form

$$A r_p = \lambda_p r_p$$

$$\Lambda \equiv diag\left(\lambda_1, \lambda_2, \ldots, \lambda_n\right)$$

$$R \equiv \left[r_1 | r_2 | r_3 | \ldots | r_n\right]$$

$$\boldsymbol{v} \equiv R^{-1}\boldsymbol{q}$$

characteristic variables

which results in m independent scalar equations with solution:

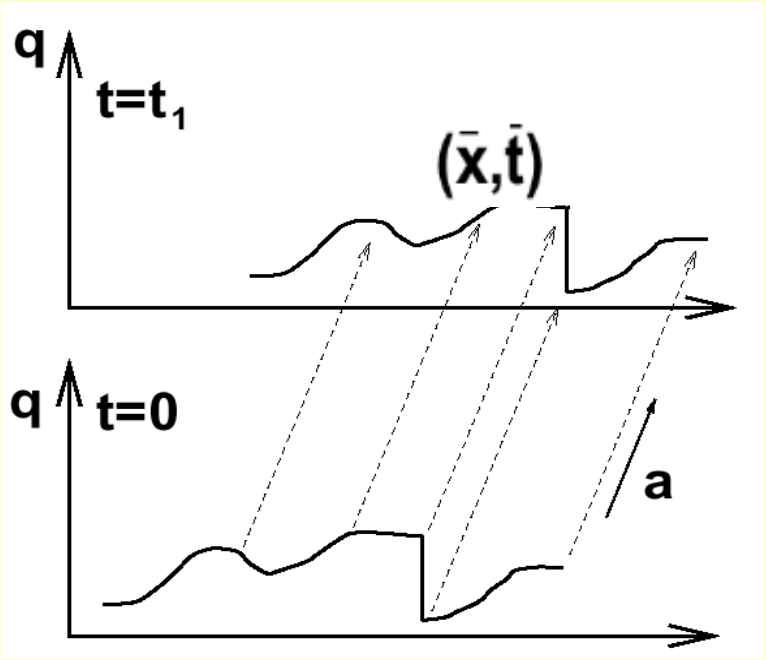
$$v_p(x, t) = v_p(x - \lambda_p t, 0) \equiv \left(R^{-1} q_0(x)\right)_p \quad \forall\, p$$

$$\frac{\partial \boldsymbol{q}}{\partial t} + A \frac{\partial \boldsymbol{q}}{\partial x} = 0 \quad \Rightarrow \quad \frac{\partial \boldsymbol{v}}{\partial t} + \Lambda \frac{\partial \boldsymbol{v}}{\partial x} = 0$$

$$\left(\quad v_p(x,t) = v_p(x - \lambda_p t, 0) \equiv \left(R^{-1} q_0(x) \right)_p \quad \forall\, p \quad \right)$$

$$\boldsymbol{q}(x,t) = R\boldsymbol{v}(x,t) = \sum_{p=1}^{m} v_p(x,t) r_p = \sum_{p=1}^{m} v_p(x - \lambda_p t, 0) r_p$$

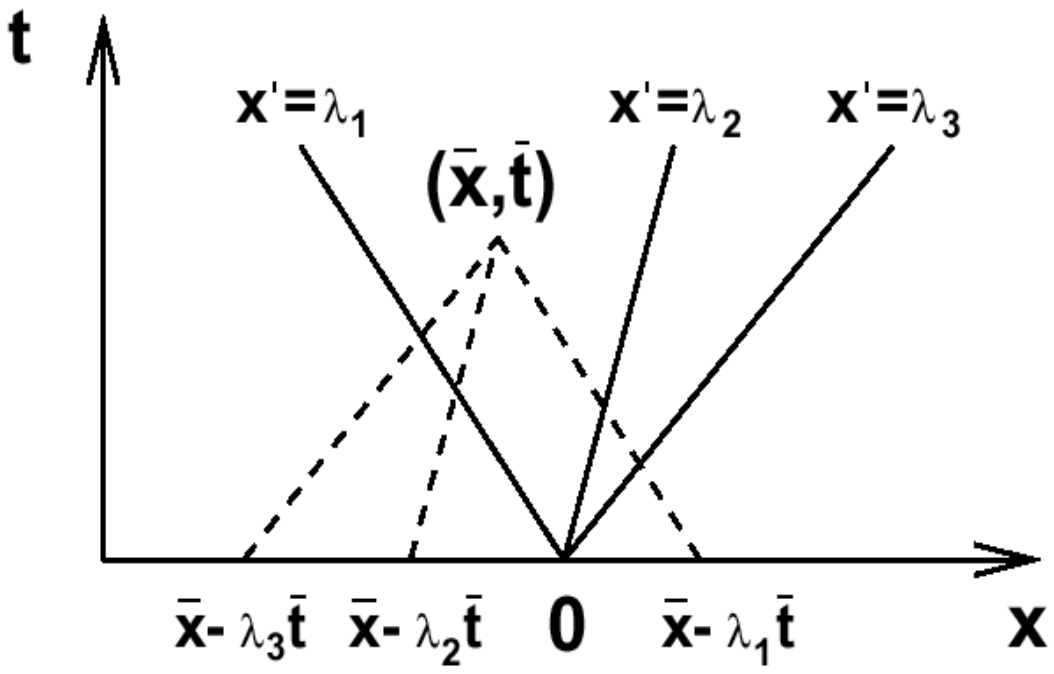
we can analytically solve
if we know the eigenvalues and the (right) eigenvectors



scalar advection equation

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

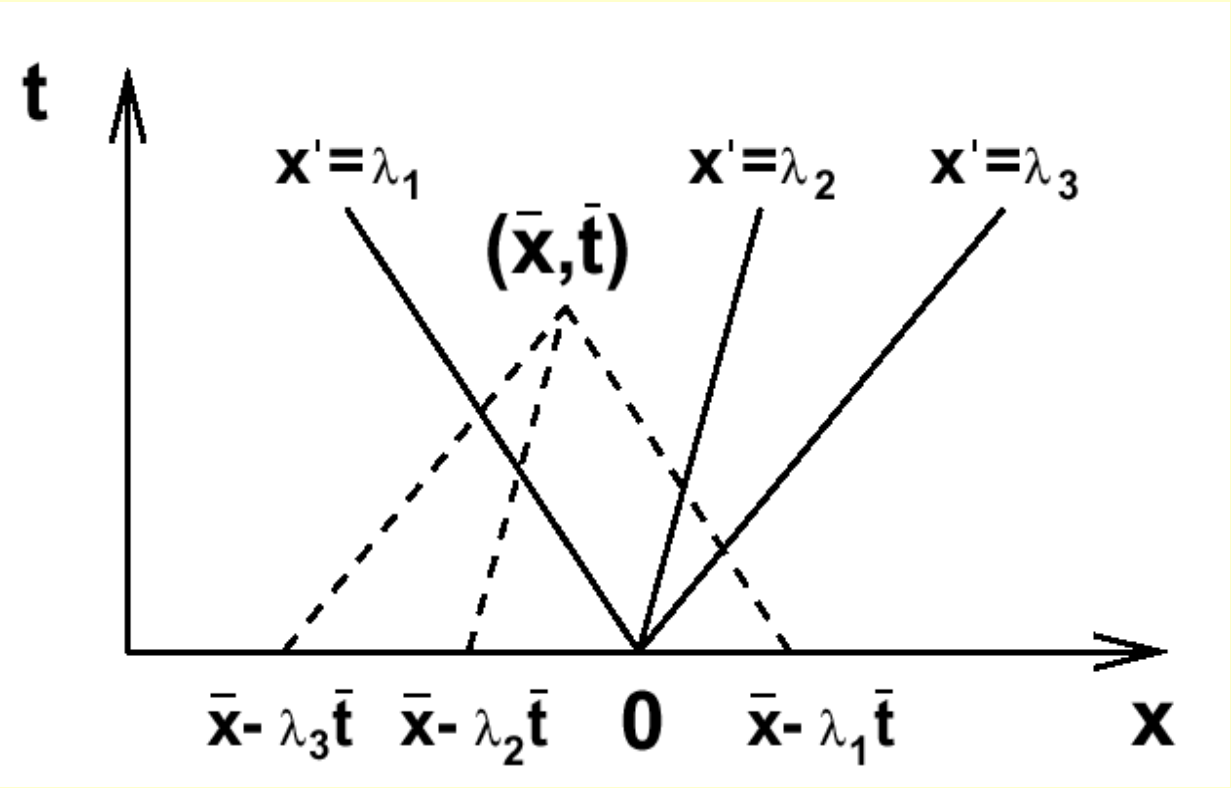
$$q(x,t) = q_0(x - at, 0)$$



system of advection equations

$$\boldsymbol{q}(x,t) = .$$

$$\sum_{p=1}^{3} v_p(x - \lambda_p t, 0) r_p$$

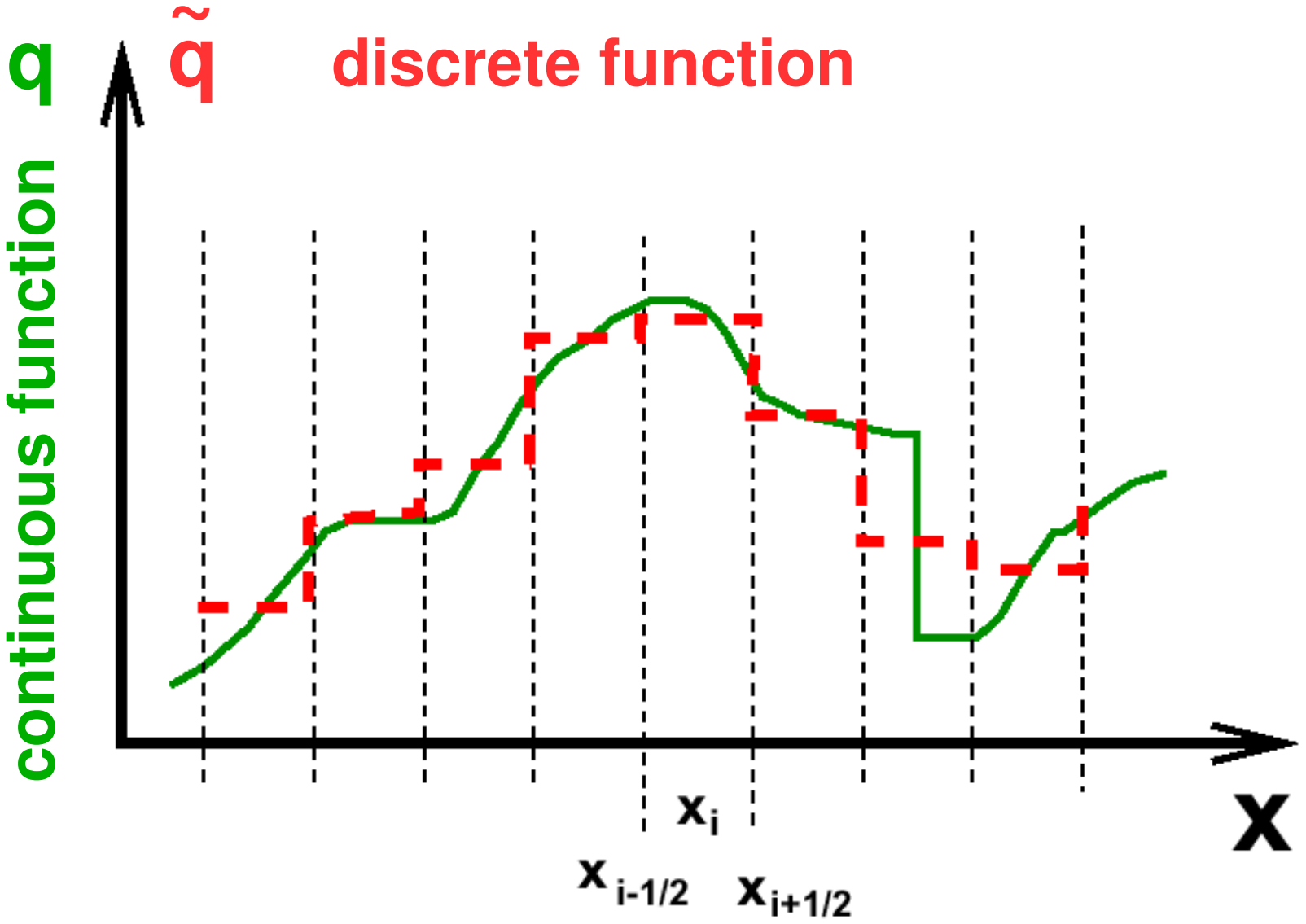


Definition:

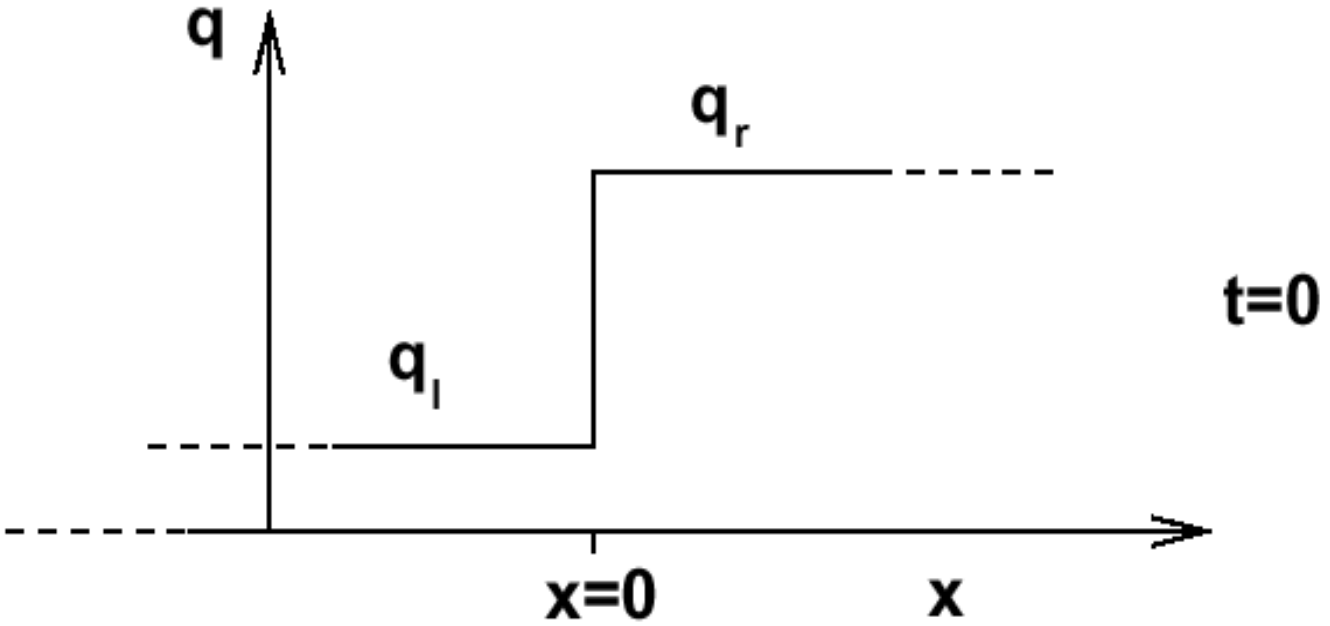
The curves $x = x_0 + \lambda_p t$ are called (p^{th}-) characteristic curves, (p^{th} characteristics).

The region between $\bar{\bar{x}} - \lambda_3 \bar{t}$ and $\bar{\bar{x}} - \lambda_1 \bar{t}$ is called the domain of dependence

How to construct an algorithm?



The Riemann- problem



$$\boldsymbol{q}(x,0)=\begin{cases}\boldsymbol{q}_l & x<0\\ \boldsymbol{q}_r & x>0\end{cases}$$

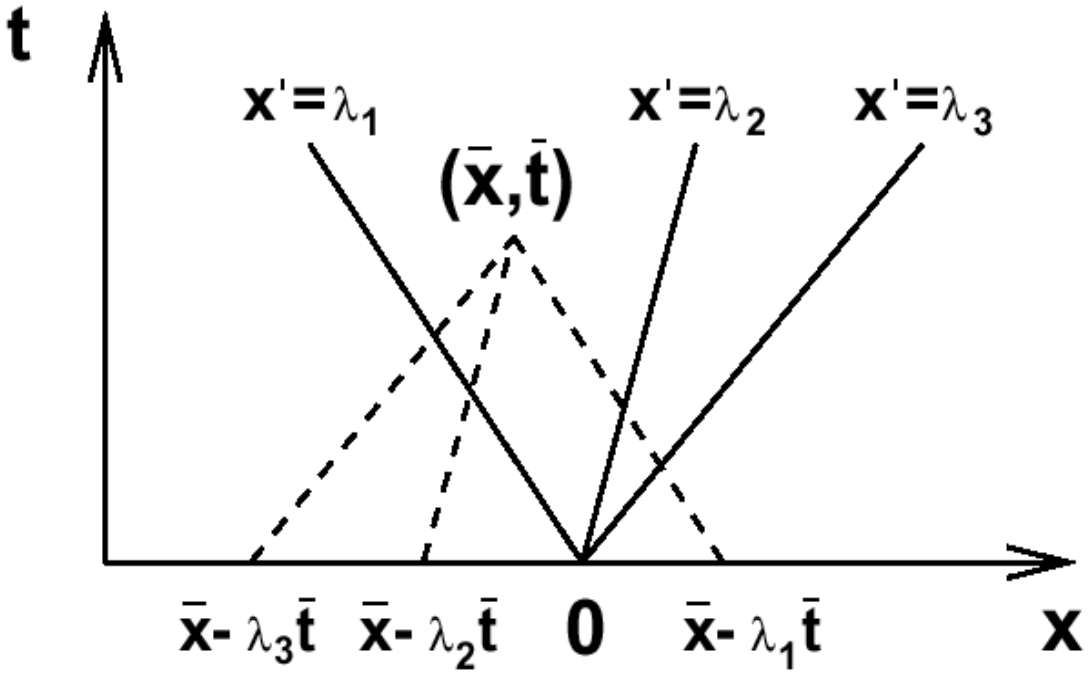
$$\boldsymbol{q}_l=\sum_{p=1}^{3} v_p^l r_p \qquad \boldsymbol{q}_l=\sum_{p=1}^{3} v_p^l r_p$$

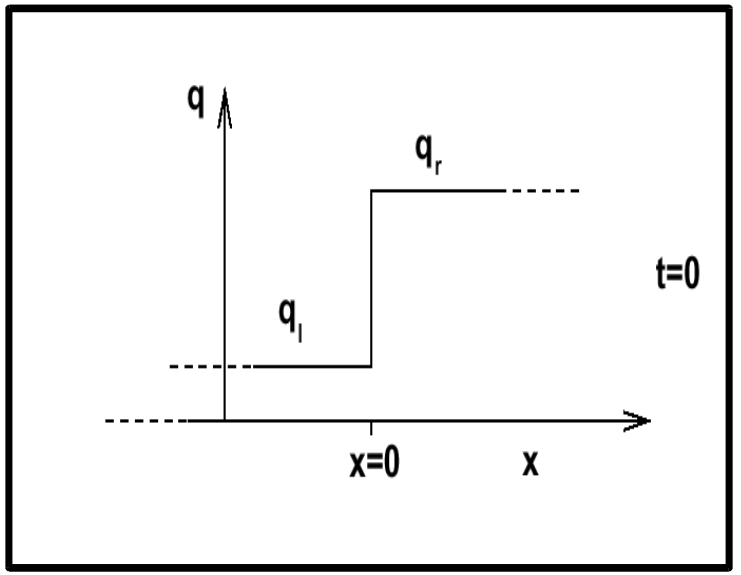
solution of the Riemann- problem

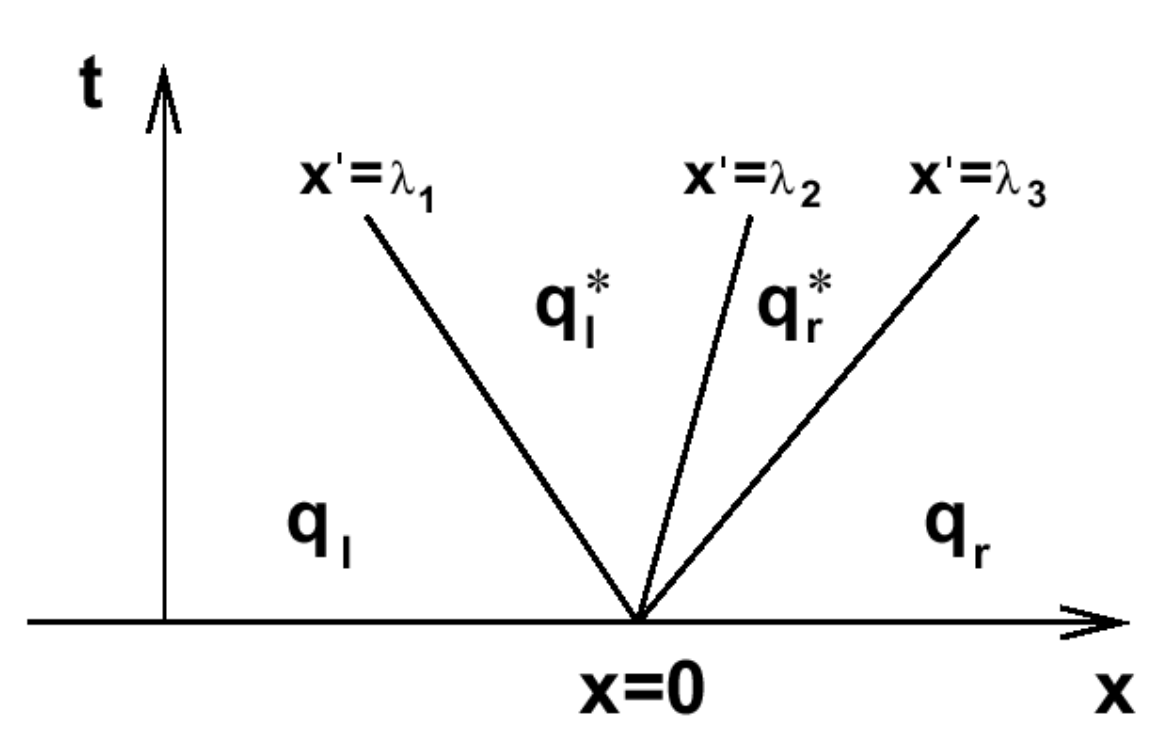
$$q(x,t)=\sum_{p=1}^{3} v_p r_p=\sum_{p=1}^{P(x,t)} v_p^r r_p+\sum_{p=P(x,t)+1}^{3} v_p^l r_p$$

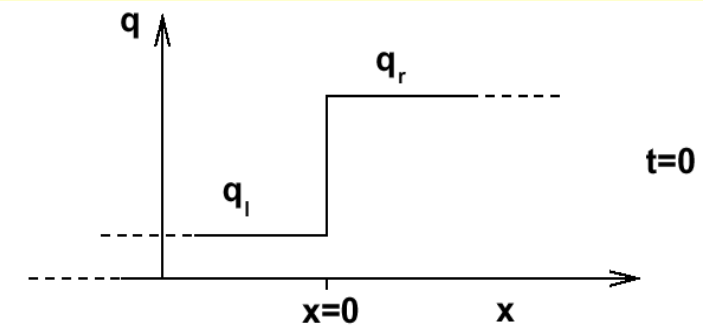
in this case: P(x,t) = 1

$$v_p(x,t)=\begin{cases} v_l & if\ (x-\lambda_{pt})<0 \\ v_r & if\ (x-\lambda_p t)>0 \end{cases}$$



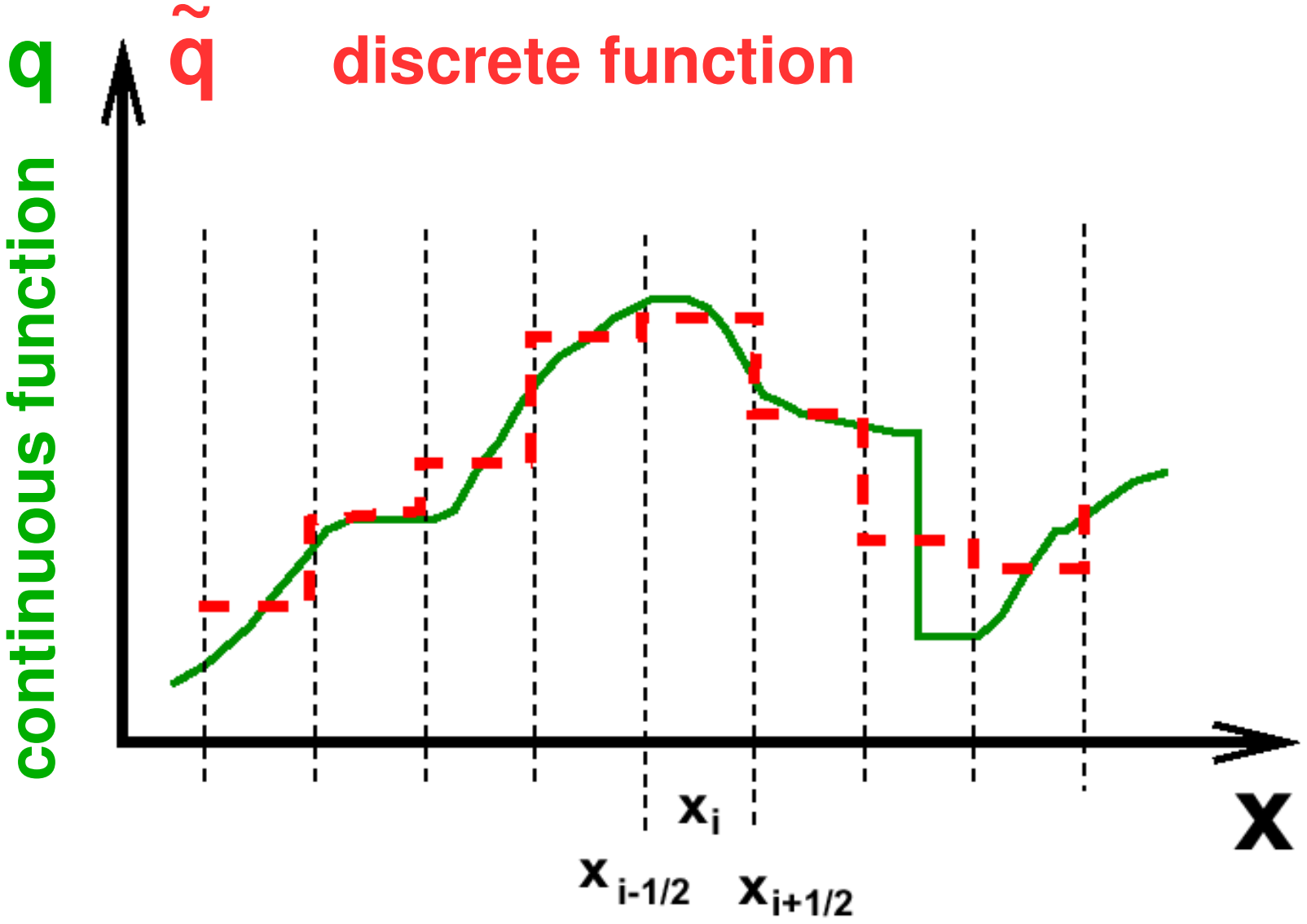




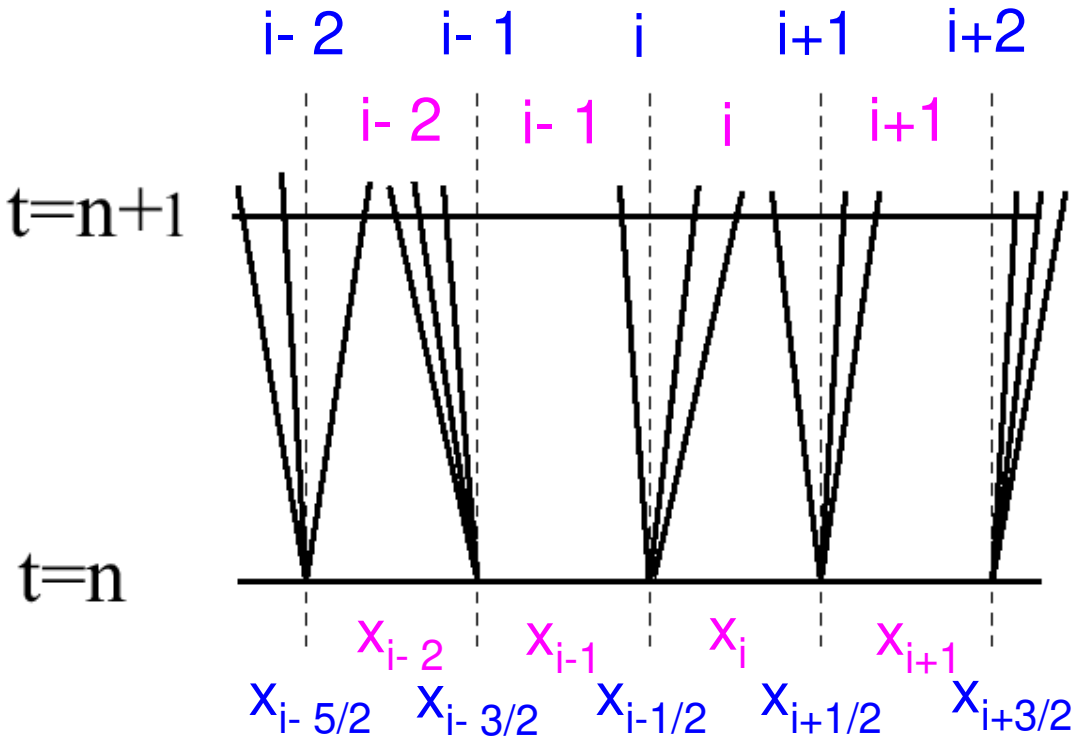


$$q_l = v_1^l r_1 + v_2^l r_2 + v_3^l r_3$$
$$q_l^* = v_1^r r_1 + v_2^l r_2 + v_3^l r_3$$
$$q_r^* = v_1^r r_1 + v_2^r r_2 + v_3^l r_3$$
$$q_r = v_1^r r_1 + v_2^r r_2 + v_3^r r_3$$

The solution consists of 4 different branches. In each branch, the solution is constant in time and is a simple sum of the eigenvector decomposition of the initial data



The discrete function on the discretization $\sum_i$ defines, at each cell interface, a series of Riemann- problems



$$q_l = \sum v_p r_p \qquad q_l^* = \sum v_p r_p$$

$$q_r^* = \sum v_p r_p \qquad q_r = \sum v_p r_p$$

fluxes

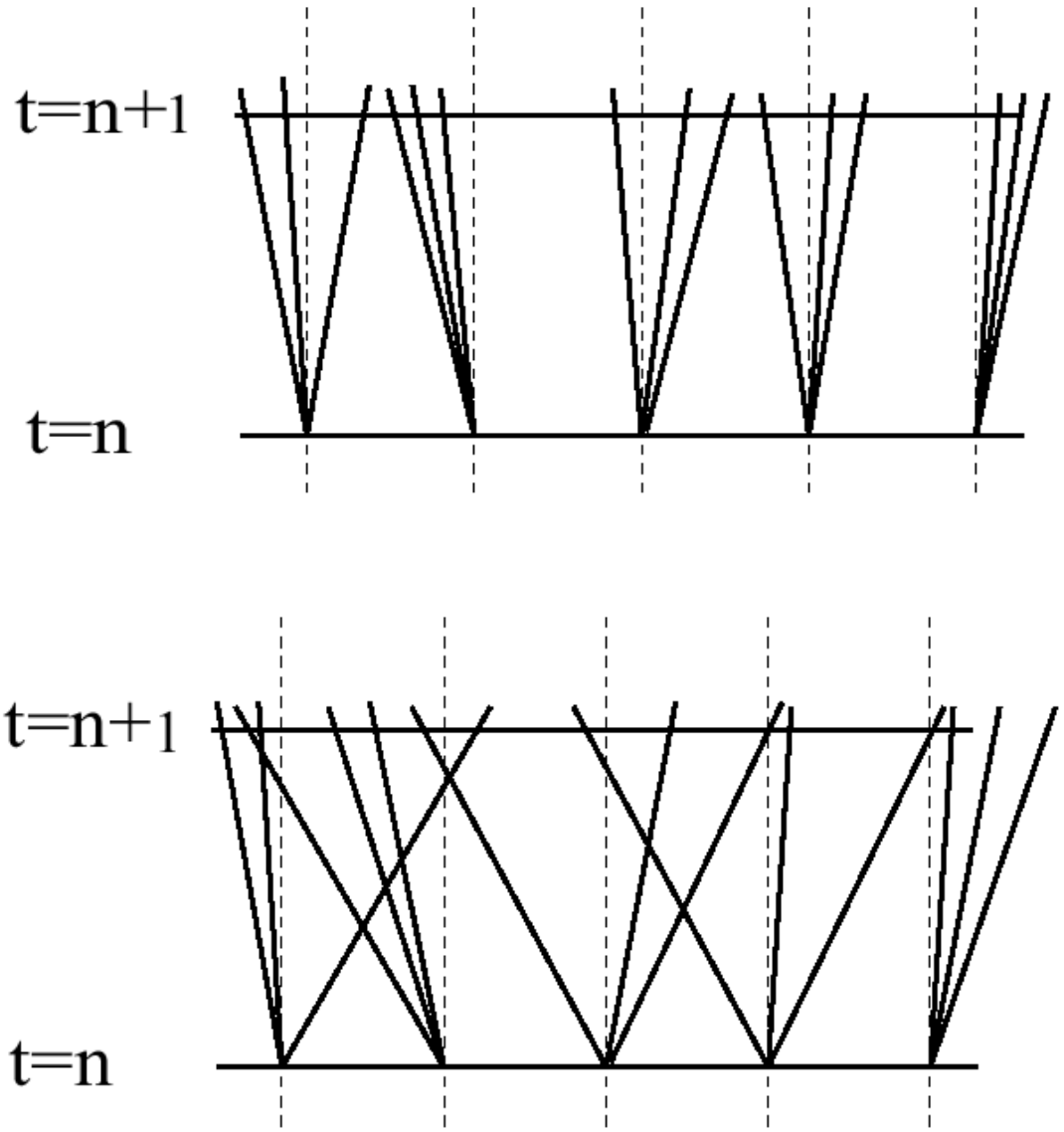
$$f_{i-2} = f(q^*_{i-2})$$
$$f_{i-1} = f(q_{i-1})$$
$$f_i = f(q^*_{i-1})$$
$$f_{i+1} = f(q^*_i)$$
$$f_{i+2} = f(q_{i+1})$$

cell updates

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\Delta x}\left[F_{i+1} - F_i\right] = \frac{\Delta t}{\Delta x}\left[f(q_i^*) - f(q_{i-1}^*)\right]$$

$$q_{i-1}^{n+1} = q_{i-1}^n - \frac{\Delta t}{\Delta x}\left[F_i - F_{i-1}\right] = \frac{\Delta t}{\Delta x}\left[f(q_{i-1}^*) - f(q_{i-1})\right]$$

The Courant- Friedrichs- Levy stability condition
(CFL - condition)



In the lower panel,
the fluxes along the cell-interfaces are no longer constant in time:
the algorithm breaks down!

$$\Delta t < \frac{\Delta x}{max_p \lambda_p}$$

CFL - condition

Note: the same result one can obtain by rigerous stability analysis

How to achieve a algorithm which is 2nd order in time and space?

(heuristic approach)

- for time: corrector- predictor

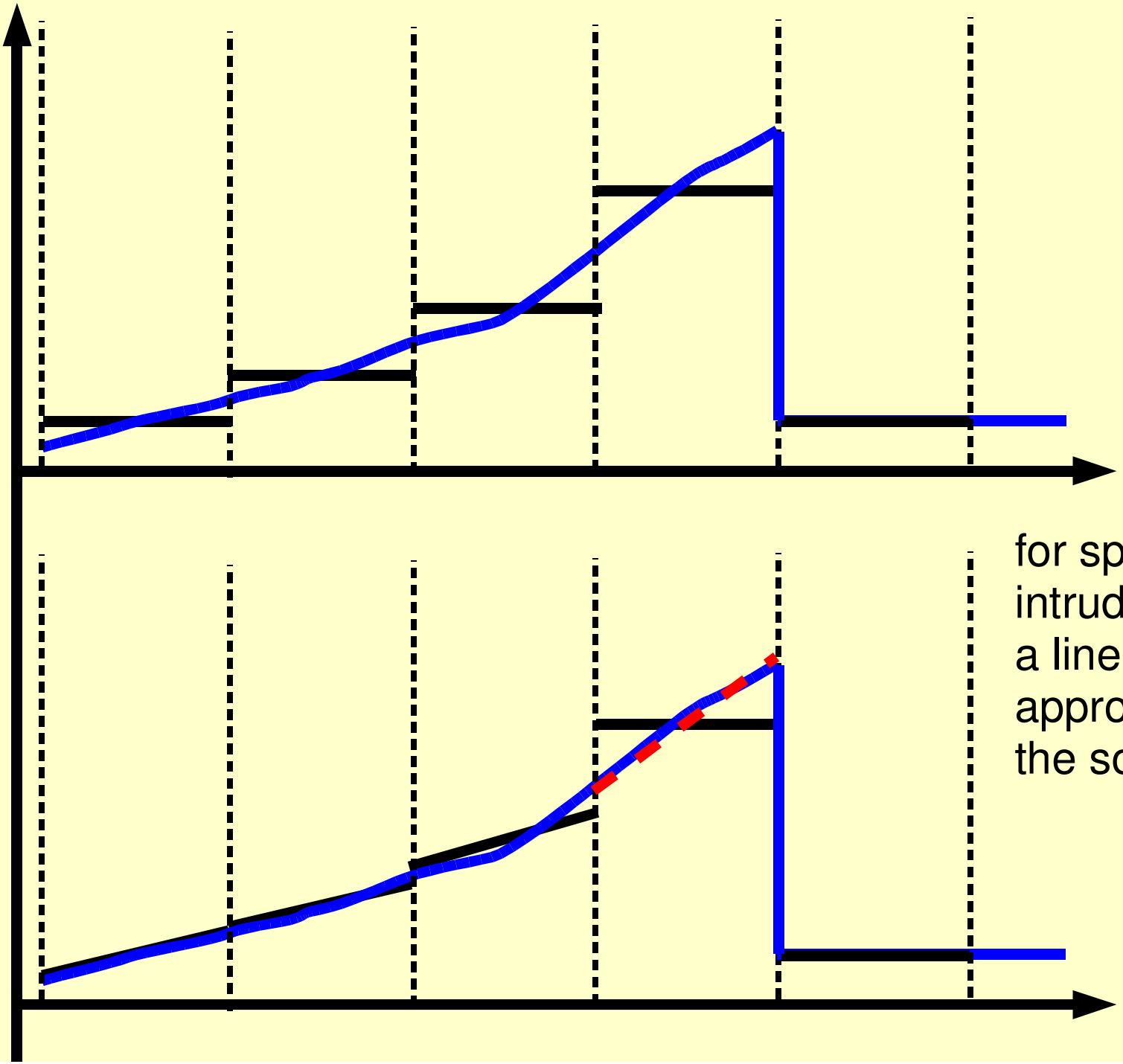
1st order

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \frac{dt}{dx}\left[\left(F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}\right)\right]$$

working with an estimation for the flux at t+1/2

2nd order

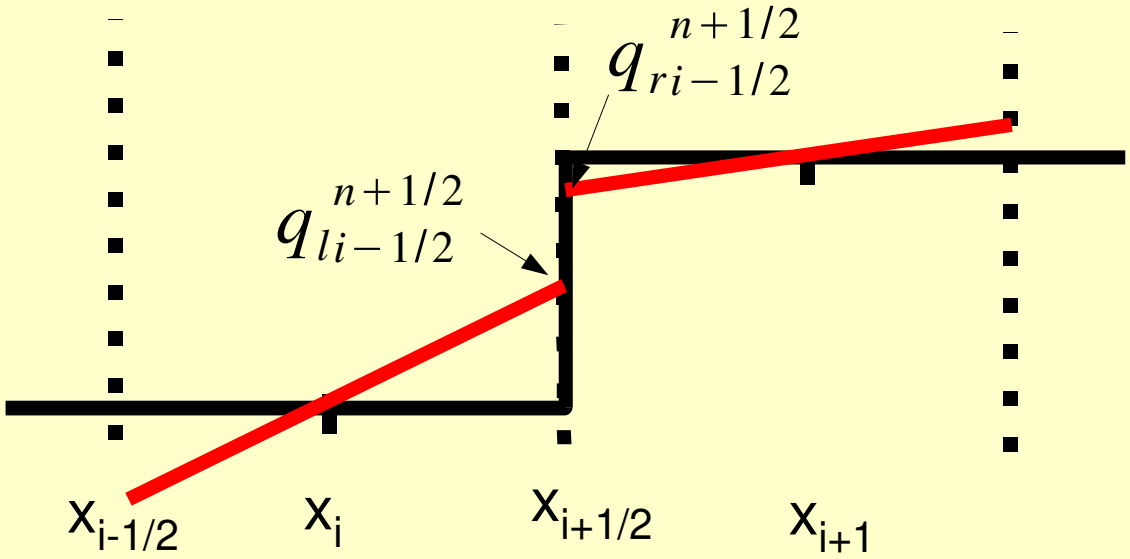
$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \frac{dt}{dx}\left[\left(F_{i+1/2,j}^{n+1/2} - F_{i-1/2,j}^{n+1/2}\right)\right]$$



for space:
intruduce slopes,
a linear
approximation to
the solution curve

How to achieve 2nd order in time and space? (formal approach)

$$q_t + \frac{\partial f}{\partial x} = 0$$



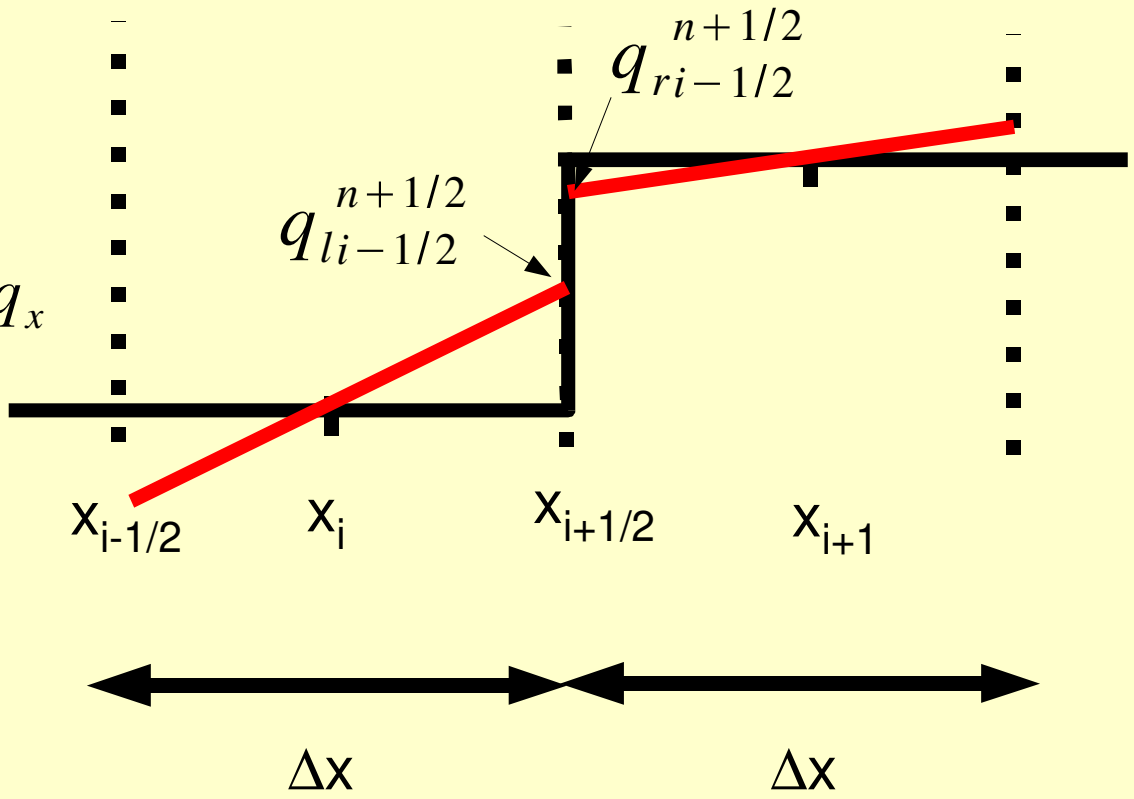
Taylor- series:

$$q^{n+1/2}_{i+1/2} \approx q^n_i + \frac{\Delta x}{2} q_x + \frac{\Delta t}{2} q_t = q^n_i + \frac{\Delta x}{2} q_x - \frac{1}{2}\frac{\partial f}{\partial u} q_x$$

$$q^{n+1/2}_{li+1/2} \approx q^n_i + \frac{1}{2}\left(\Delta x - \frac{\partial f}{\partial q}\right) q_x \qquad q^{n+1/2}_{ri-1/2} \approx q^n_{i+1} - \frac{1}{2}\left(\Delta x + \frac{\partial f}{\partial q}\right) q_x$$

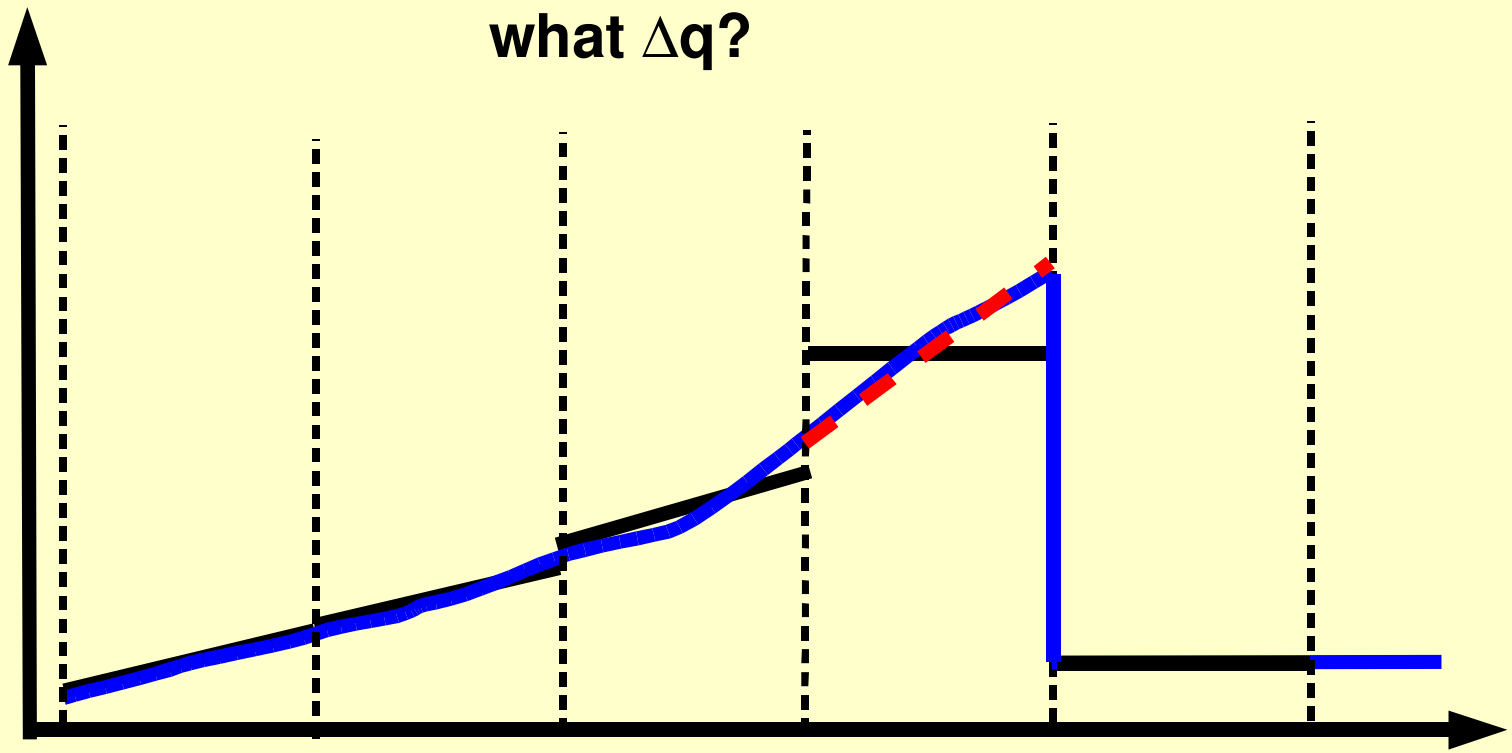
$$q_{li+1/2}^{n+1/2} \approx q_i^n + \frac{1}{2}\left(\Delta x - \frac{\partial f}{\partial q}\right) q_x$$

$$q_{ri-1/2}^{n+1/2} \approx q_{i+1}^n - \frac{1}{2}\left(\Delta x + \frac{\partial f}{\partial q}\right) q_x$$



q_x slopes ~ Δq/ Δx

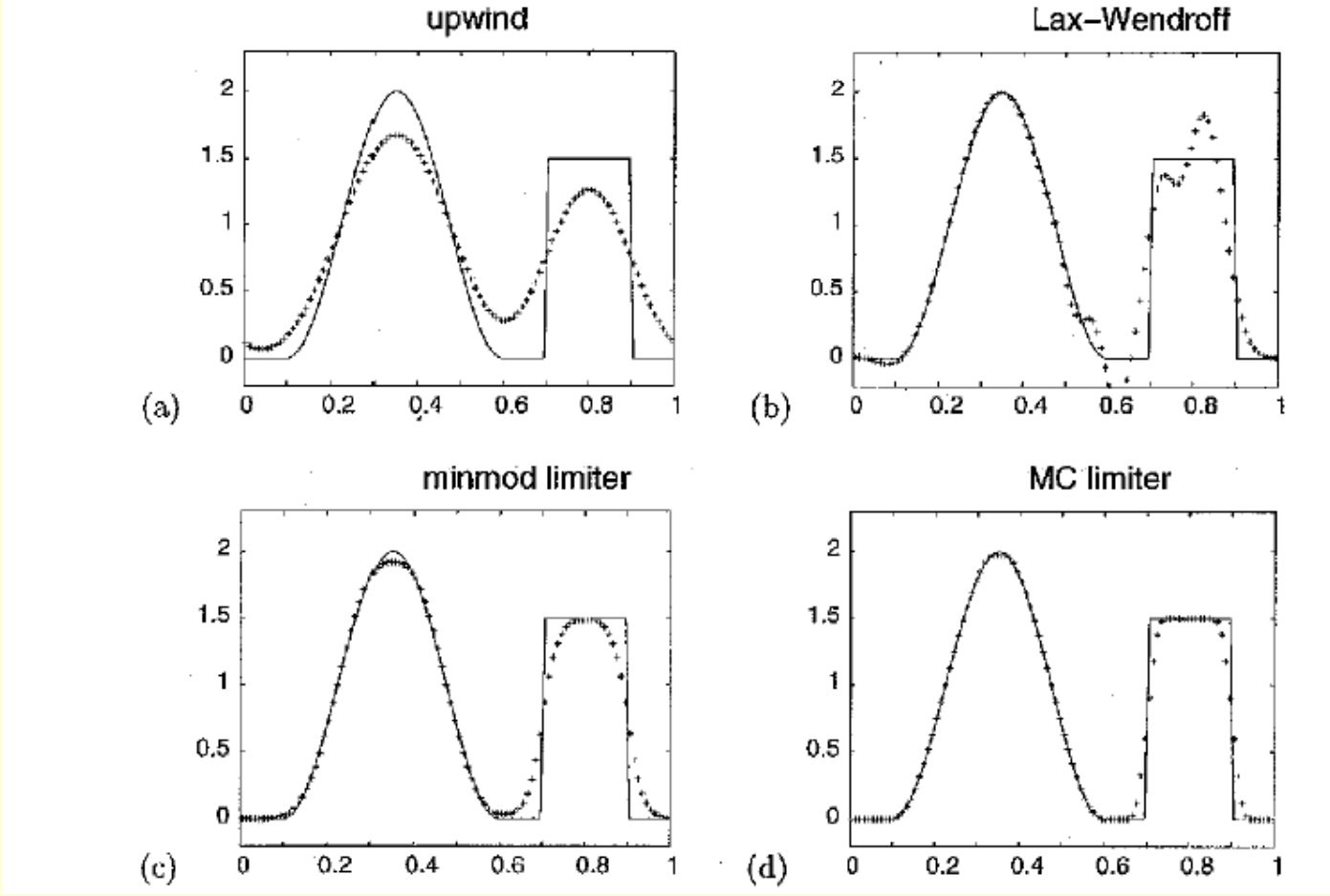
$\frac{\partial f}{\partial q}$ evaluate this again in characteristic variables



minmod limiter

$$\Delta q = \begin{cases} min((q_i - q_{i-1}),(q_{i+1} - q_i)) & if \;\; (q_i - q_{i-1}) > 0, \;\; (q_{i+1} - q_i) > 0 \\ max((q_i - q_{i-1}),(q_{i+1} - q_i)) & if \;\; (q_i - q_{i-1}) < 0, \;\; (q_{i+1} - q_i) < 0 \\ 0 & if \;\; (q_i - q_{i-1})(q_{i+1} - q_i) < 0 \end{cases}$$

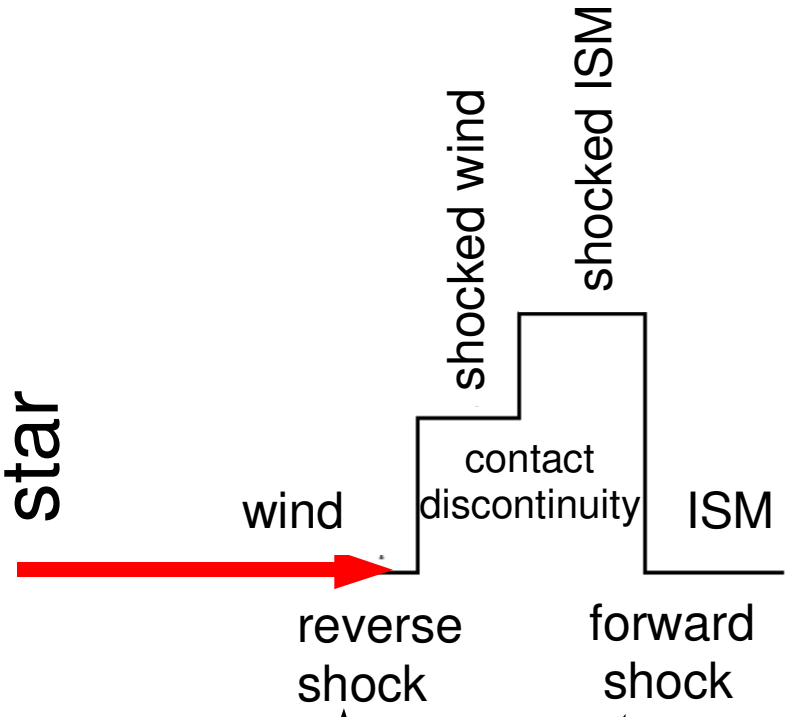
Different limiters

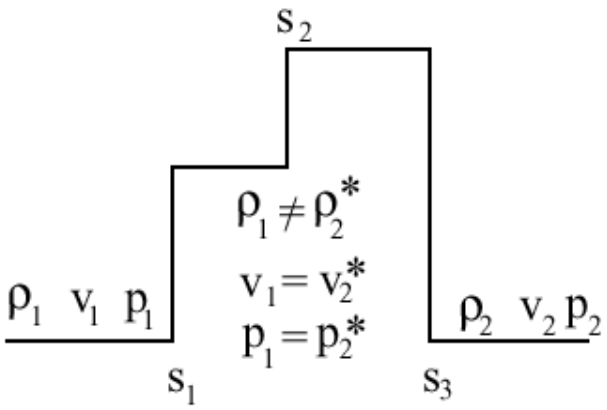


The Riemann problem of the gas dynamics

(as an example of a
non- linear conservation law)

Example 1:
Interaction of a stellar wind with the ISM





shock (1- wave)

contact (2- wave)

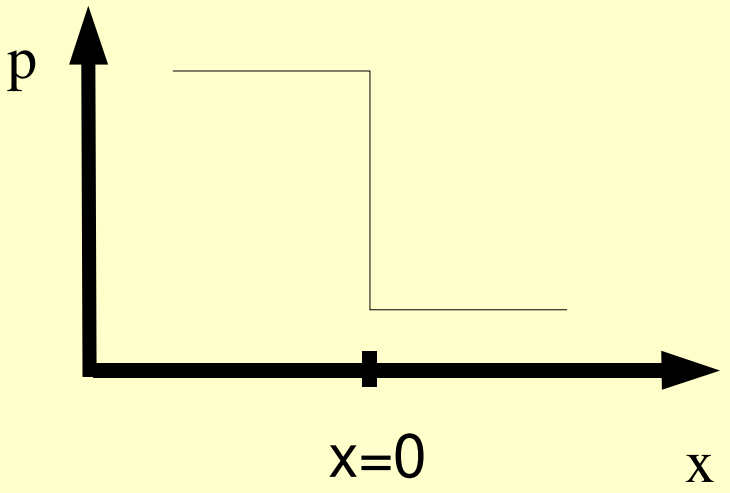
shock (3- wave)

Example 2:
The Sod problem (x=0, t=0): $\rho_l = \rho_r$

$v_l = v_r$

$p_l > p_r$

simularity to SNR !!

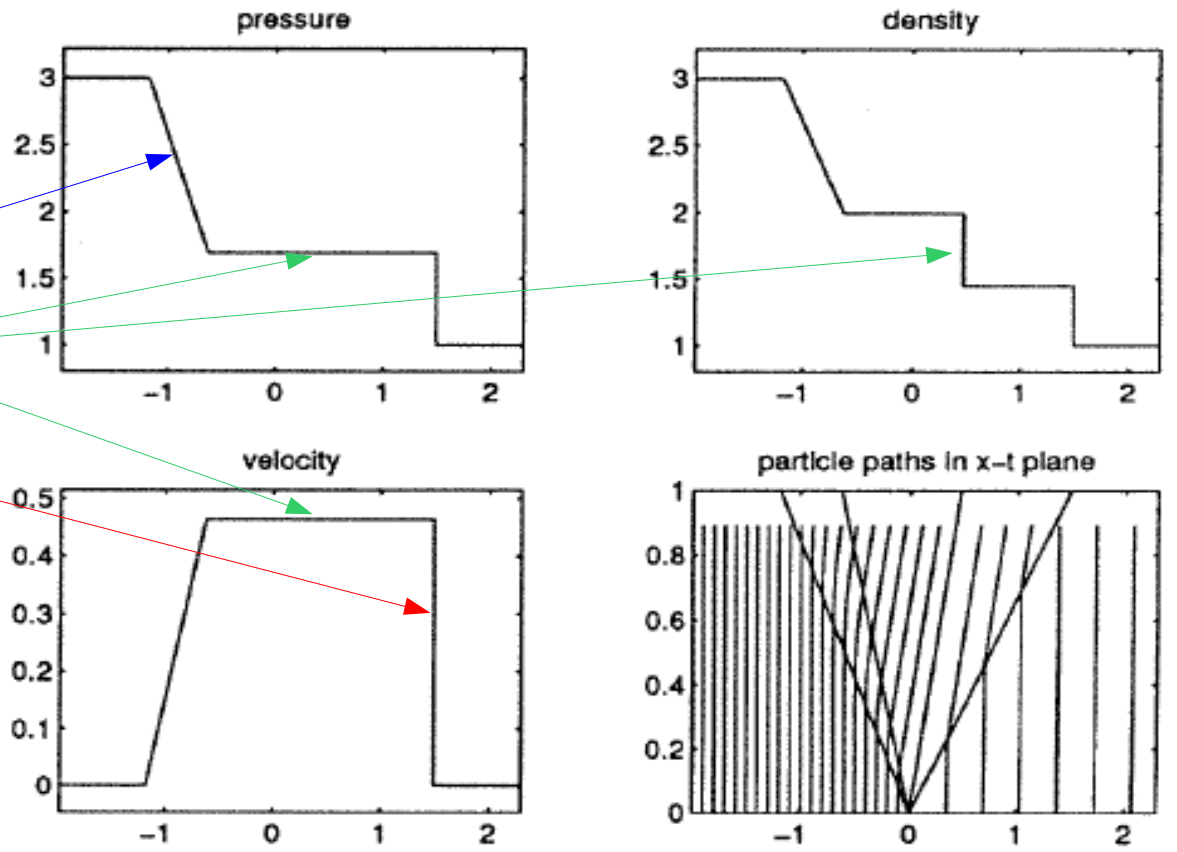


Solution for t > 0:

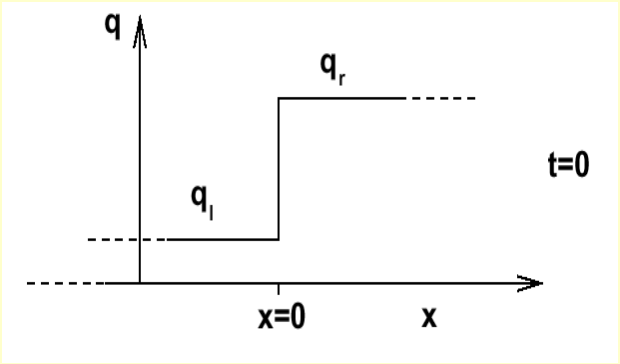
rarefaction (1- wave)

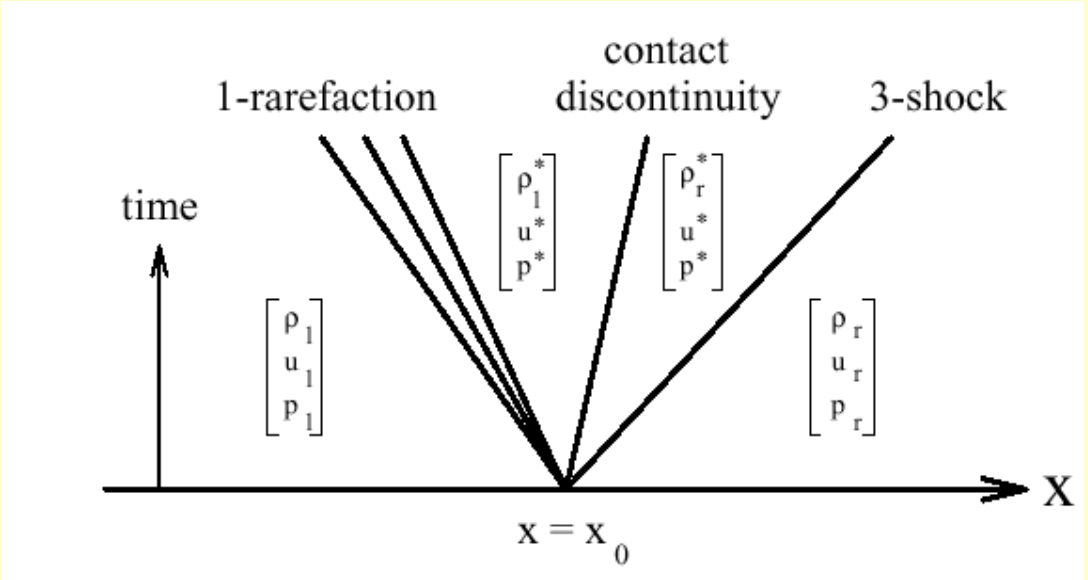
contact (2- wave)

shock (3- wave)



The solution of the Riemann-Problem of the gas- dynamics





Theorem

For a polytropic equation of state with certain restrictions to γ, the Riemann problem of the gas- dynamics has a unique, self- similar solution, consisting of 3 waves:

- the 1- wave is either a shock or a rarefaction wave
- the 2- wave is a contact- discontinuity
- the 3- wave is either a shock or a rarefaction wave

Multi- dimensional Algorithms

Dimensional Godunov Splitting

Solve 1D conservation law in x- direction with initial data q^n for one time- step Δt

$$q_t + f_x = 0$$ to obtain $\tilde{q}^{n+1/2}$

Solve 1D conservation law in y- direction with initial data $\tilde{q}^{n+1/2}$ for one time- step Δt

$$q_t + g_y = 0$$ to obtain q^n

Formally, this is a 1st order in time algorithm
even if 2nd order algorithms are used to solve the conservation laws

Dimensional Strang Splitting

Solve 1D conservation law in x- direction with initial data q^n for one half time- step $\Delta t/2$

$$q_t + f_x = 0$$ to obtain $q^{n+1/2}$

Solve 1D conservation law in y- direction with initial data $q^{n+1/2}$ for one time- step Δt

$$q_t + g_y = 0$$ to obtain $q^{n+1/2}$

Solve 1D conservation law in x- direction with initial data $q^{n+1/2}$ for one half time- step $\Delta t/2$

$$q_t + f_x = 0$$ to obtain q^{n+1}

2^{nd} order in time if the solver for the conservation law is 2^{nd} order as well

A truly multi- dimensional algorithm

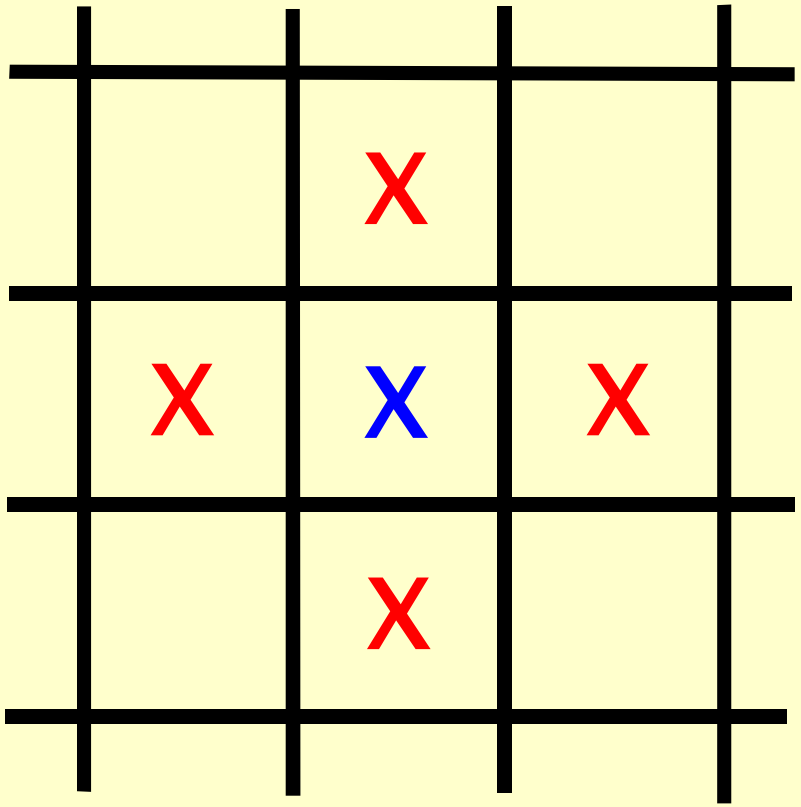
$$u_t + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0$$

$$q_{i+1/2}^{n+1/2} \approx q_i^n + \frac{\Delta x}{2} q_x + \frac{\Delta t}{2} q_t = q_i^n + \frac{\Delta x}{2} q_x - \frac{1}{2}\left[\frac{\partial f}{\partial u} q_x + \frac{\partial g}{\partial y}\right]$$

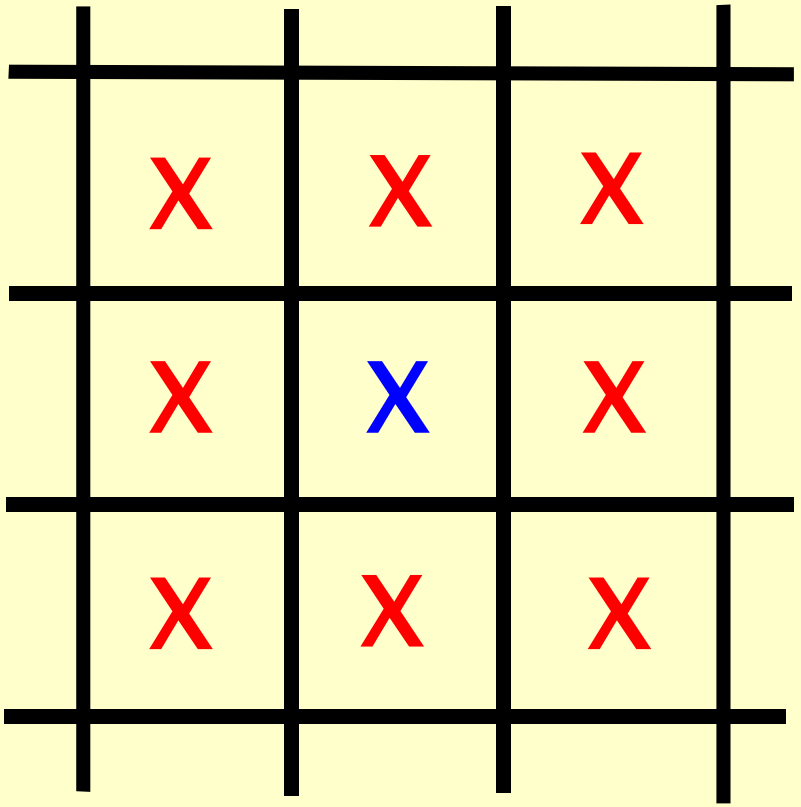
$$q_{li+1/2}^{n+1/2} \approx q_i^n + \underbrace{\frac{1}{2}\left(\Delta x - \frac{\partial f}{\partial q}\right) q_x} - \frac{\partial g}{\partial y}$$

treat this term like in 1D

solve a 1D- Riemann- problem to evaluate flux function g, and differentiate then the flux function



splitted algorithms



unsplitted , truly multidimensional algorithms

P. Colella: *Multidimensional Upwind Methods for Conservation Laws,* JCP 87,171 (1990)

M.J. Berger and R.J. LeVeque: *Adaptive Mesh Using wave-propagation for Hyperbolic Conservation Laws*, SIAM J. Numer. Anal 35,2296 (1998)

A note on div B = 0

3 different approaches are under discussion
(see G. Toth, JCP 161, 605- 652 (2000))

- staggered grids can fulfill div B exactly (but have other problems)
  (Boris with a FCT- scheme in the eighties, Zeus, and many other codes)

- projection Methods: one decomposes the magnetic field into a part
  which is divergence free and a part which has no rotation.
  The field B is the projected onto the divergence-free sub- space
  0ne has to solve a Poisson- problem, which is computationally costly
  Ryu&Francks and many others

- 8- wave solver with source terms
  K. Powell et al., JCP 154, 284- 309 (1999),
  S. Falle, S. Komissarov, and P. Joarder, MNRAS 297, 265- 277 (1998)

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\,\boldsymbol{v}\\ \boldsymbol{B}\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho\,\boldsymbol{v}\\ \rho\,\boldsymbol{v}\,\boldsymbol{v}+\left(p+\frac{1}{2\mu_0}B^2\right)\boldsymbol{I}-\frac{1}{\mu_0}\boldsymbol{B}\,\boldsymbol{B}\\ \boldsymbol{v}\,\boldsymbol{B}-\boldsymbol{B}\,\boldsymbol{v}\\ \boldsymbol{v}\left(E+p+\frac{1}{2\mu_0}B^2\right)-\frac{1}{\mu_0}\boldsymbol{B}\left(\boldsymbol{v}\cdot\boldsymbol{B}\right)\end{bmatrix}=\nabla\boldsymbol{B}\begin{bmatrix}0\\ \boldsymbol{B}\\ \boldsymbol{v}\\ \boldsymbol{v}\cdot\boldsymbol{B}\end{bmatrix}+\begin{matrix}non\\ ideal\\ terms\end{matrix}$$

8- wave algorithm with source terms

- solve the homogeneous conservation law as discussed
  ----> this generates small monopols

- solve the ODE for the source- terms consisting of these monopoles
  ----> this step reduces the monopoles

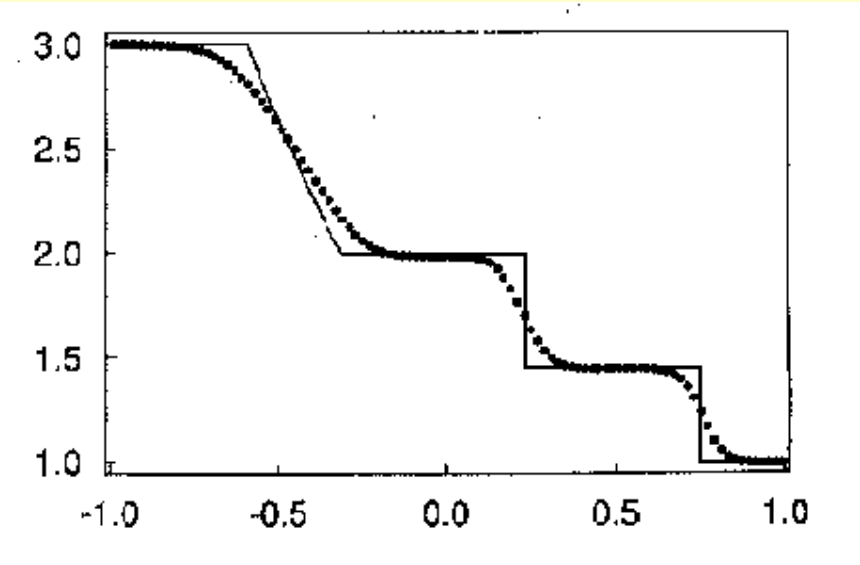
One can show that the B- field computed by this algorithm is divergence free up to truncation errors

Algorithms for

the viscous, diffusive and resistive equations

(parabolic part)

A note on numerical diffusion



consider the linear advection equation

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

solve e.g. with Lax- Friedrichs (1st order accurate method)

$$U_j^{n+1} = \frac{1}{2}\left(U_{j+1}^{n+1} + U_{j-1}^{n+1}\right) - \frac{k}{2h}A\left(U_{j+1}^{n+1} - U_{j-1}^{n+1}\right)$$

Local truncation error is (proportional to u_{xx}!)

$$L_k(x,t) = \frac{1}{2}k\left(A^2 - \frac{h^2}{k^2}\right)u_{xx}(x,t) + O\left(k^2\right)$$

$$L_k(x,t) = \frac{1}{2} k \left(A^2 - \frac{h^2}{k^2} \right) u_{xx}(x,t) + O\left(k^2\right)$$

Therefore, the method solves more accurately the following advection- diffusion equation:

$$\frac{\partial u}{\partial t} + A \frac{\partial u}{\partial x} = \frac{1}{2} \frac{h^2}{k} \left(I - \frac{k^2}{h^2} A^2 \right) u_{xx}(x,t)$$

Hence, the numerical diffusivity is of the order:

$$D = k \left(A^2 - \frac{h^2}{k^2} \right) u_{xx}(x,t) + O\left(k^2\right)$$

$$D = \frac{1}{2}\frac{h^2}{k}\left(I - \frac{k^2}{h^2}A\right)$$

- ➔ as k-> 0, the diffusive term vanishes
- ➔ diffusion depends on the transport operator A
- ➔ diffusion depends on the discretization length h

  [D ---> 0 (h --- > 0)]

- ➔the analysis can be generalized to higher order methods: beside a dissipative, a dispersive terms will be present
- ➔ in 2nd order methods, the dispersive terms dominates at discontinuities ----> strong oscialliations appear

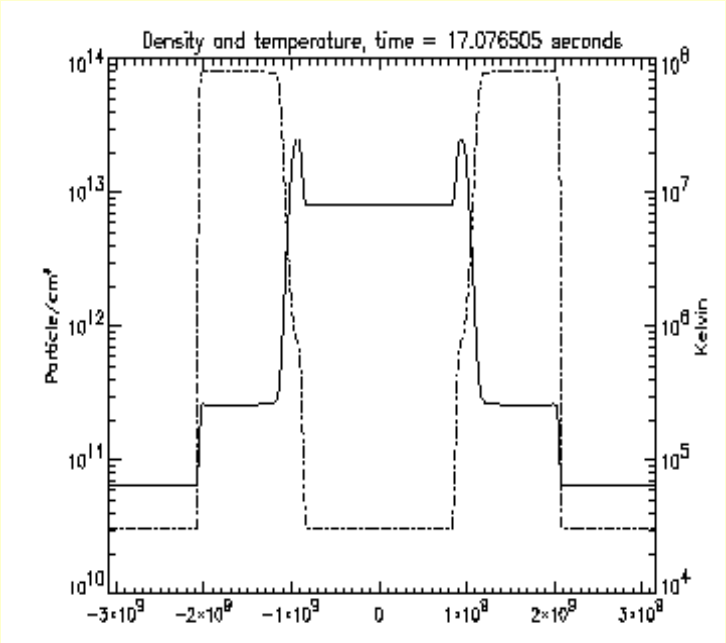
We discuss diffusive processes along the heat-transport equation

$$\frac{\partial T}{\partial t} - \frac{\partial}{\partial x}\left(\kappa_{NL} T^{\beta} \frac{\partial T}{\partial x}\right) = 0$$ non- linear case

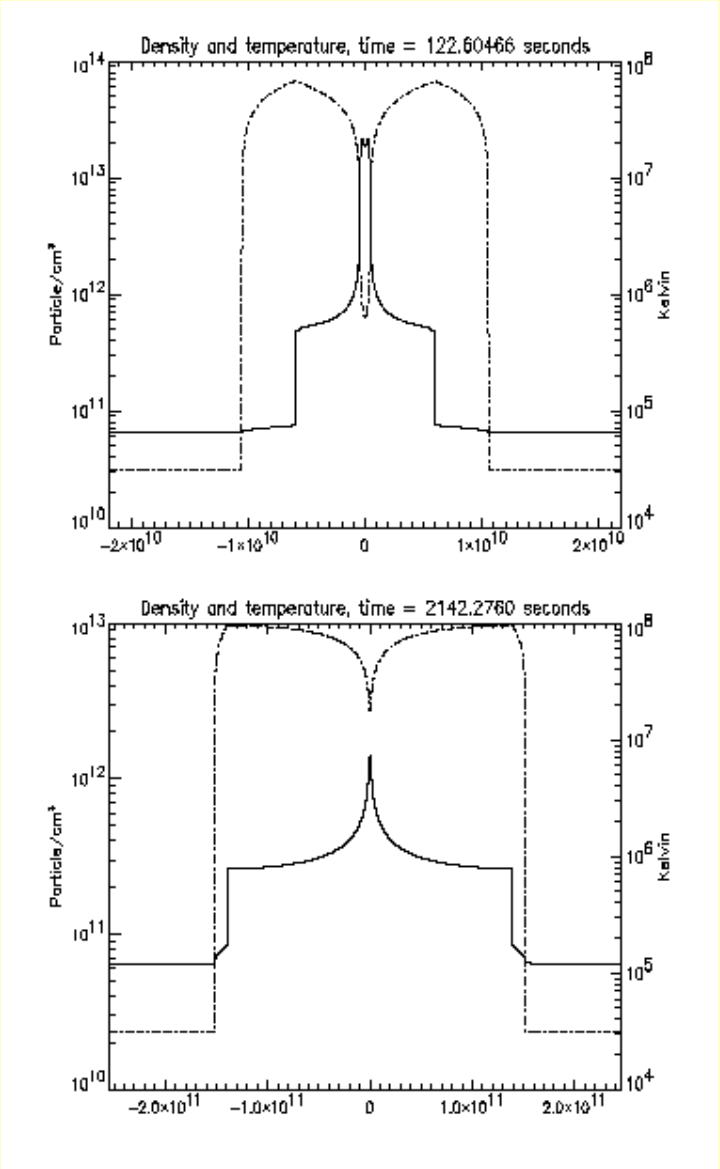
$$\frac{\partial T}{\partial t} - \kappa_L \Delta T = 0$$ linear case ($\beta = 0$)

Note 1: this is a stiff equation with a very complex Eigenvector structure
-----> Solve with a implicit method

Note 2: $\Delta t_{explicit} \approx \frac{1}{2} \frac{(\Delta x)^2}{\kappa T^{\beta}}$ wheras in the advection equations:
$\Delta dt \sim \Delta x$



without heat conduction



with heat conduction

Semi- discrete methods (method of lines):

1. Discretize first in space ---->

   obtain a system of ordinary differential equations (ODE)

2. Solve the system with any (well- known) method to solve ODE

Advantages: can use the huge knowledge on ODE- solvers packages
---> see talk of Doris

$$\frac{\partial T}{\partial t} - \frac{\partial}{\partial x}\left(\kappa_{NL} T^{\beta} \frac{\partial T}{\partial x}\right) = 0$$

$$\frac{\partial T}{\partial t} - \frac{\kappa_{NL}}{\beta + 1} \Delta_x T^{\beta + 1} = 0$$

$$\frac{\partial T_i}{\partial t} = A_{\Delta x} T_i$$

$$A_{\Delta x} T_i^n \equiv \frac{\kappa_{NL}}{\beta + 1} \frac{\left(T_{i+1}^n\right)^{\beta+1} - 2\left(T_i^n\right)^{\beta+1} + \left(T_{i-1}^n\right)^{\beta+1}}{2\Delta x}$$

Crank- Nicolson (2nd order in space and time, A- stable)

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{2}\left(A_{\Delta x} T_i^n + A_{\Delta x} T_i^{n+1} \right)$$

Trapez- BDF2 (Backward- Differencing Formula
2nd order in space and time, L- stable)

$$T_i^{n+1/2} = T_i^n + \frac{\Delta t/2}{2}\left(A_{\Delta x} T_i^n + A_{\Delta x} T_i^{n+1/2} \right)$$

$$T_i^{n+1} = \frac{1}{3}\left[-T_i^n + 4\,T_i^{n+1/2} + \Delta t\, A_{\Delta x} U_i^{n+1} \right]$$

How to solve for T^{n+1}?

- How to treat the non- linearity
- Always to solve large linear systems of equations!
  Extended theory how to do this fast and accurate ---> Doris' talk

Here: how to treat non- linearity?
time- steps and stability of different solvers

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{2}\left(A_{\Delta x} T_i^n + A_{\Delta x} T_i^{n+1} \right)$$

$$\left(I - \frac{\Delta t}{2} A_{\Delta x} \right) T_i^{n+1} - \left(I + \frac{\Delta t}{2} A_{\Delta x} \right) T_i^n = 0$$

$$\left(I-\frac{\Delta t}{2}A_{\Delta x}\right)T_i^{n+1}-\underbrace{\left(I+\frac{\Delta t}{2}A_{\Delta x}\right)}_{K}T_i^n=0$$

$$\overbrace{T_i^{n+1}\underbrace{\frac{-\Delta t}{2}\frac{\kappa_{NL}}{\beta+1}\left[\left(T_{i-1}^{n+1}\right)^{\beta+1}-2\left(T_i^{n+1}\right)^{\beta+1}+\left(T_{i+1}i^{n+1}\right)^{\beta+1}\right]}_{G}-K}^{F}=0$$

$$\boldsymbol{T}=\left(T_{1,}\ldots,T_i,\ldots,T_n\right)^T$$

$$\boldsymbol{F}\left(\boldsymbol{T}^{\boldsymbol{n+1}}\right)=0 \qquad \boldsymbol{G}\left(\boldsymbol{T}^{\boldsymbol{n+1}}\right)=\boldsymbol{T}^{\boldsymbol{n+1}}$$

We now set $T \equiv T^{n+1}$, and denote m as iteration index, and set T^0 equal a starting value

Newton- iteration

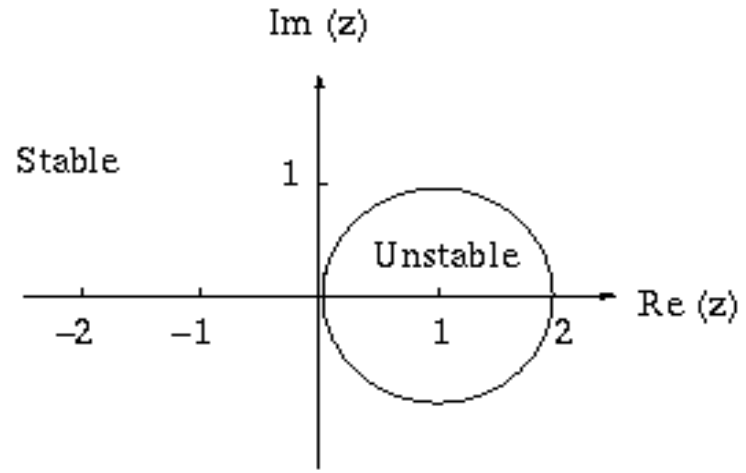
$$D\,F\left(\boldsymbol{T}^{\boldsymbol{m}}\right)\Delta\boldsymbol{T}^{\boldsymbol{m}} = -\boldsymbol{F}\left(\boldsymbol{T}^{\boldsymbol{m}}\right)$$
$$\boldsymbol{T}^{\boldsymbol{m}+1} = \boldsymbol{T}^{\boldsymbol{m}} + \Delta\boldsymbol{T}^{\boldsymbol{m}}$$

if DF is not known analytically:
rank 1 - update (e.g. Breudon)

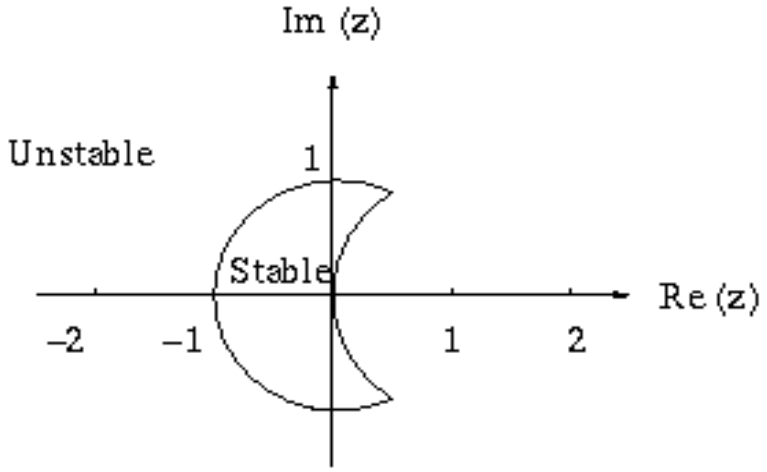
$$\boldsymbol{J}^{m}\left(\boldsymbol{T}^{\boldsymbol{m}}\right)\Delta\boldsymbol{T}^{\boldsymbol{m}} = -\boldsymbol{F}\left(\boldsymbol{T}^{\boldsymbol{m}}\right)$$
$$\boldsymbol{T}^{\boldsymbol{m}+1} = \boldsymbol{T}^{\boldsymbol{m}} + \Delta\boldsymbol{T}^{\boldsymbol{m}}$$
$$\boldsymbol{J}^{m+1} = \boldsymbol{J}^{m} + \frac{\boldsymbol{F}\left(\boldsymbol{T}^{\boldsymbol{m}}\right)\left(\Delta\boldsymbol{T}^{\boldsymbol{m}}\right)^{T}}{\left(\Delta\boldsymbol{T}^{\boldsymbol{m}}\right)^{T}\Delta\boldsymbol{T}^{\boldsymbol{m}}}$$

Fix- point- iteration

$$\boldsymbol{G}\left(\boldsymbol{T}^{\boldsymbol{m}}\right) = \boldsymbol{T}^{\boldsymbol{m}+1}$$



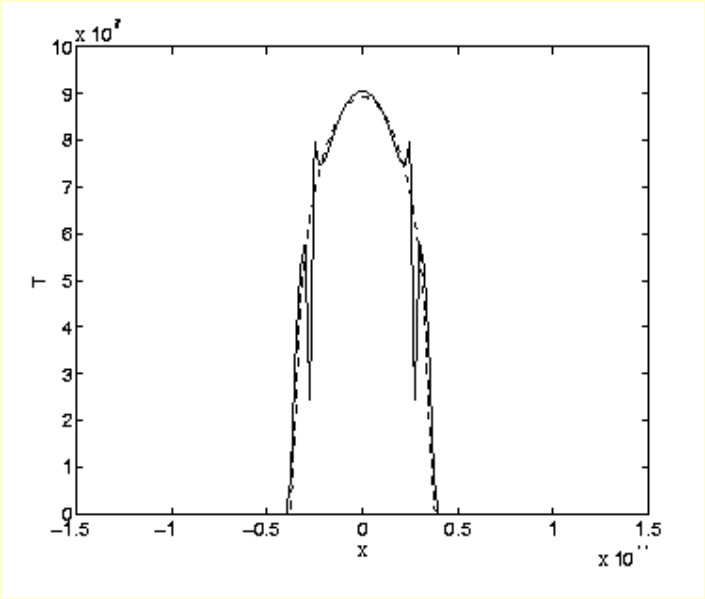
stability region of the implicit Euler method for a scalar test problem



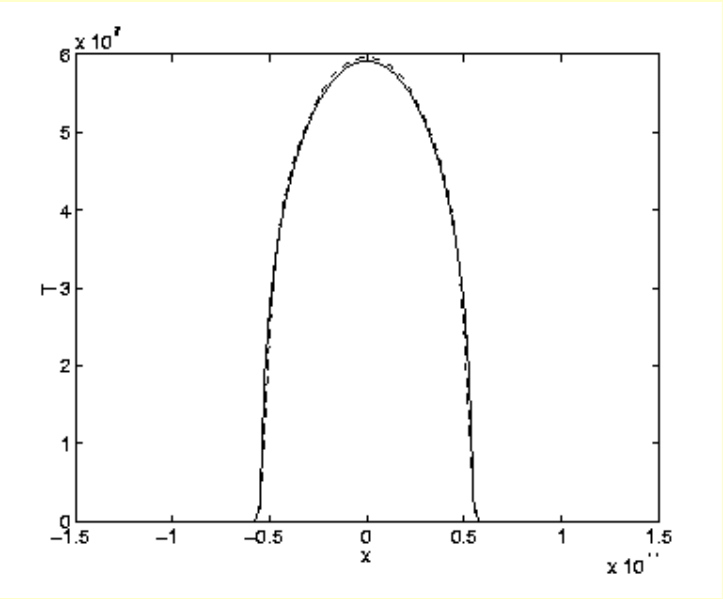
stability region of the implicit Euler method when using fix- point-iteration

Simin Motamen, ETH Thesis, 2002

The difference between a A- and L- stable algorithm



The same time- step for the A- stable (solid line) and the L- stable algorithm (dashed line)



100 times bigger time- step for the L- stable (dashed line) than for the A- stable algorithm (solid line)

Simin Motamen, ETH Thesis, 2002

analysis of some algorithms

Scheme	Solver	CFL_{expl}	$\triangle t$	N_{It}
TR-BDF2*	Newton	100	0.02	5
CN	Newton	10	0.002	50
TR-BDF2*	Fixed Point	1	0.0002	500
DW	Fixed Point	1	0.0002	500
CN	Fixed Point	1	0.0002	500

maximal stable time- step

Scheme	Solver	CFL_{expl}	$\|\|E\|\|_1$	$\|\|E\|\|_2$	$\|\|E\|\|_\infty$
TR-BDF2*	Newton	100	0.0590	0.0594	0.0416
CN	Newton	10	0.0590	0.0648	0.0554
TR-BDF2*	Fixed Point	1	0.0590	0.0643	0.0544
DW	Fixed Point	1	0.0590	0.0643	0.0544
CN	Fixed Point	1	0.0590	0.0643	0.1321

error analysis

Numerical Scheme	Solver	CFL_{expl}	CPU(s)
TR-BDF2*	Newton	100	3.3
CN	Newton	10	12.2
TR-BDF2*	Fixed Point	1	126
DW	Fixed Point	1	495
CN	Fixed Point	1	140

CPU- time comparison

Simin Motamen, ETH Thesis, 2002