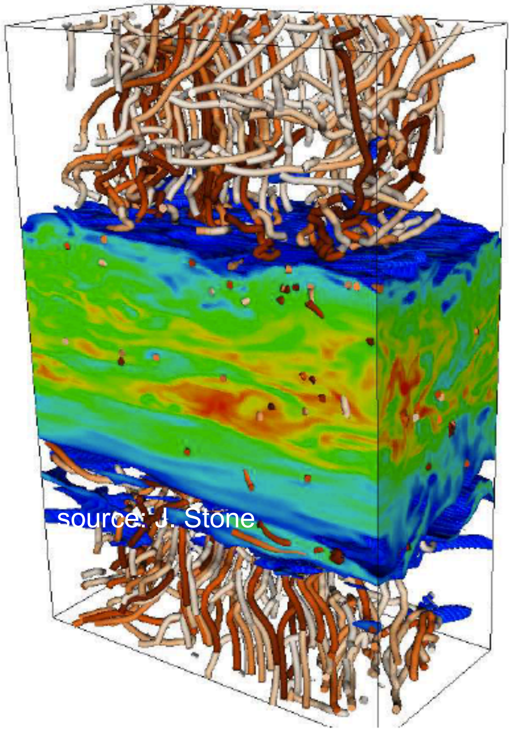
Winter School on Computational Astrophysics, Shanghai, 2018/01/31

Computational Magnetohydrodynamics


source: J. Stone

Xuening Bai (白雪宁)

Institute for Advanced Study (IASTU) &

Tsinghua Center for Astrophysics (THCA)

清華大學

Tsinghua University

Why computational MHD?

- Astrophysical plasma phenomena are typically highly-nonlinear, and in multi-D: need reliable numerical algorithms to solve MHD equations.

  Computational science has constantly been growing with progress in computer hardware, numerical analysis, software engineering.

  High-performance computing has become standard practice in science.

- MHD provides accurate description of collisional plasmas.
- MHD is often reasonably OK to describe the large-scale phenomenon (even) for collisionless plasmas.
- Computational MHD has been playing a major role in nearly all subfields of astrophysics from star/planet formation to cosmology.

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Useful reference: *Finite Volume Methods for Hyperbolic Problems*, LeVeque, 2002, Cambridge University Press.

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Types of PDEs

Hydro and MHD equations are a system of partial differential equations (PDEs).

There are in general 3 types of PDEs. For a 2nd order PDE of the form

$$a\partial^2_{xx}u + b\partial^2_{xy}u + c\partial^2_{yy}u + d\partial_x u + e\partial_y u + fu = g$$

it can be categorized based on the discriminant:

$$b^2 - 4ac \begin{cases} < 0 & \rightarrow \quad \text{elliptic,} \\ = 0 & \rightarrow \quad \text{parabolic,} \\ > 0 & \rightarrow \quad \text{hyperbolic,} \end{cases}$$

Hydro/ideal MHD equations are hyperbolic PDEs, but source terms (resistivity/viscosity/self-gravity) can be of other types.

Types of PDEs

Prototype of elliptic PDE:

Poisson equation: $\nabla^2 u = f$ (self-gravity)

Prototype of parabolic PDE:

Diffusion equation: $\partial_t u = D\partial^2_{xx} u$ (viscosity, resistivity, heat conduction)

Prototype of hyperbolic PDE:

Wave equation: $\partial^2_{tt} u - c^2 \partial^2_{xx} u = 0$

Linear advection equation: $\partial_t u + A\partial_x u = 0$

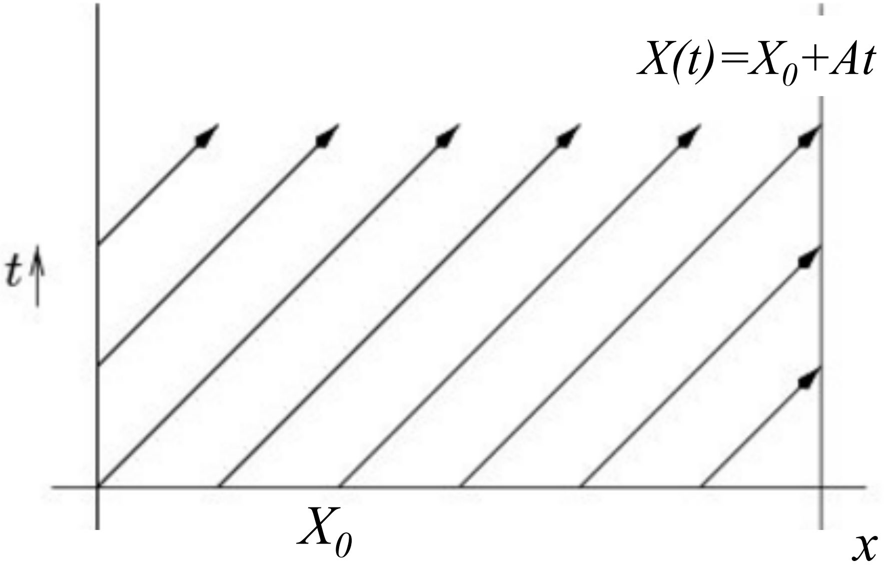
The linear advection equation (const coeff)

Consider linear advection eqs with constant A: $\partial_t u + A\partial_x u = 0$

Solution: $u(x,t) = u_0(x - At)$

The solution is constant along the ray (called the characteristic curve):

$$X(t) = X_0 + At$$



Proof:

$$\frac{d}{dt}u(X(t),t) = \partial_t u(X(t),t) + X'(t)\partial_x u(X(t),t)$$

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

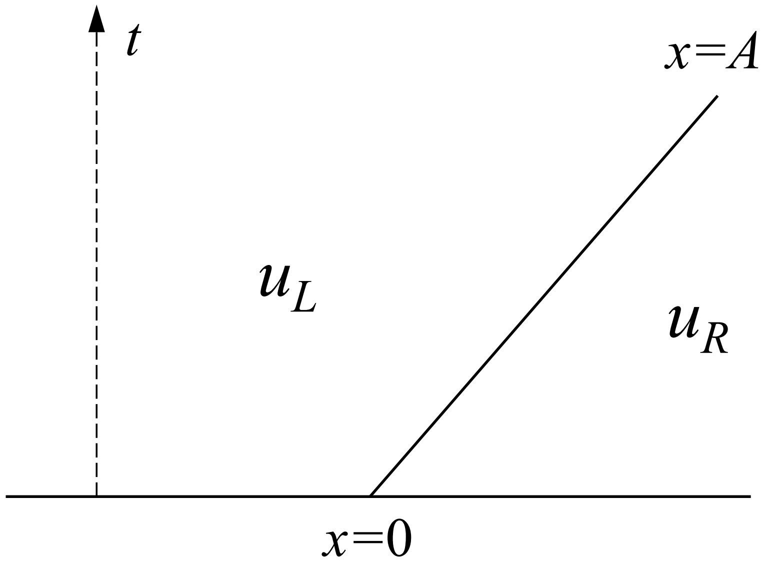
The Riemann problem (for linear advection eq)

$$\partial_t u + A\partial_x u = 0$$

Initial condition: $u=u_L$, $(x<0)$

$u=u_R$, $(x\geq 0)$

Result: discontinuity propagates along the characteristic curve.



Hyperbolicity of linear systems

A linear system of the form

$$\partial_t \boldsymbol{u} + \mathbf{A} \cdot \partial_x \boldsymbol{u} = 0$$

is hyperbolic if matrix **A** is diagonalizable with real eigenvalues.

Let us denote the eigenvalues by $\lambda^1 \leq \lambda^2 \leq ... \leq \lambda^m$

The matrix is diagonalizable if there is a complete set of eigenvectors such that

$$A\boldsymbol{r}^p = \lambda^p \boldsymbol{r}^p$$

The right-eigenvectors jointly form a matrix: $R \equiv (\boldsymbol{r}^1, \boldsymbol{r}^2, ..., \boldsymbol{r}^m)$

so that $AR = \Lambda R$ where $\Lambda \equiv \mathrm{diag}(\lambda^1, \lambda^2, ..., \lambda^m)$.

In this way, the matrix A is diagonalized as: $\boxed{R^{-1}AR = \Lambda}$

Hyperbolicity of linear systems

A linear system of the form

$$\partial_t \boldsymbol{u} + \mathbf{A} \cdot \partial_x \boldsymbol{u} = 0$$

is hyperbolic if matrix **A** is diagonalizable with real eigenvalues.

For any vector $\boldsymbol{u}$, we can rewrite the original equation into:

$$\partial_t (R^{-1}\boldsymbol{u}) + (R^{-1}AR)\partial_x (R^{-1}\boldsymbol{u}) = 0$$

By defining characteristic variables as $\boxed{\boldsymbol{w} = R^{-1}\boldsymbol{u}}$, the linear system becomes

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

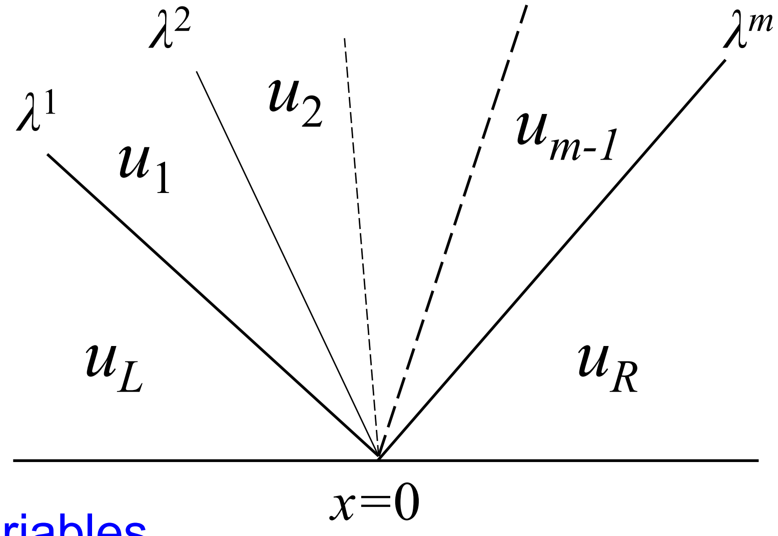
Or,

$$\boxed{\partial_t w^p + \lambda^p \partial_x w^p = 0} \qquad (p\text{=1,2,…,m})$$

This is a set of decoupled linear advection equations, with λ^p being wave speeds.

The Riemann problem (for a linear system)

$$\partial_t \boldsymbol{u} + \mathbf{A} \cdot \partial_x \boldsymbol{u} = 0$$



Initial condition: $\boldsymbol{u}$=$\boldsymbol{u}_L$, (x<0)

$\boldsymbol{u}$=$\boldsymbol{u}_R$, (x≥0)

Solution:

1). Decompose u_L, u_R into characteristic variables.

$$\boldsymbol{u}_{L,R} = \sum_p w^p_{L,R} \boldsymbol{r}^p$$

2). Each characteristic variable evolves according to its own characteristics.

$$w^p(x,t) = w^p_L \ \text{ if } x - \lambda^p t < 0\text{, otherwise, } \ w^p(x,t) = w^p_R$$

3). Convert back to original variables.

$$\boldsymbol{u}(x,t) = \sum_{p:\lambda^p < x/t} w^p_R \boldsymbol{r}^p + \sum_{p:\lambda^p > x/t} w^p_L \boldsymbol{r}^p$$

Boundary conditions

Now consider solving the previous problem in a bounded interval $a \leq x \leq b$.

With m waves, m boundary conditions (BCs) are needed.

The problem is more transparent when looking at characteristic variables:

$$\partial_t w^p + \lambda^p \partial_x w^p = 0$$

Clearly, if $\lambda^p > 0$, a BC must be supplied at $x = a$;
if $\lambda^p < 0$, a BC must be supplied at $x = b$.

Such BCs can be pure inflow, or some kind of reflection (as a function of incoming waves).

However, BCs are generally determined by the physical setup, which are usually not in terms of characteristic variables.

It helps greatly to know the characteristic structure, and to set the correct # of BCs to avoid over-constraining or under-determining the solution.

Hyperbolic PDEs and conservation laws

- An important class of hyperbolic PDEs is conservation laws:

$$\partial_t \boldsymbol{u} + \partial_x \boldsymbol{F}(\boldsymbol{u}) = 0$$

where $\boldsymbol{F}(\boldsymbol{u})$ is the flux function. It can be rewritten in quasi-linear form

$$\partial_t \boldsymbol{u} + \boldsymbol{F}'(\boldsymbol{u}) \cdot \partial_x \boldsymbol{u} = 0$$

It is hyperbolic if $\boldsymbol{F}'(\boldsymbol{u})$ is diagonalizable with real eigenvalues for all $\boldsymbol{u}$.

- Ideal MHD equations are (non-linear, multi-D) conservation laws:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) &= 0, \\
\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{B} + \mathbf{P}^*) &= 0, \\
\frac{\partial E}{\partial t} + \nabla \cdot [(E + P^*)\boldsymbol{v} - \boldsymbol{B}(\boldsymbol{B} \cdot \boldsymbol{v})] &= 0, \\
\frac{\partial \boldsymbol{B}}{\partial t} - \nabla \times (\boldsymbol{v} \times \boldsymbol{B}) &= 0,
\end{aligned}$$

Ideal MHD equations are hyperbolic because all wave speeds are real.

Example:

Recall: in our lecture on hydrodynamics, we linearized adiabatic hydrodynamic equations in 1D in primitive variables ***W***=(ρ, **v**, *P*):

$$\frac{\partial \boldsymbol{W}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{W}}{\partial x} = 0 \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} v_x & \rho & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 1/\rho \\ 0 & 0 & v_x & 0 & 0 \\ 0 & 0 & 0 & v_x & 0 \\ 0 & \gamma P & 0 & 0 & v_x \end{bmatrix}$$

We can redo the problem in conserved variables ***U***=(ρ, ρ**v**, *E*):

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{F}'(\boldsymbol{U})\frac{\partial \boldsymbol{U}}{\partial x} = 0$$

where $$\boldsymbol{F}'(\boldsymbol{U}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -v_x^2 + \gamma' v^2/2 & -(\gamma - 3)v_x & -\gamma' v_y & -\gamma' v_z & \gamma' \\ -v_x v_y & v_y & v_x & 0 & 0 \\ -v_x v_z & v_z & 0 & v_x & 0 \\ -v_x H + \gamma' v_x v^2/2 & -\gamma' v_x^2 + H & -\gamma' v_x v_y & -\gamma' v_x v_z & \gamma v_x \end{bmatrix}$$

and $H = (E + P)/\rho$

This matrix has exactly the same eigenvalues: it describes the same physics.

The eigensystem for MHD in conserved variables is even more complicated.

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Characteristics in non-linear equations

Consider scalar conservation law in quasi-linear form:

$$\partial_t u + f'(u)\partial_x u = 0$$

with $u(x, t=0)=u_0(x)$.

It's characteristics curve (along which solution is constant) satisfies

$$\boxed{X'(t) = f'[u(X(t),t)]}$$

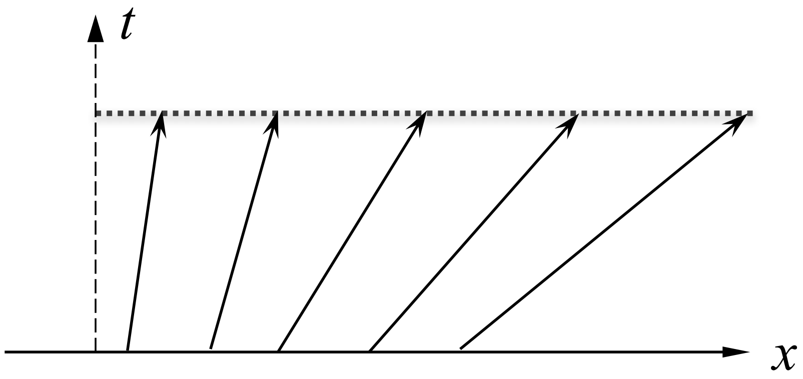
(proof is the same as in the linear problem)

Because u is constant along $X(t)$, it can be integrated to give:

$$X(t) = X_0 + f'[u_0(X_0)]t$$

(for a curve originating from $X=X_0$, $t=0$)

For scalar conservation law, u is constant on characteristics, which are straight lines, *as long as the solution remains smooth*.



Example: Berger's equation

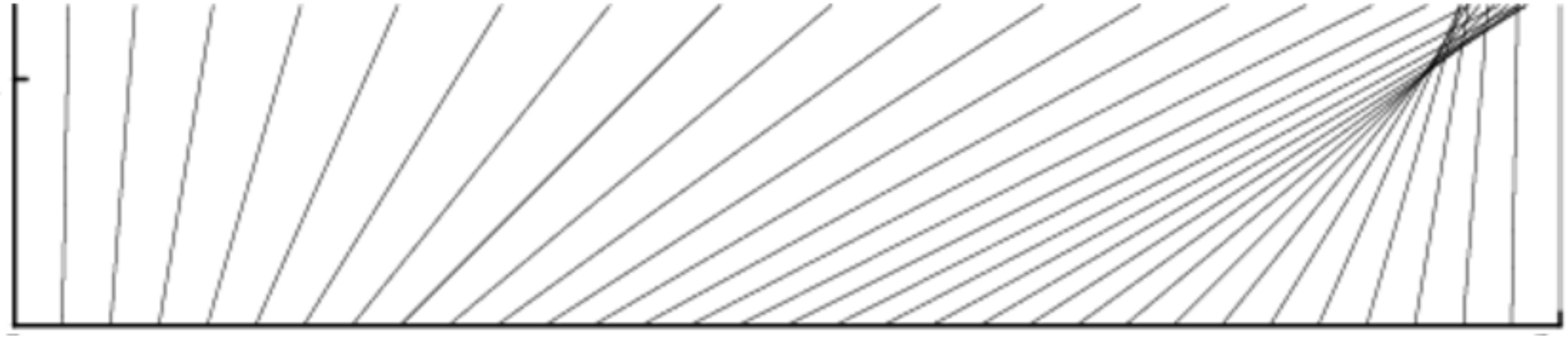
The simply non-linear conservation law is Berger's equation:

$$\partial_t u + \partial_x (u^2/2) = 0 \quad \text{or} \quad \partial_t u + u\partial_x u = 0$$

Characteristic curve: $X(t) = X_0 + u_0(X_0)t$

Characteristic curves converge when $\partial_x u_0 > 0$

Characteristic curves diverge when $\partial_x u_0 < 0$



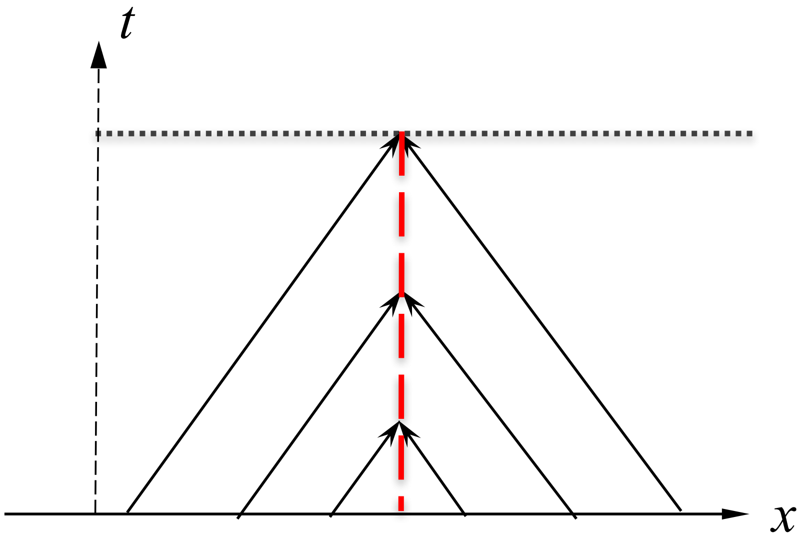
When converging, characteristic curves cross, which is unphysical!

When characteristics cross:

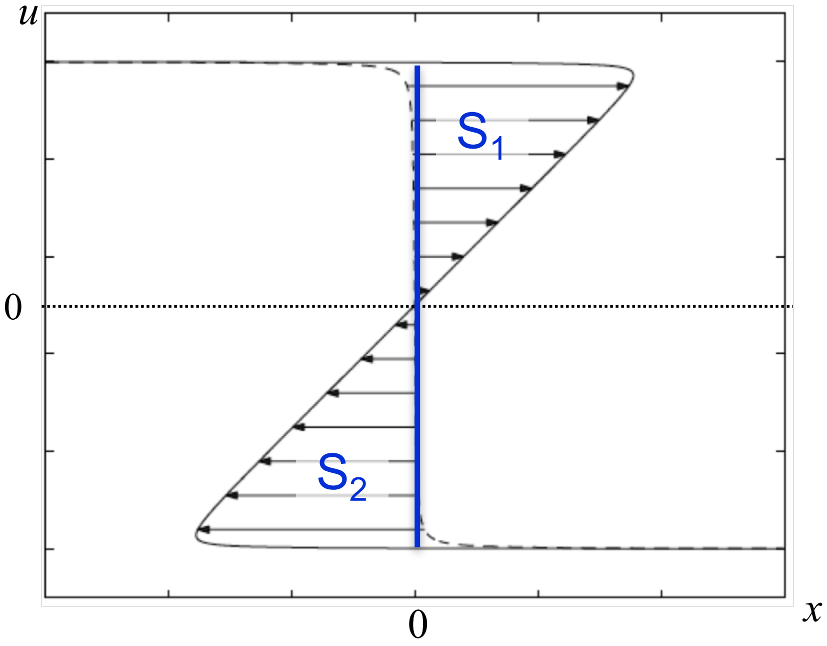
Consider the Riemann problem for Berger's equation: $\partial_t u + u\partial_x u = 0$ with u=u_L= $u_0/2$, (x<0)
u=u_R=-$u_0/2$, (x≥0)

Characteristics cross at x=0, forming a shock.

Characteristic curves



Profile of u



Equal area rule: the position of the shock corresponds to S_1=S_2.

Shock speed

Consider the Riemann problem for a general scalar conservation law:

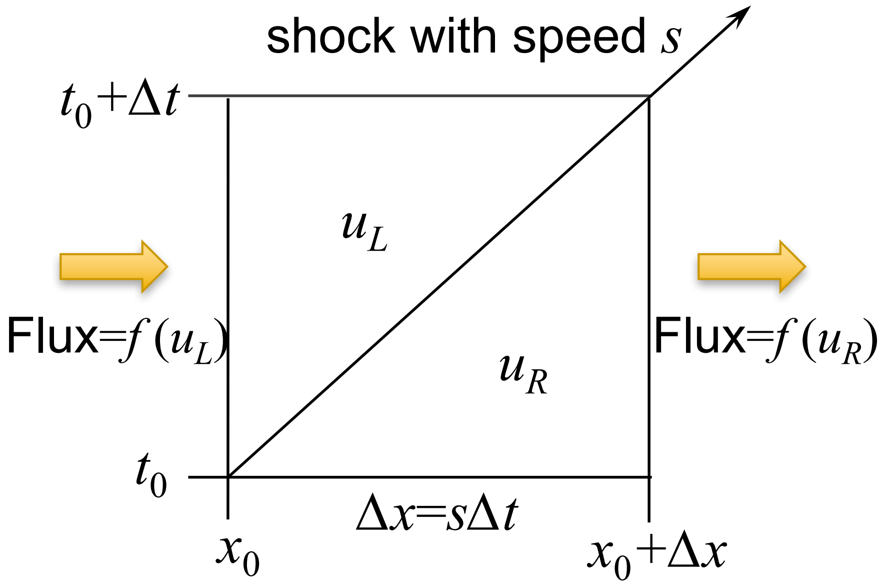
$$\partial_t u + \partial_x f(u) = 0 \quad \text{with} \quad \begin{array}{l} u=u_L,\ (x<0) \\ u=u_R,\ (x\geq 0) \end{array}$$

From the conservation law:

$$(u_L - u_R)\Delta x = [f(u_L) - f(u_R)]\Delta t$$

Now we can find the shock speed:

$$s = \frac{f(u_r) - f(u_l)}{u_r - u_l}$$



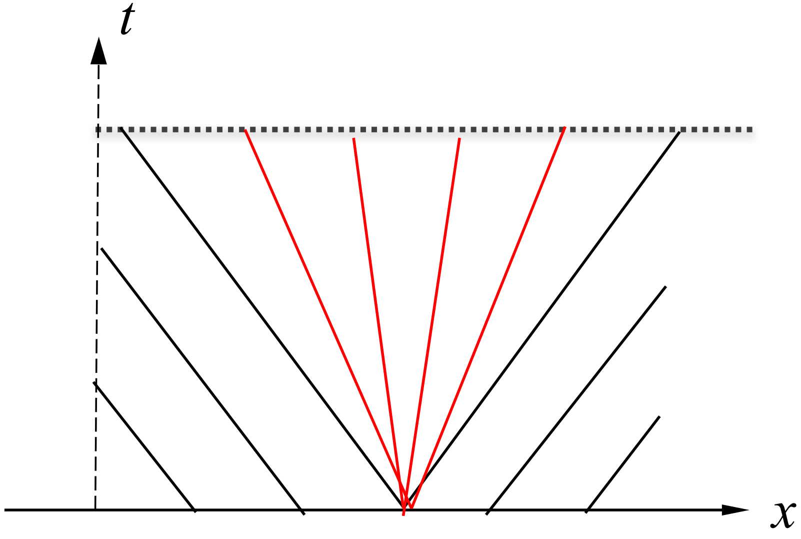
This can be generalized to system of equations, from which one finds Rankine-Hugoniot jump conditions.

Rarefaction wave:

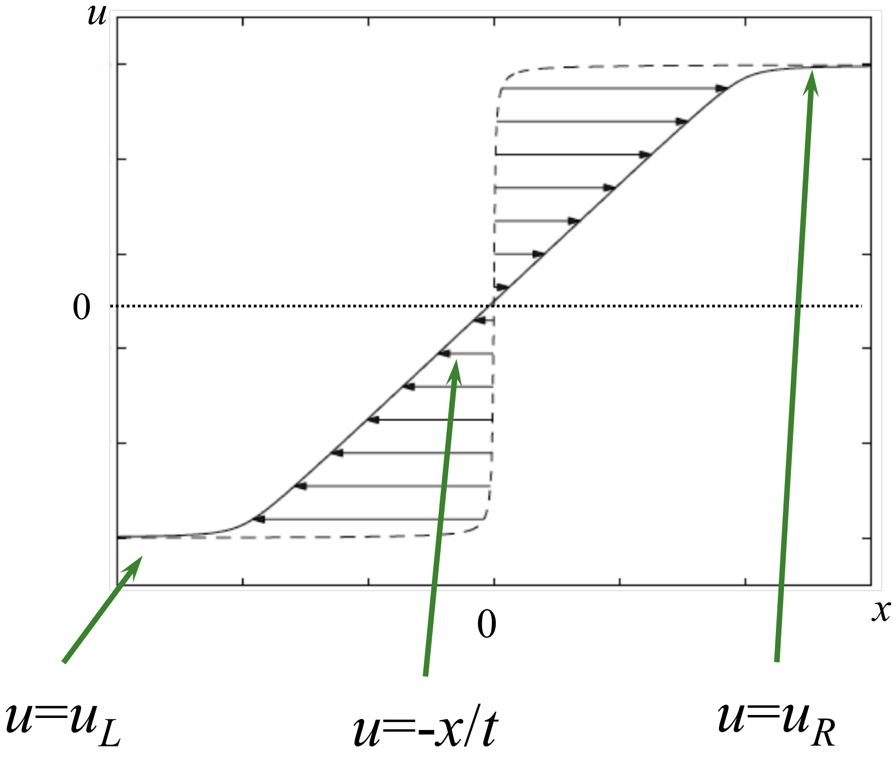
Consider the Riemann problem for Berger's equation: $\partial_t u + u\partial_x u = 0$ with $u=u_L=-u_0/2,\ (x<0)$ $u=u_R= u_0/2,\ (x\geq 0)$

Characteristics diverge at x=0, forming a rarefaction wave.

Characteristic curves



Profile of u



Weak solutions

At discontinuities, the classical form of the PDEs fail. These solutions are captured in the integral form (more fundamental):

$$\int_{x_1}^{x_2} [u(x,t_2) - u(x,t_1)]dx = \int_{t_1}^{t_2} F[u(x_1,t)] - F[u(x_2,t)]dt$$

for any t_1 ,t_2, x_1, x_2.

Solutions to this form are called weak solutions.

However, weak solutions are not necessarily unique.

One way to eliminate the non-uniqueness is by adding a small "viscous" term:

$$\partial_t u + \partial_x f(u) = \epsilon \partial_{xx} u$$

=> slightly smooth the solution => converges to the physical solution as $\varepsilon \to 0$.

Alternatively, physical solutions can be found by imposing certain "entropy conditions".

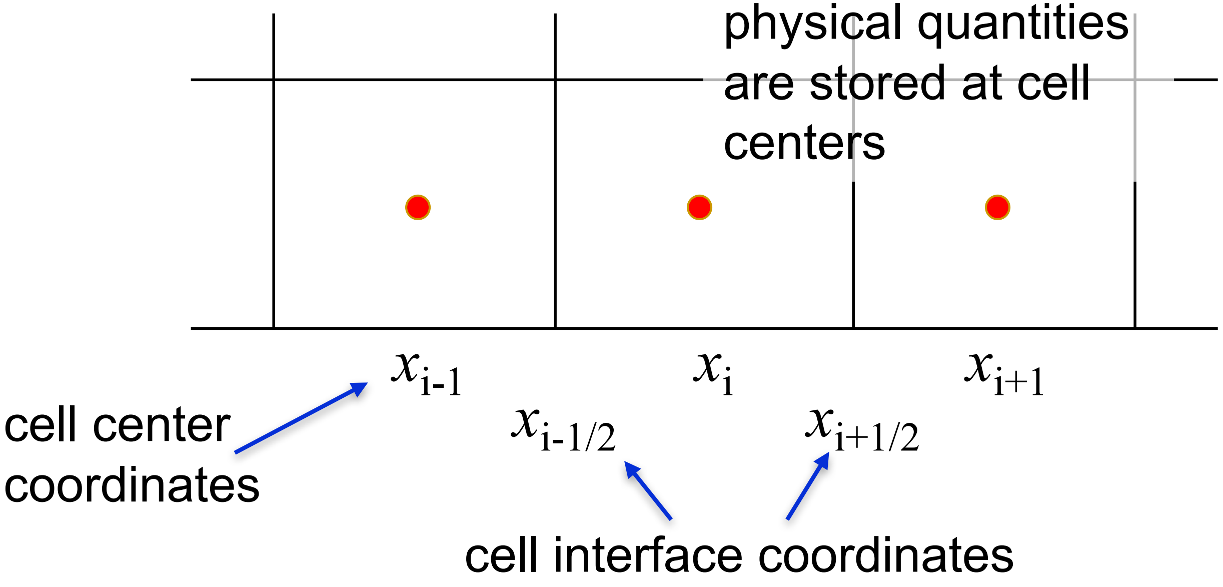
Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Discretization

Solving PDEs by grid-based methods inevitably involves discretization and use finite differencing to approximate time/spatial derivatives.

For simplicity, we focus on Cartesian grid with uniform grid spacing Δx.


physical quantities are stored at cell centers
x_{i-1}
x_i
x_{i+1}
$x_{i-1/2}$
$x_{i+1/2}$
cell center coordinates
cell interface coordinates

Order of accuracy in finite differencing

If a function $f(x)$ is sufficiently smooth:

$$f(x_0+\Delta x)=f(x_0)+f'_x(x_0)\Delta x+\frac{\Delta x^2}{2}f''_{xx}(x_0)+\frac{\Delta x^3}{6}f'''_{xxx}(x_0)+\ldots$$

To approximate $f'(x)$ at $x=x_0$, we may consider forward differencing

$$\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}=f'_x(x_0)+\boxed{\frac{\Delta x}{2}f''_{xx}(x_0)+\frac{\Delta x^2}{6}f'''_{xxx}(x_0)+\ldots}$$

1st order accurate

truncation error

Or central differencing

$$\frac{f(x_0+\Delta x)-f(x_0-\Delta x)}{2\Delta x}=f'_x(x_0)+\boxed{\frac{\Delta x^2}{6}f'''_{xxx}(x_0)+\ldots}$$

2nd order accurate

Solve the linear advection equation on a grid

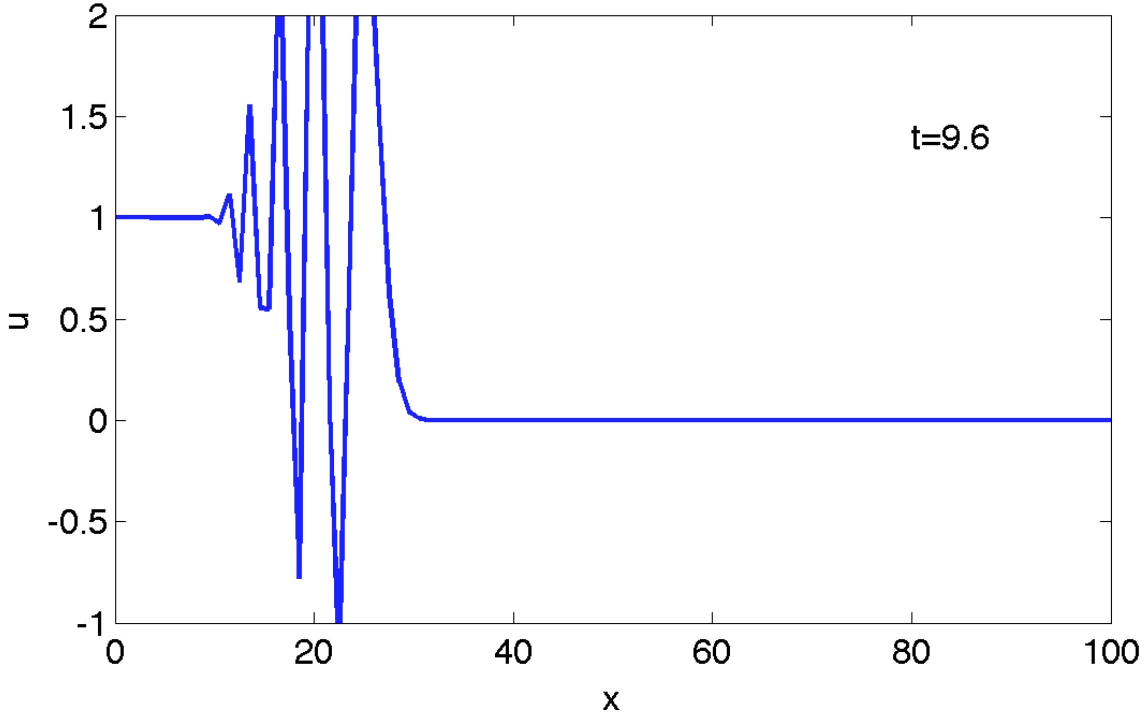
$$\partial_t u + A\partial_x u = 0$$

Initial condition: u=1 (x<50), u=0 (x>50), A=1.

1. Forward-time central-space (FTCS):

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

1st order in time
2nd order in space



The method is unconditionally unstable!

von Neumann stability analysis

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

Represent the discretized solution by a finite Fourier series (strictly speaking, this is appropriate only for linear problem in a periodic domain).

Pick up one mode of the form: $u_i^n = ae^{\mathrm{i}k(i\Delta x)}$

Solution at the next timestep is: $u_i^{n+1} = A_k u_i^n$

A$_k$: amplification factor (complex number)

For FTCS method, it is straightforward to obtain:

$$A_k = 1 - \mathrm{i}\frac{A\Delta t}{\Delta x}\sin(k\Delta x)$$

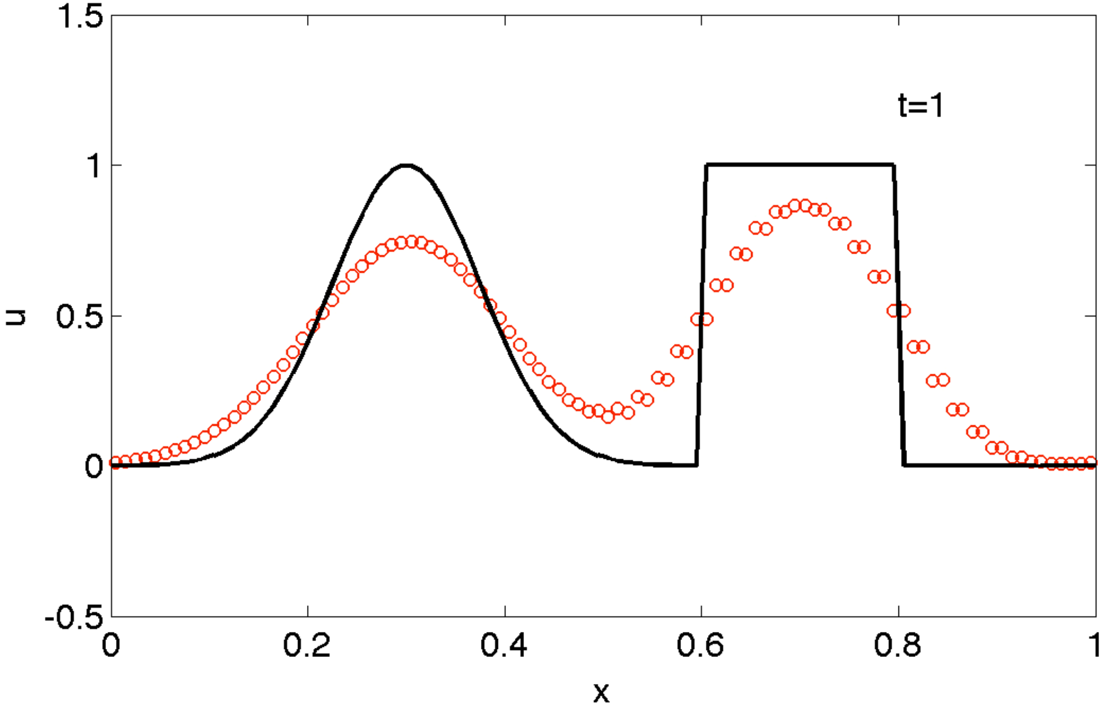
We see that no matter what timestep we choose, |*A*$_k$|>1. This means that the solution is exponentially amplified in time, being unconditionally unstable!

Lax-Friedrichs (LF) method

$$\partial_t u + A\partial_x u = 0$$

IC: one Gaussian, one square waves, A=1, periodic BC.

$$\frac{u_i^{n+1} - (u_{i-1}^n + u_{i+1}^n)/2}{\Delta t} = -A\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right)$$



Effectively, added extra numerical diffusion to stabilize FTCS.

The method is stable, but VERY diffusive!

von Neumann stability analysis

$$\frac{u_i^{n+1} - (u_{i-1}^n + u_{i+1}^n)/2}{\Delta t} = -A\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right)$$

Take a Fourier mode of the form: $u_i^n = ae^{\mathrm{i}k(i\Delta x)}$

Solution at the next timestep is: $u_i^{n+1} = A_k u_i^n$

For the Lax-Friedrich method, it is straightforward to obtain:

$$A_k - \cos(k\Delta x) = -\mathrm{i}\frac{A\Delta t}{\Delta x}\sin(k\Delta x)$$

Clearly, the method is stable (|A_k|<1) if $\frac{A\Delta t}{\Delta x} < 1$, or $\boxed{\Delta t < \frac{\Delta x}{A}}$

Why is this method stable but diffusive?

$$\frac{u_i^{n+1} - (u_{i-1}^n + u_{i+1}^n)/2}{\Delta t} = -A\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right)$$

Rewriting its formulation we obtain:

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

$$\sim \frac{\Delta x^2}{2\Delta t}\partial_{xx}u$$

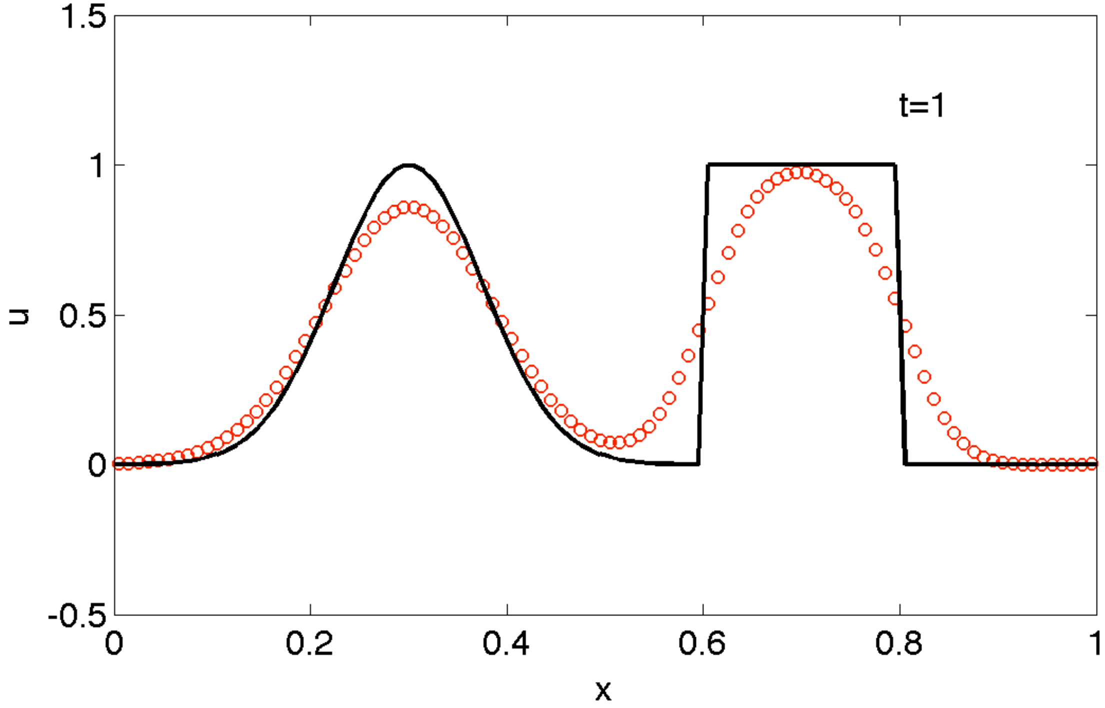
Essentially, this adds (a lot of) numerical diffusion.

Upwind method

$$\partial_t u + A\partial_x u = 0$$

IC: one Gaussian, one square waves, A=1, periodic BC.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \begin{cases} -A(u_i^n - u_{i-1}^n)/\Delta x & (A \geq 0) \\ -A(u_{i+1}^n - u_i^n)/\Delta x & (A < 0) \end{cases}$$



The method is stable and less diffusive, though still only first order accurate.

Can be improved using higher-order spatial interpolation schemes.

von Neumann stability analysis

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -A(u_i^n - u_{i-1}^n)/\Delta x \quad (A \geq 0)$$

This can be rewritten as: $u_i^{n+1} = (1-\mu)u_i^n + \mu u_{i-1}^n$ where $\mu \equiv \frac{A\Delta t}{\Delta x}$

Setting $u_i^n = a e^{\mathrm{i}k(i\Delta x)}$ and $u_i^{n+1} = A_k u_i^n$, we find

$$A_k = (1-\mu) + \mu e^{-\mathrm{i}k\Delta x}$$

and hence $|A_k|^2 = (1 - \mu + \mu e^{-\mathrm{i}k\Delta x})(1 - \mu + \mu e^{\mathrm{i}k\Delta x})$

$$= 1 - 2\mu(1-\mu)(1 - \cos k\Delta x)$$

Stability requires $\mu(1-\mu) > 0$, or again $\boxed{\Delta t < \frac{\Delta x}{A}}$

Modified equation analysis

Numerical methods solve PDEs only approximately.

For a given method, is there a PDE to which our numerical method gives an exact solution?

For the upwind method for linear advection problem, suppose $v(x,t)$ is the numerical solution satisfying:

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = -A\left(\frac{v_i^n - v_{i-1}^n}{\Delta x}\right) \; , \quad (A \geq 0)$$

Now we Taylor expand $v(x,t)$ to obtain

$$\partial_t v + \frac{1}{2}\Delta t \partial_{tt} v + \ldots = -A(\partial_x v - \frac{1}{2}\Delta x \partial_{xx} v + \ldots)$$

$$\Rightarrow \qquad \partial_t v + A\partial_x v = -\frac{1}{2}\Delta t \partial_{tt} v + \frac{A}{2}\Delta x \partial_{xx} v + \ldots$$

Modified equation analysis

$$\partial_t v + A\partial_x v = -\frac{1}{2}\Delta t \partial_{tt} v + \frac{A}{2}\Delta x \partial_{xx} v + \ldots$$

This is the equation that v satisfies, but not quite convenient: want to eliminate the time derivatives on the RHS.

We can differentiate the above w.r.t. t and x, to obtain:

$$\begin{aligned}\partial_{tt} v + A\partial_{xt} v &= -\frac{1}{2}\Delta t \partial_{ttt} v + \frac{A}{2}\Delta x \partial_{xxt} v + \ldots \\ \partial_{tx} v + A\partial_{xx} v &= -\frac{1}{2}\Delta t \partial_{ttx} v + \frac{A}{2}\Delta x \partial_{xxx} v + \ldots\end{aligned} \quad \Rightarrow \quad \partial_{tt} v = A^2 \partial_{xx} v + O(\Delta t)$$

Here we arrive at the desired modified equation:

$$\partial_t v + A\partial_x v = \frac{A}{2}(\Delta x - A\Delta t)\partial_{xx} v + O(\Delta t^2)$$

Numerical diffusivity depends on Δt

This is a diffusion term!

Towards higher order of accuracy

$$\partial_t u + A\partial_x u = 0$$

IC: one Gaussian, one square waves, A=1, periodic BC.

We would like to construct a scheme that minimize numerical diffusion.

For a smooth solution $u(x,t)$, we can Taylor expand it as

$$u(x,t+\Delta t) = u(x,t) + \Delta t\partial_t u(x,t) + \frac{1}{2}\Delta t^2\partial_{tt}u(x,t) + ...$$

$$= u(x,t) - A\Delta t\partial_x u(x,t) + \frac{1}{2}A^2\Delta t^2\partial_{xx}u(x,t) + ...$$

This motivates us to consider the following scheme (Lax-Wendroff):

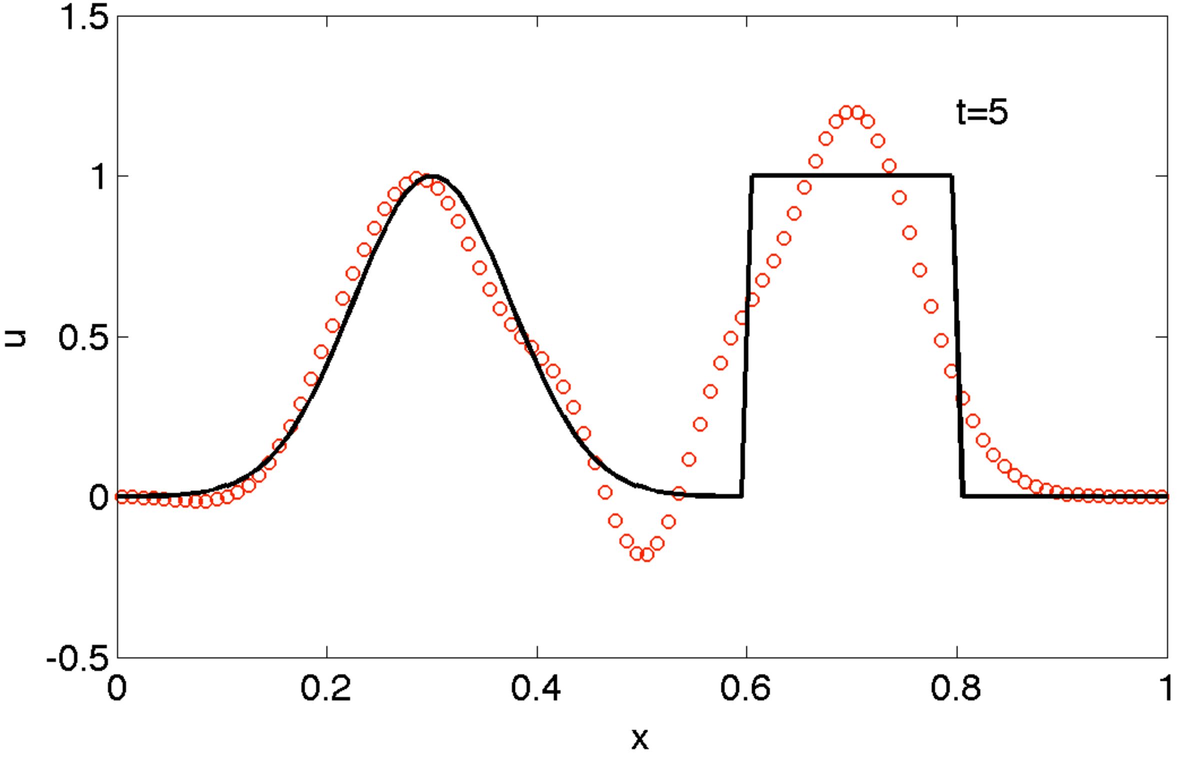
$$u_i^{n+1} = u_i^n - \frac{A\Delta t}{2\Delta x}(u_{i+1}^n - u_{i-1}^n) + \frac{A^2\Delta t^2}{2\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n)$$

Lax-Wendroff method

$$\partial_t u + A\partial_x u = 0$$

IC: one Gaussian, one square waves, A=1, periodic BC.

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



Method is stable but:

1). Oscillatory solution at discontinuities.

2). Phase shift in the smooth region.

Modified equation analysis

Going through the same procedures for the Lax-Wendroff method, we can find that the corresponding modified equation reads

$$\partial_t v + A\partial_x v = -\frac{A}{6}\Delta x^2\left[1-\left(\frac{A\Delta t}{\Delta x}\right)^2\right]\partial_{xxx} v$$

This term is dispersive!

Note on dispersive behavior:

Consider a prototype equation of the form $\partial_t v + a\partial_{xxx} v = 0$

For a wave-like solution $v(x,t) = Ce^{\mathrm{i}(kx-\omega t)}$

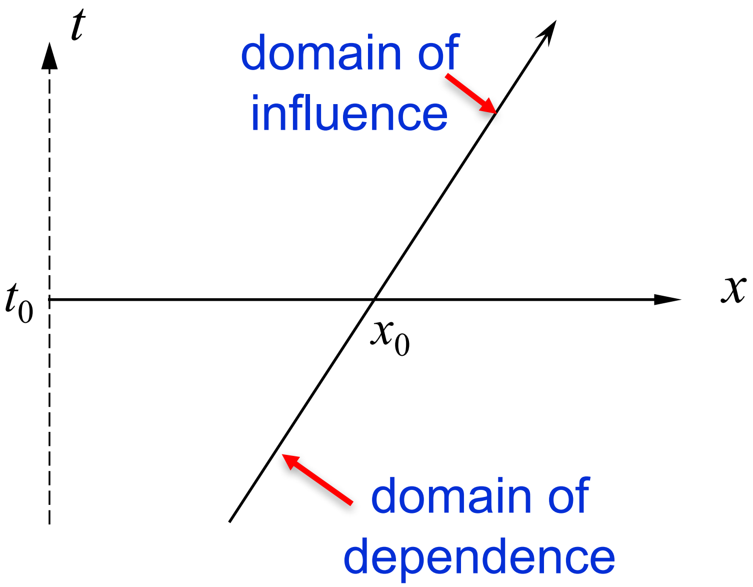
We obtain a dispersion relation: $\omega = -ak^3$

The waves are dispersive: phase speed ω/k depends on k.

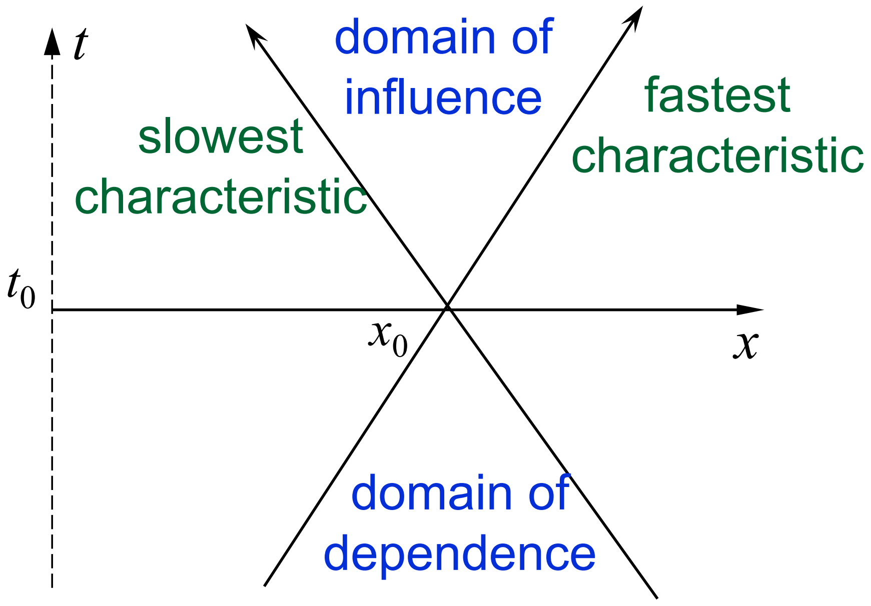
Domain of dependence/influence

The stable timestepping of numerical methods is closely related to the concept of domain of dependence and domain of influence.

For a scalar advection Eq.:



For a system of hyperbolic Eqs.:



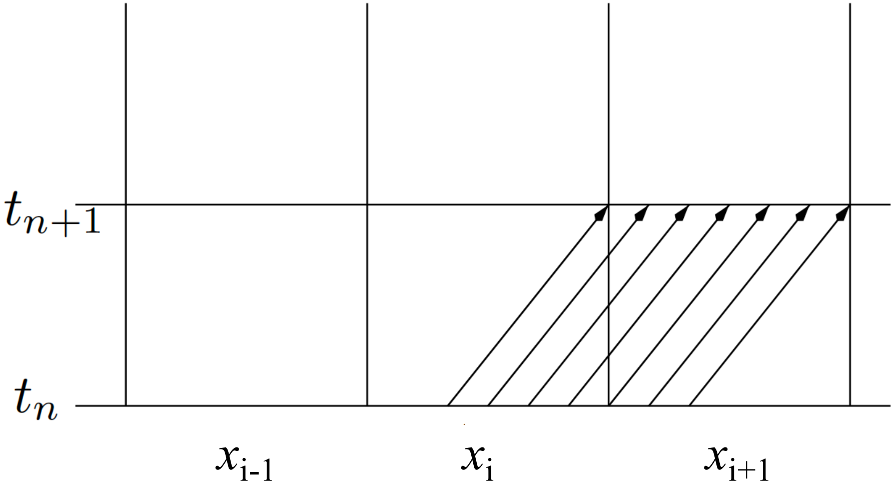
The Courant-Friedrichs-Lewy (CFL) condition

- A numerical method is convergent only if its domain of dependence contains the true domain of dependence of the PDE.
- In other words, the timestep Δt must be sufficiently small so that information propagates no more than one grid point per timestep.

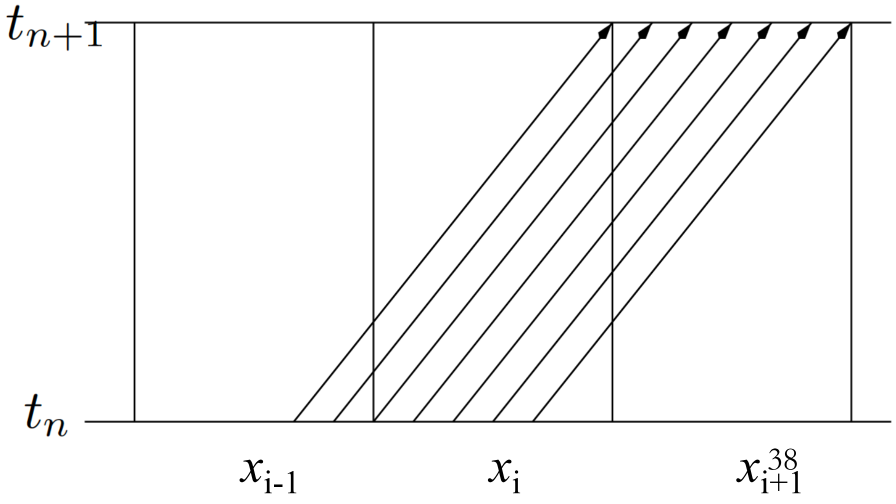
For the linear advection problem $\partial_t u + A\partial_x u = 0$,

$$\Delta t \equiv C\frac{\Delta x}{A} \quad , \quad \text{where the } \textbf{CFL number } C \leq 1.$$

When time step is appropriate (C<1)



When time step is too large (C>1)

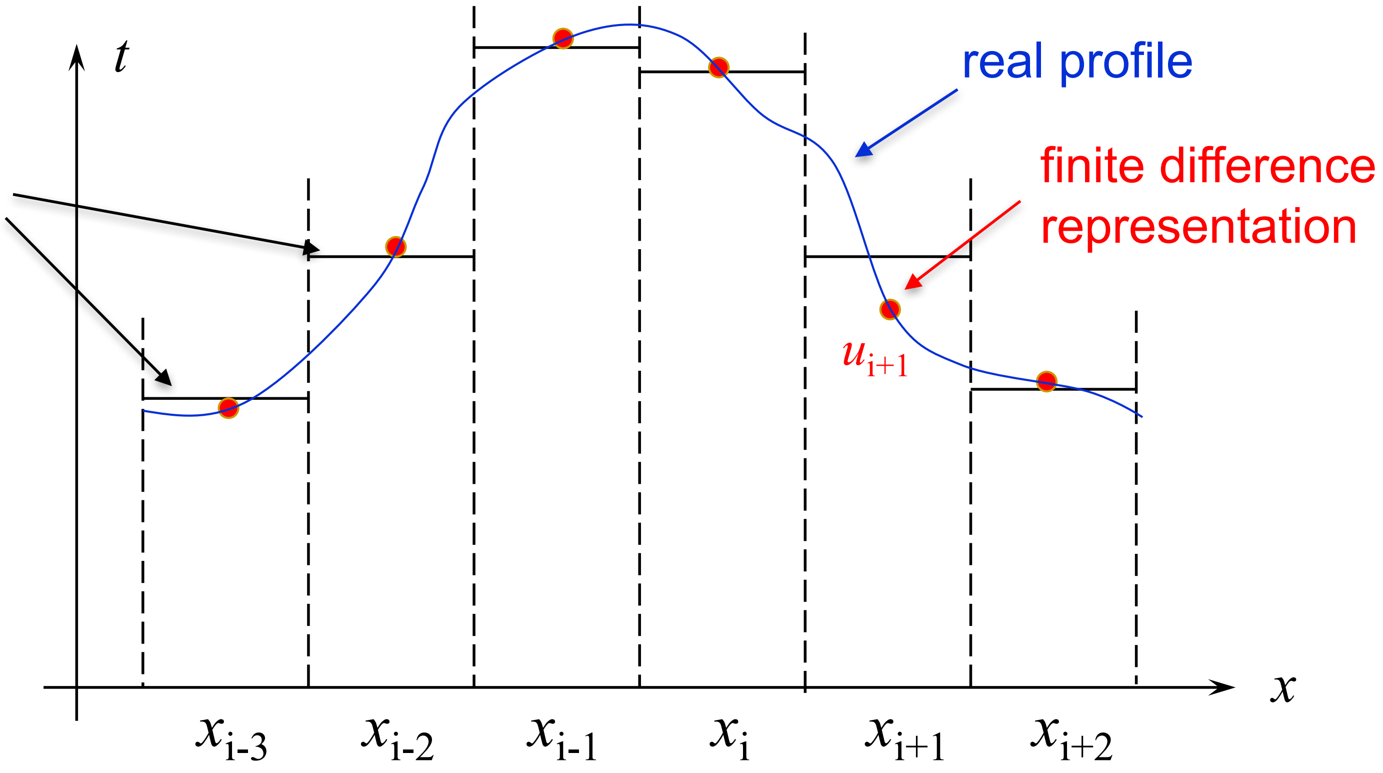


Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Finite volume vs finite difference methods

FVM interprets grid discretization very differently.



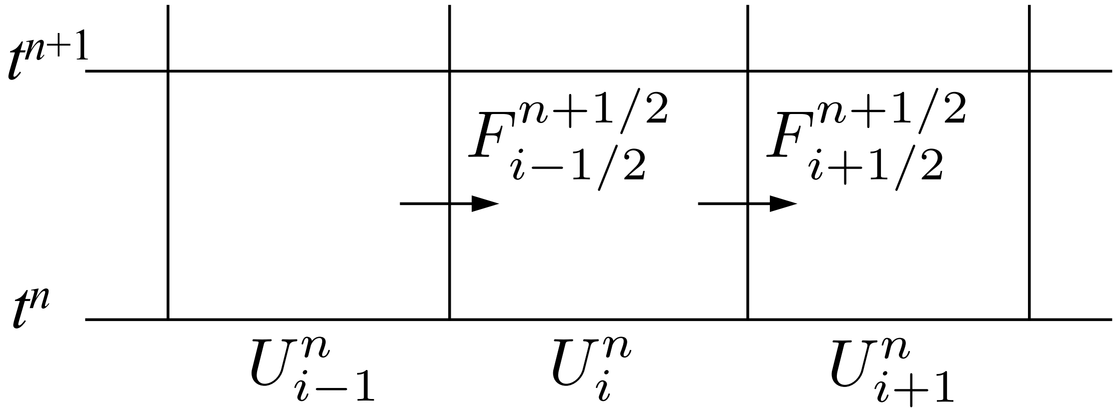
- FVM works with the integral form of the conservation laws.

Conserved variables are volume-averaged:

$$U_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t_n)dx$$

Finite volume method

- FVM works with the integral form of the conservation laws.



Conserved variables are volume-averaged:

$$U_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n) dx$$

Interface fluxes are time-averaged:

$$F_{i-1/2}^{n+1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2}, t)) dt$$

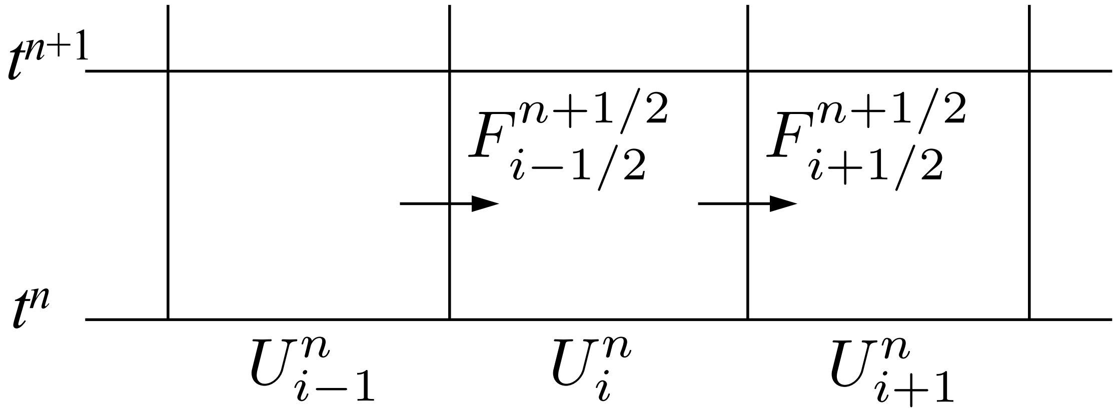
Finite volume update:

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x} (F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2})$$

Conserved variables are conserved to machine accuracy.

How to compute the fluxes?

Consider a general scalar conservation law: $\partial_t u + \partial_x f(u) = 0$



We only know the volume-averaged values U.

To get the flux at cell interfaces, we essentially need to know the value of u at $x_{i+1/2}$ through some sort of averaging/interpolation.

Alternatively, need to find ways to approximate interface fluxes directly.

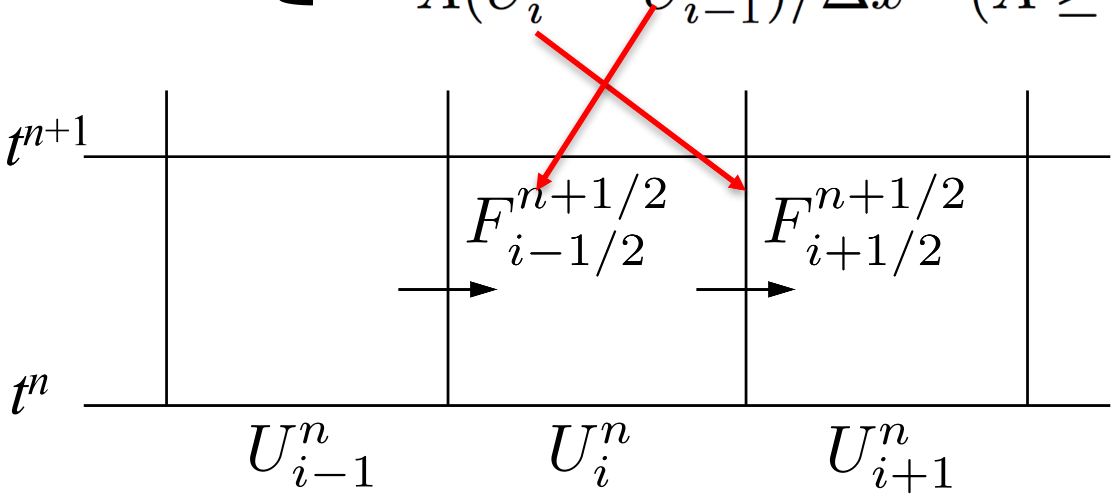
This is the key to finite volume methods.

Linear advection equation

For the linear advection equation, the aforementioned finite-difference methods can be given finite-volume interpretations:

For the upwind method:

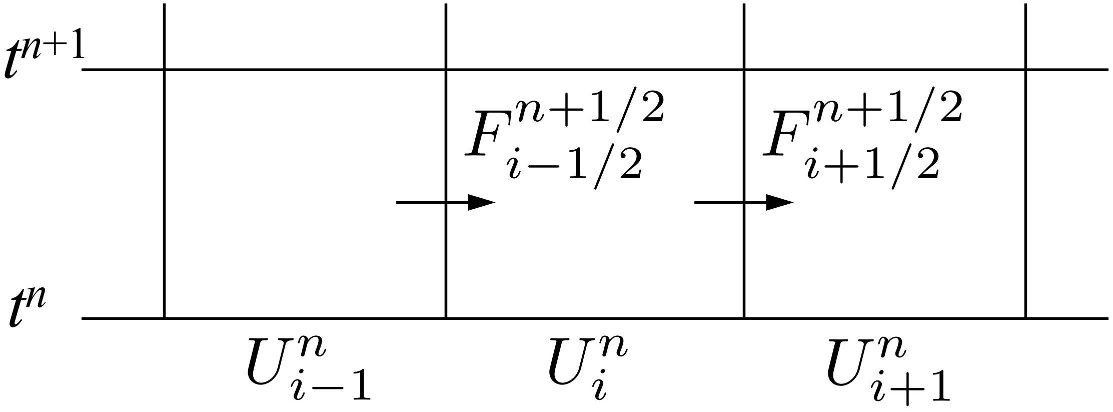
$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = \begin{cases} -A(U_{i+1}^n - U_i^n)/\Delta x & (A < 0) \\ -A(U_i^n - U_{i-1}^n)/\Delta x & (A \geq 0) \end{cases}$$



Upwind flux: $F_{i-1/2} = \begin{cases} AU_{i-1} & (A \geq 0) \\ AU_i & (A < 0) \end{cases}$

Linear advection equation

Similarly, Lax-Friedreich and Lax-Wendroff methods can also be interpreted in the finite volume framework:



Lax-Friedrichs:

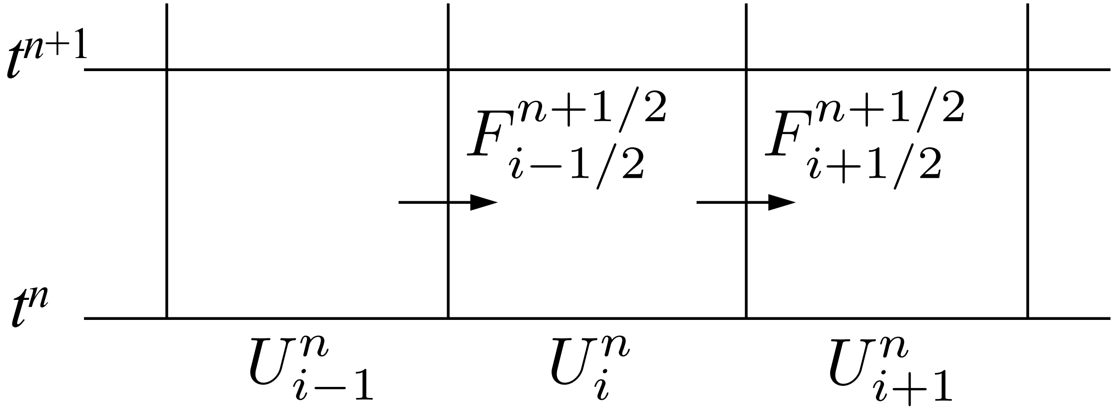
$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = -A\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right) + \frac{\Delta x^2}{2\Delta t}\left(\frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x^2}\right)$$

$$\Downarrow$$

$$F_{i-1/2} = \frac{1}{2}A(U_{i-1}+U_i) - \frac{\Delta x}{2\Delta t}(U_i - U_{i-1})$$

Linear advection equation

Similarly, Lax-Friedreich and Lax-Wendroff methods can also be interpreted in the finite volume framework:



Lax-Wendroff:

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

$$\Downarrow$$

$$F_{i-1/2} = \frac{1}{2}A(U_{i-1}+U_i) - \frac{\Delta t}{2\Delta x}A^2(U_i - U_{i-1})$$

Godunov method (Basic idea) (Godunov, 1959)

- 1. Given volume averaged values U_i^n (defined at each cell), reconstruct piecewise polynomial function $\tilde{u}^n(x)$ (defined at all x).

  Simplest scenario (piecewise constant/donor cell):

  $$\tilde{u}^n(x) = U_i^n \quad \text{for } x_{i-1/2} \le x < x_{i_1/2}$$

- 2. Using $\tilde{u}^n(x)$ as initial condition, evolve the hyperbolic equation exactly (or approximately) for Δt to obtain $\tilde{u}^{n+1}(x)$.

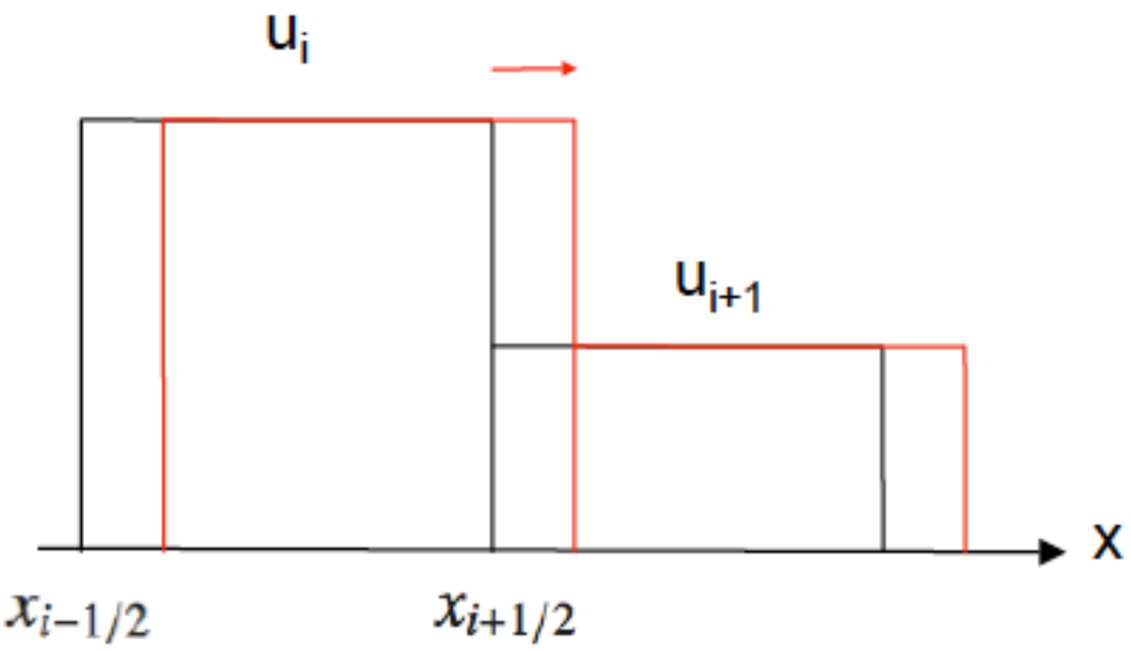
- 3. Average $\tilde{u}^{n+1}(x)$ over each cell to obtain new cell averages:

$$U_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \tilde{u}^{n+1}(x) dx$$

Godunov method (Basic idea)

A finite volume method originally proposed by Godunov (1959) for solving (non-linear) equations of gas dynamics.

Reconstruct ->
Evolve ->
Average



For linear advection equations, Godunov method with piecewise constant reconstruction = upwind method.

Key property: flux is properly upwinded to avoid spurious oscillations.

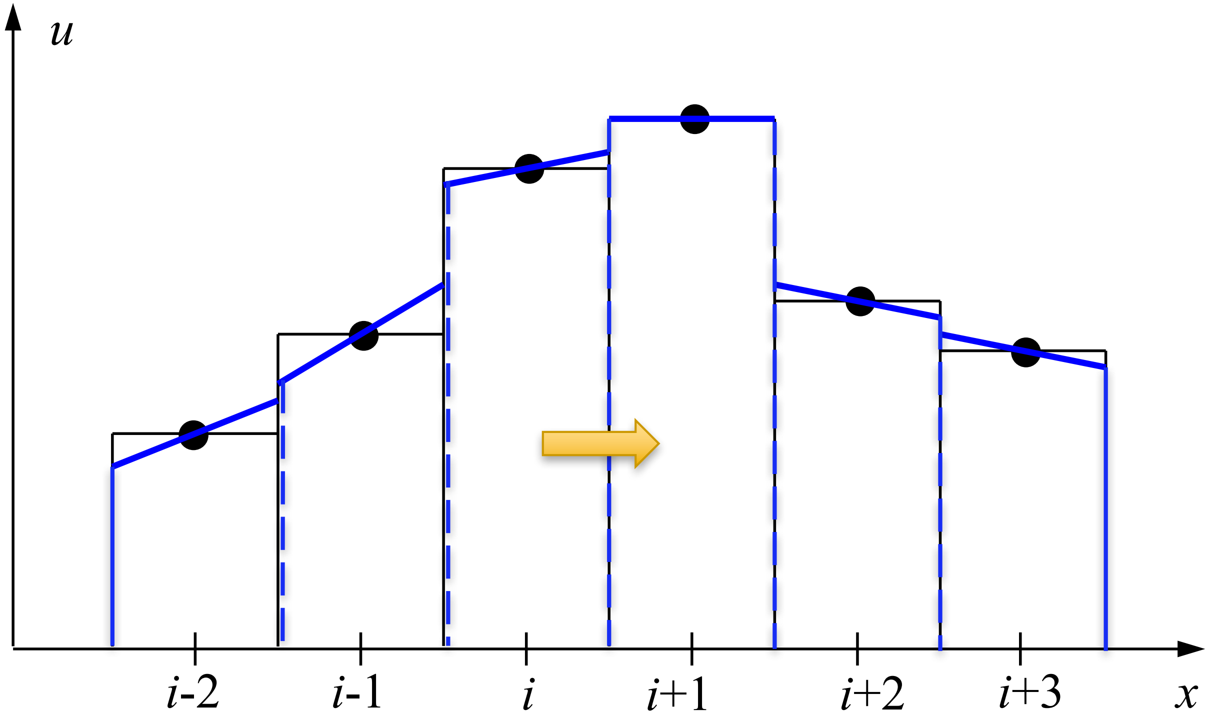
Toward higher order accuracy

Piecewise linear reconstruction:

$$u(x) = u_i + \sigma_i (x - x_i) \quad \text{for } (x_{i-1/2} \le x < x_{i+1/2})$$

Evolve reconstructed profile according to the (linear advection) equation.

Volume average the evolved profile to the grid structure.



Can also be done at 3rd order: Piecewise-parabolic method (Colella & Woodward, 1984)

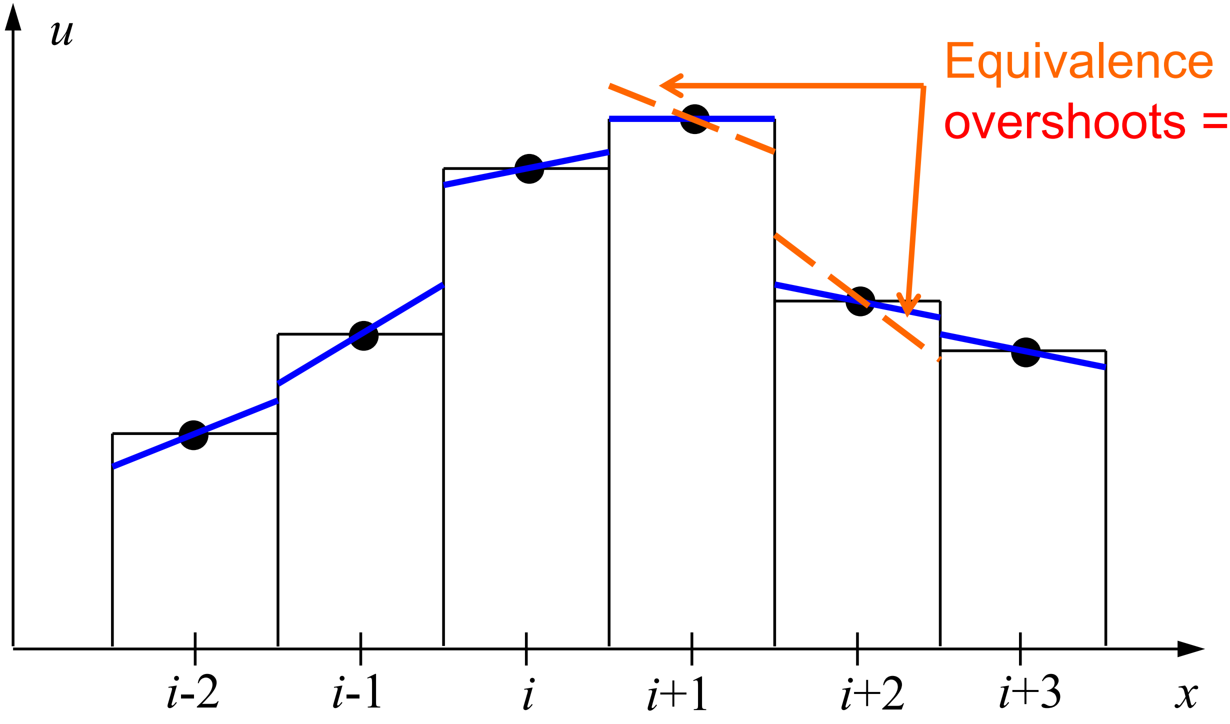
Toward higher order accuracy

How to choose the slopes?

$$u(x) = u_i + \sigma_i(x - x_i) \quad \text{for } (x_{i-1/2} \leq x < x_{i+1/2})$$

Simplest choice: $\sigma_i = (u_{i+1} - u_{i-1})/2\Delta x$

This is in fact equivalent to the Lax-Wendroff method.



Equivalence of the Lax-Wendroff:
overshoots => oscillations

Slope limiters

A slope limiter needs to be “monotonicity preserving”: should not produce any additional local extrema.

- Effectively, slope limiters reduce the order of accuracy to 1st order near discontinuities.
- This makes sense because near discontinuities, higher order derivatives are not well defined.

It can be shown that “total variation diminishing” (TVD) limiters are monotonicity preserving, where “total variation” is defined as

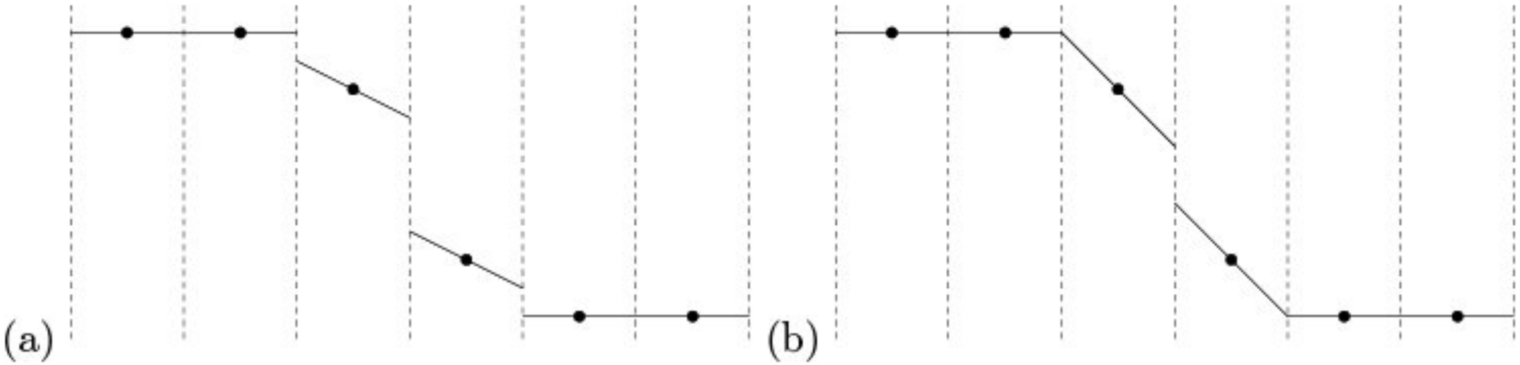
$$\mathrm{TV}(U) = \sum_{i=-\infty}^{\infty} |U_i - U_{i-1}|$$

TVD means $\mathrm{TV}(U^{n+1}) \leq \mathrm{TV}(U^n)$.

Popular choices of slope limiters include minmod, MC, van Leer, etc.

Slope limiters

Popular choices of slope limiters include minmod, MC, van Leer, etc.



minmod monotonized central-difference (MC)

minmod limiter: $$\sigma_i = \text{minmod}\left(\frac{U_i - U_{i-1}}{\Delta t}, \frac{U_{i+1} - U_i}{\Delta x}\right)$$

MC limiter: $$\sigma_i = \text{minmod}\left(\frac{U_{i+1} - U_{i-1}}{2\Delta x}, 2\frac{U_i - U_{i-1}}{\Delta x}, 2\frac{U_{i+1} - U_i}{\Delta x}\right)$$

where $$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab \leq 0. \end{cases}$$

Flux limiter

The procedure of slope limiting can be understood as a switch between from high-resolution method (Lax-Wendroff) and low-resolution method (upwind).

One can also interpret this process as a switch on interface fluxes between high and low resolutions:

$$F_{i-1/2}(U_{i-1},U_i) = F_{\text{low}}(U_{i-1},U_i) + \phi(r)[F_{\text{high}}(U_{i-1},U_i) - F_{\text{low}}(U_{i-1},U_i)]$$

Flux limiter

where $r = \dfrac{U_{i-1} - U_{i-2}}{U_i - U_{i-1}}$ (assuming A>0).

There is a one-to-one correspondence between flux limiter and slope limiter (for the linear advection problem).

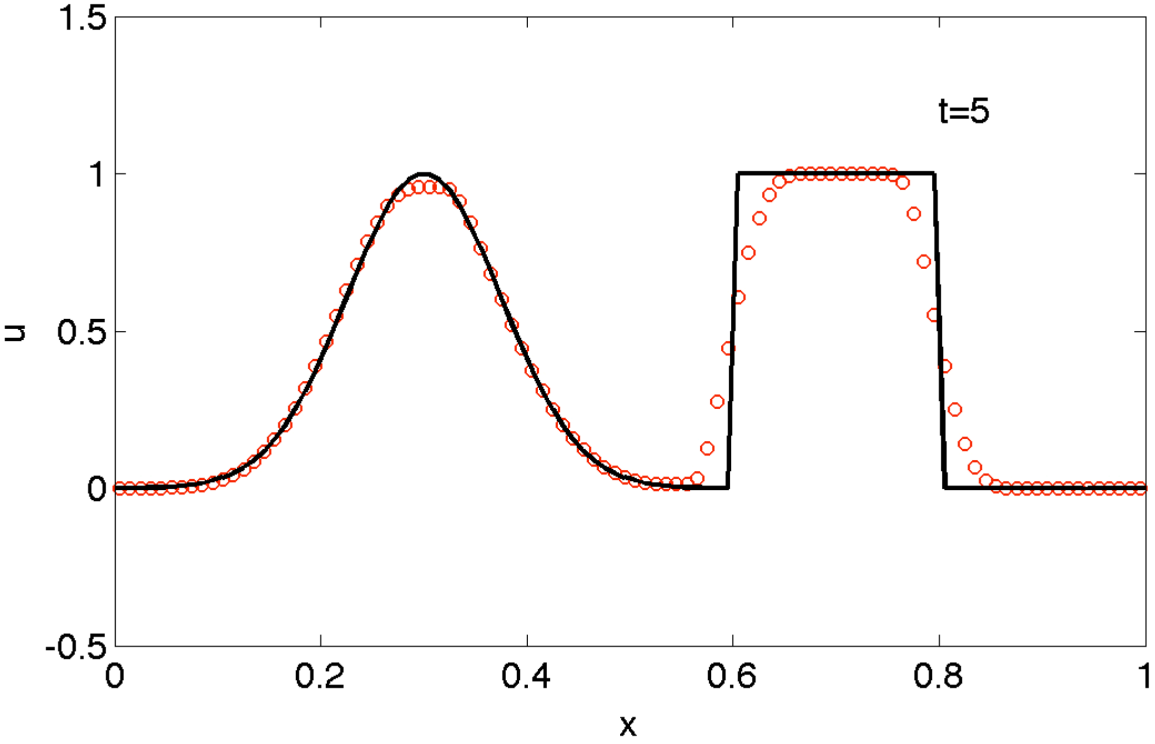
Solve the linear advection equation

Higher-order Godunov method with piecewise linear reconstruction

+the MC slope limiter.

$$\partial_t u + A \partial_x u = 0$$

Initial condition: one Gaussian, one square waves, $A=1$, periodic BC.



The method is stable and much more accurate:

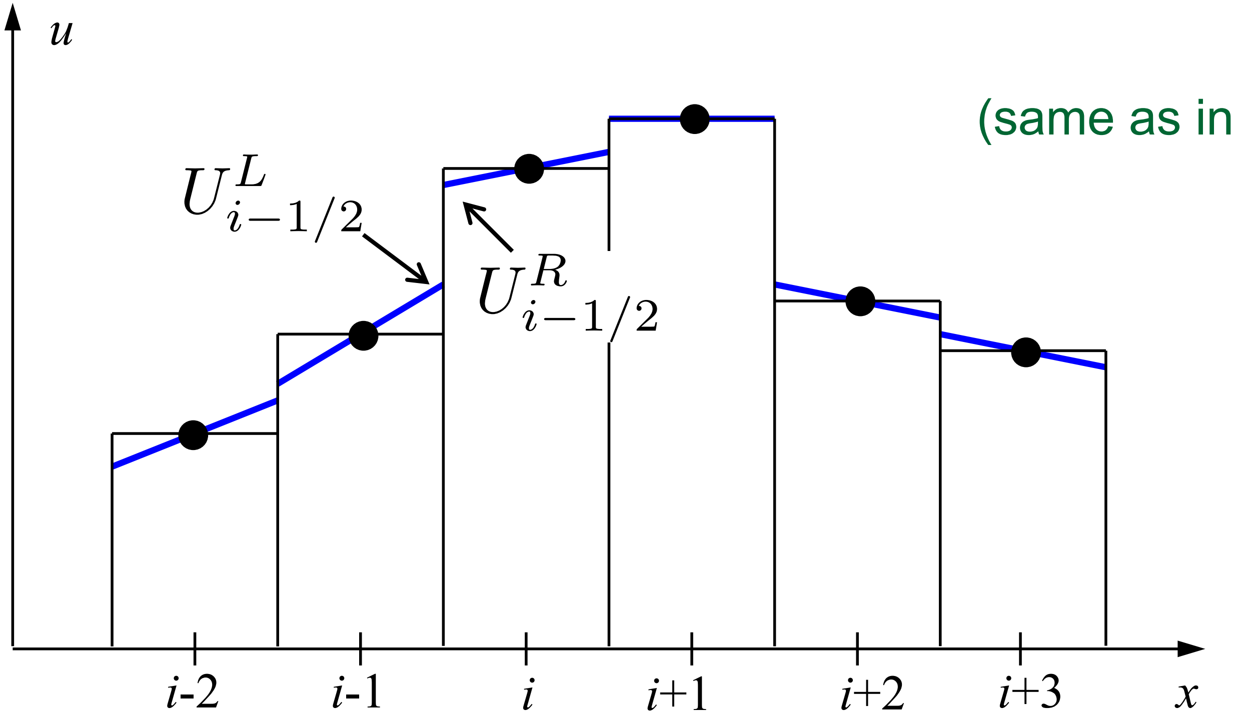
2nd order accurate for smooth flow
1st order accurate at discontinuities.

Non-linear scalar conservation law

To solve a non-linear scalar conservation law with Godunov method:

$$\partial_t u + \partial_x f(u) = 0$$

1. Given volume averaged values U_i^n, compute the left and right states $U_{L,i\text{-}1/2}/U_{R,i\text{-}1/2}$ at cell interfaces based on a reconstruction method.



(same as in the linear problem)

Non-linear scalar conservation law

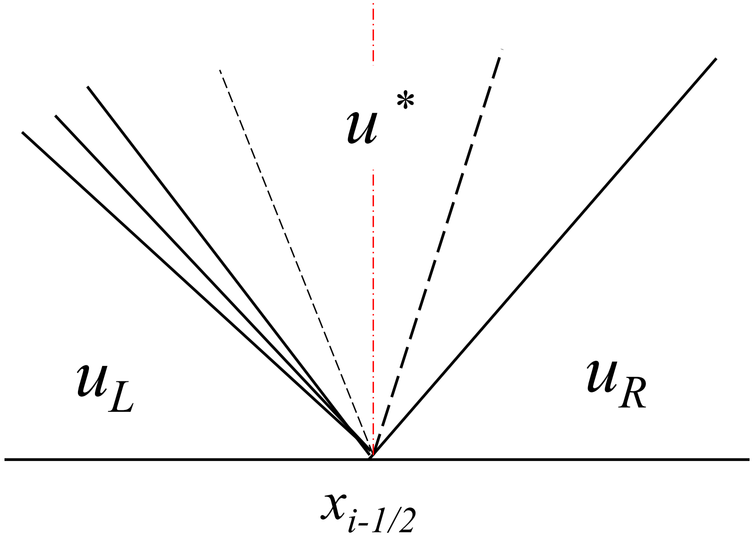
2. However, one generally can no longer directly evolve the system. This is replaced by a Riemann solver.

$$\partial_t u + \partial_x f(u) = 0 \quad \text{with} \quad \begin{array}{l} u=u_L,\ (x<0) \\ u=u_R,\ (x\geq 0) \end{array}$$

The key to a Riemann solver is to return the intermediate state u^* so that the interface flux is given by:

$$F_{i+1/2}=f(u^*)$$

(This approach automatically captures the shocks with dissipation)



3. Apply the flux-differencing formula:

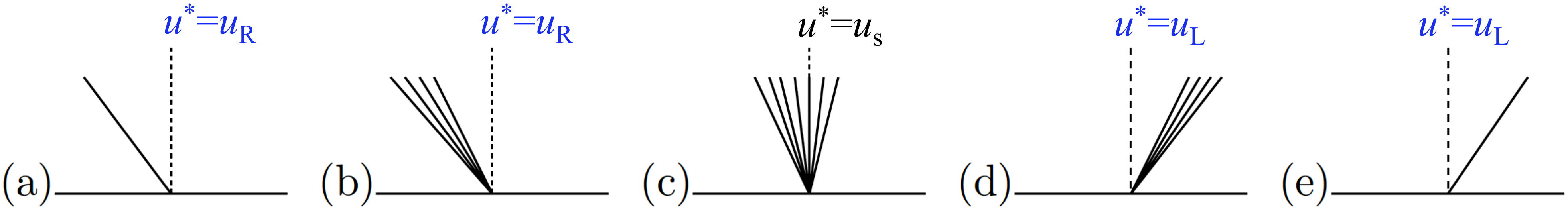
$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2} - F_{i-1/2})$$

Riemann problem (non-linear scalar conservation law)

For a non-linear scalar conservation law

$$\partial_t u + \partial_x f(u) = 0 \quad \text{with} \quad \begin{array}{l} u=u_L,\ (x<0) \\ u=u_R,\ (x\geq 0) \end{array}$$

The Riemann problem has only 5 possibilities:



shock | rarefaction | transonic rarefaction | rarefaction | shock

where u_s is the value at which $f'(u)=0$ (stagnation point).

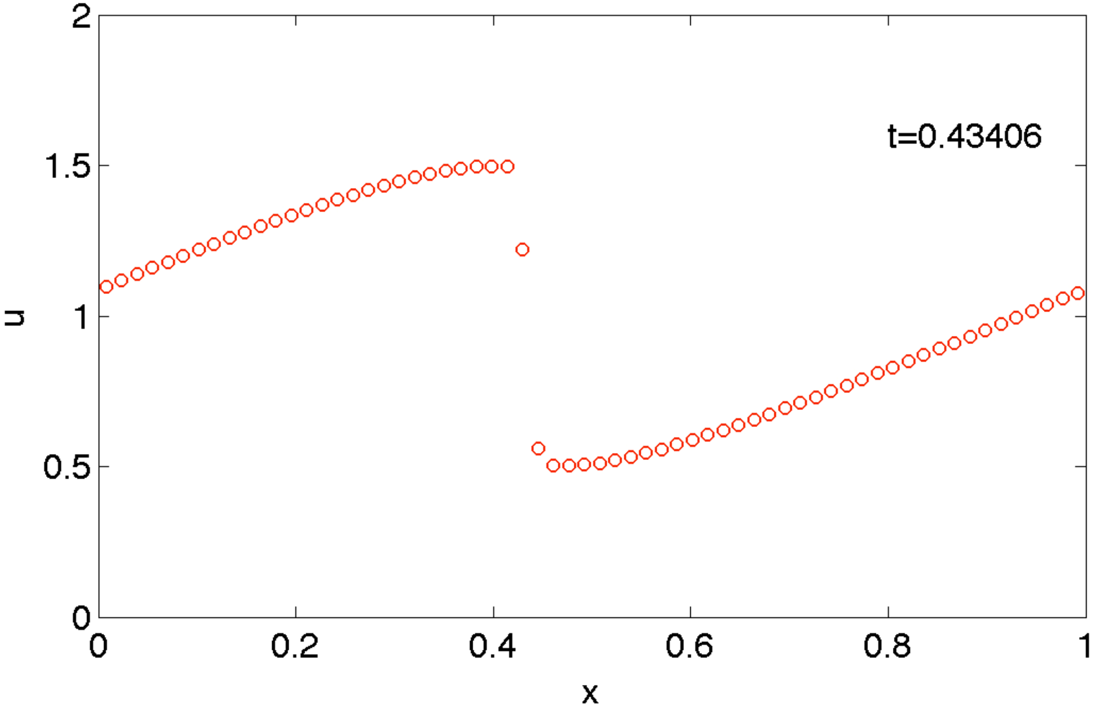
Solving non-linear equations

- Simplest example: Burger's Eqs $\partial_t u + u\partial_x u = 0$

In conservative form: $$\partial_t u + \partial_x\left(\frac{u^2}{2}\right) = 0$$

Initial condition: u=1-sin(2πx)/2 in [0, 1], periodic BC.

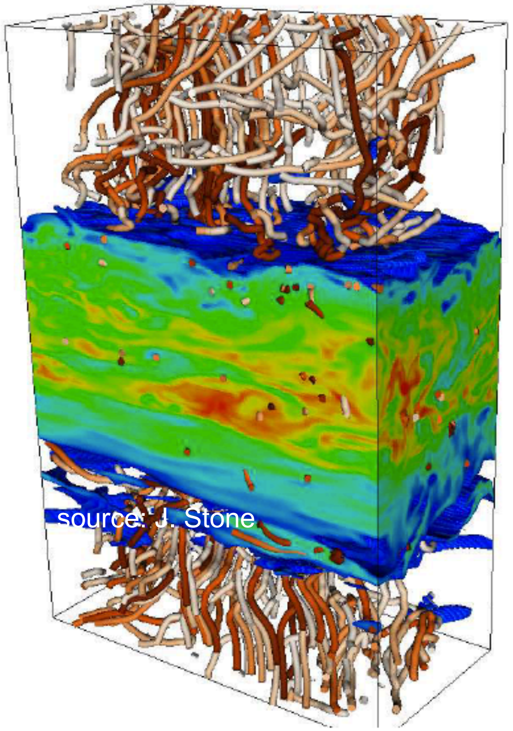
Solved with Godunov method + 2nd order reconstruction



Winter School on Computational Astrophysics, Shanghai, 2018/02/01

Computational Magnetohydrodynamics

(continued)


source: J. Stone

Xuening Bai (白雪宁)

Institute for Advanced Study (IASTU) &

Tsinghua Center for Astrophysics (THCA)

清華大學

Tsinghua University

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Non-linear system of equations

From a linear system of hyperbolic equations:

General Riemann problem can be solved by decomposing the jump into eigenvectors of the coefficient matrix, corresponding traveling waves with characteristic speeds.

From non-linear scalar problems:

Waves can deform into compression or expansion waves. Solution to the Riemann problem consists of a single shock or a rarefaction wave.

Now, we combine them into a general for non-linear systems of Eqs.

Like the linear case, for a system with m equations, the jump is split into m separate waves.

For each of these waves, like in the non-linear case, can now be either a shock or a rarefaction.

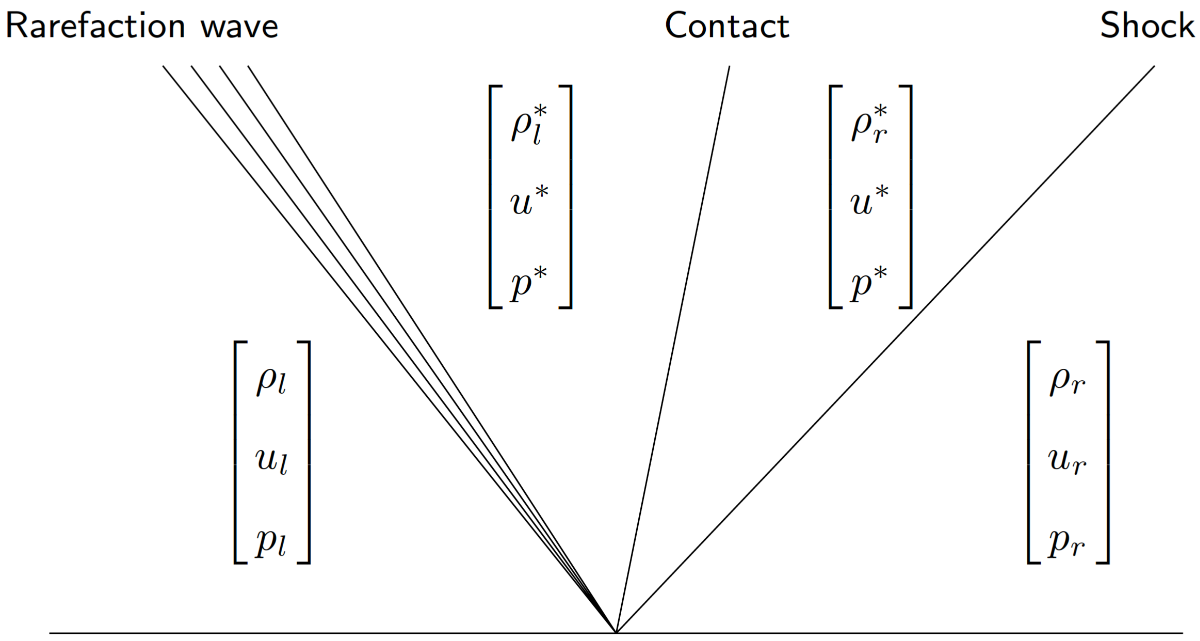
1D Hydro Equations

In conservative form (and ignore transverse velocities):

$$\begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{bmatrix}_x = 0.$$

3 waves: 2 sound waves
+ 1 entropy wave

A typical structure of the Riemann problem consists of:

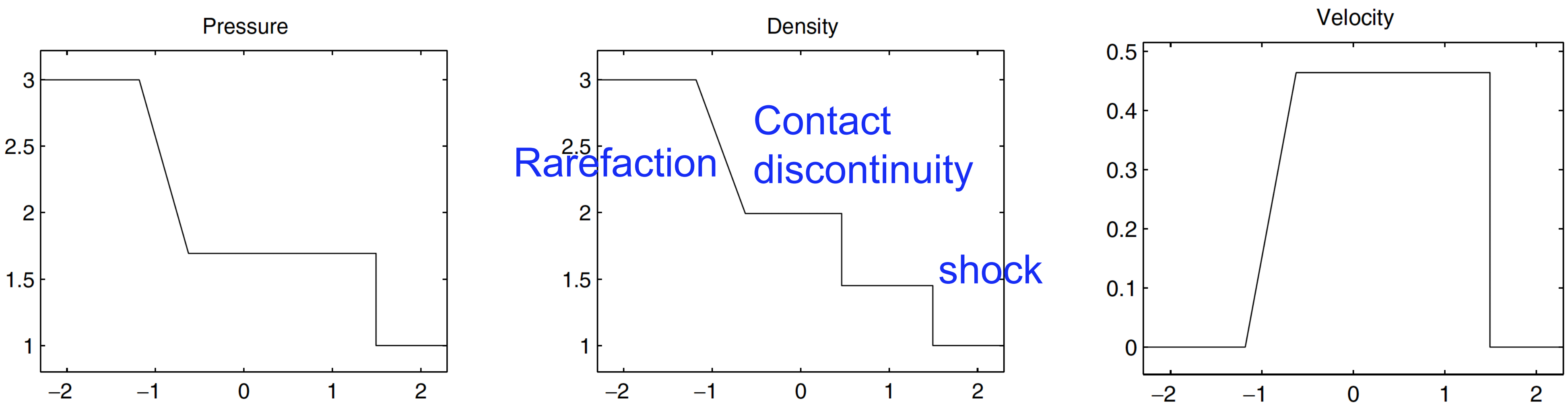

Rarefaction wave
Contact
Shock
$\begin{bmatrix} \rho_l^* \\ u^* \\ p^* \end{bmatrix}$
$\begin{bmatrix} \rho_r^* \\ u^* \\ p^* \end{bmatrix}$
$\begin{bmatrix} \rho_l \\ u_l \\ p_l \end{bmatrix}$
$\begin{bmatrix} \rho_r \\ u_r \\ p_r \end{bmatrix}$

Exact hydro Riemann solver is possible, but numerically very expensive.

Sod shock tube

Experimental shock tube: two different gas states separated by a membrane at $x=0$ => special Riemann problem with $u_l=u_r=0$.

Solution generally gives a shock + contact disc. + rarefaction:



For other initial conditions, it is possible to obtain two shocks or two rarefactions.

1D MHD Equations

1D equations are plane-symmetric: $\nabla \cdot \boldsymbol{B} = 0$ => $B_x = \mathrm{const}$

1D adiabatic MHD equations in conservative form: $\dfrac{\partial \boldsymbol{U}}{\partial t} + \dfrac{\partial \boldsymbol{F}}{\partial x} = 0$

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ M_x \\ M_y \\ M_z \\ E \\ B_y \\ B_z \end{bmatrix} \qquad \boldsymbol{F} = \begin{array}{cc} \left[\begin{array}{c} \rho v_x \\ \rho v_x^2 + P + B^2/2 - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ (E + P^*) v_x - (\boldsymbol{B} \cdot \boldsymbol{v}) B_x \\ B_y v_x - B_x v_y \\ B_z v_x - B_x v_z \end{array}\right] & \begin{array}{c} \\ \\ \\ \\ \\ E_z \\ -E_y \end{array} \end{array}$$

7 variables, 7 waves

The Eigensystem

The Jacobian matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}$ has 7 real eigenvalues, one for each wave:

2 fast magnetosonic waves: $\lambda_1 = v_x - c_f \qquad \lambda_7 = v_x + c_f$

2 Alfven waves: $\lambda_2 = v_x - v_{A,x} \qquad \lambda_6 = v_x + v_{A,x}$

2 slow magnetosonic waves: $\lambda_3 = v_x - c_s \qquad \lambda_5 = v_x + c_s$

1 entropy wave: $\lambda_4 = v_x$

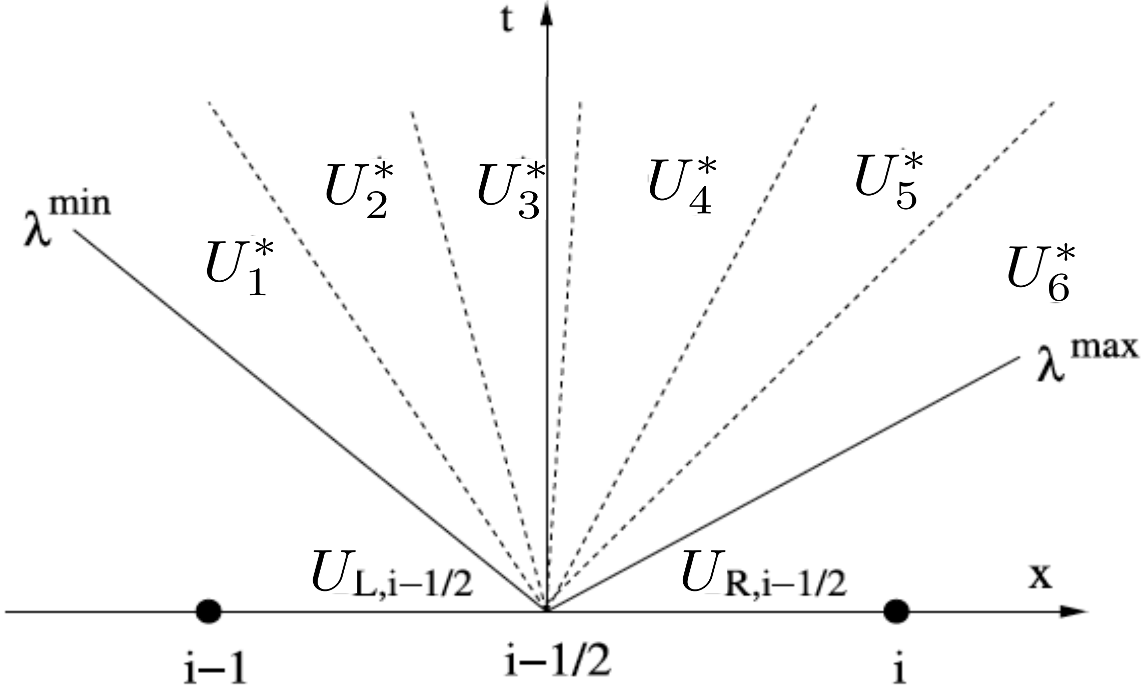
where $$c_{f,s}^2 = \frac{1}{2}\left[(a^2 + v_A^2) \pm \sqrt{(a^2 + v_A^2)^2 - 4a^2 v_{A,x}^2}\right]$$

$$v_{A,x} = \frac{B_x}{\sqrt{4\pi\rho}} \qquad v_A = \frac{B}{\sqrt{4\pi\rho}} \qquad a^2 = \gamma P/\rho$$

An “entropy wave” is a contact discontinuity.

For isothermal MHD, the number reduces to 6 (no entropy wave).

The MHD Riemann problem



Goal: find the intermediate state at x=$x_{i-1/2}$.

MHD Riemann solvers are much more complex, and in some cases, 2 of the 3 waves are degenerate (i.e., not strictly hyperbolic).

In practice, HD/MHD Godunov schemes use approximate Riemann solvers and/or linearized Riemann solvers.

The Roe's solver (Roe, 1981, Cargo & Gallice, 1997)

Replace the non-linear problem $\partial_t u + \partial_x f(u) = 0$

by some linear problem defined at each cell interface:

$$\partial_t u + \hat{A}_{i-1/2} \partial_x u = 0 \quad \text{with} \quad \begin{array}{l} u=u_L,\ (x<x_{i-1/2}) \\ u=u_R,\ (x\geq x_{i-1/2}) \end{array}$$

where $\hat{A}_{i-1/2}$ is some approximation to $f'(\hat{u})$ valid between u_L and u_R.

Roe's average: $\hat{u} = \dfrac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}$

Pros: Good resolution for all 7 waves, and hence is generally less diffusive and more accurate.

Cons: Requires characteristic decomposition in conserved variables (expensive).

Difficult to add new physics.

Fail at strong rarefactions.

The HLLE solver

(Harten, Lax & van Leer, 1983, Einfeldt et al. 1991)

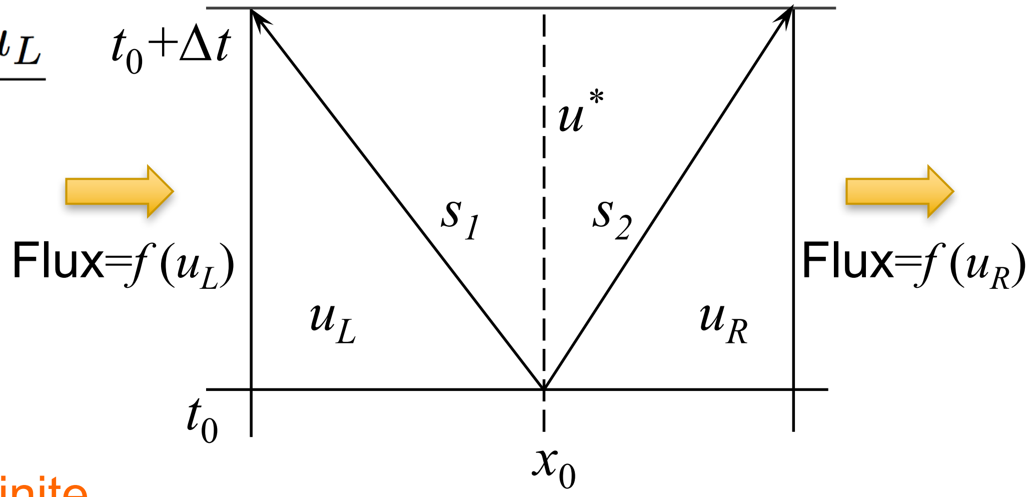
Estimate the largest/smallest wave speeds arising in the Riemann solution, assuming the solution has only these two waves (only 1 intermediate state).

Then we can solve the intermediate state from the conservation laws:

$$[f(u_L) - f(u_R)]\Delta t = (s_2 - s_1)u^*\Delta t - s_2 u_R \Delta t + s_1 u_L \Delta t$$

This gives the intermediate state as:

$$u^* = \frac{f(u_L) - f(u_R) + s_2 u_R - s_1 u_L}{s_2 - s_1}$$



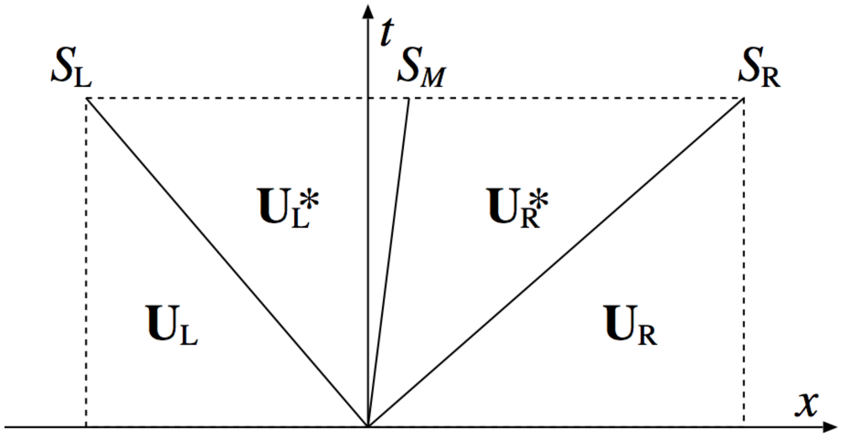
Pros: Very simple and efficient.
Intermediate state is positive definite.

Cons: Very diffusive, especially at contact discontinuities.

The HLLC/HLLD solver

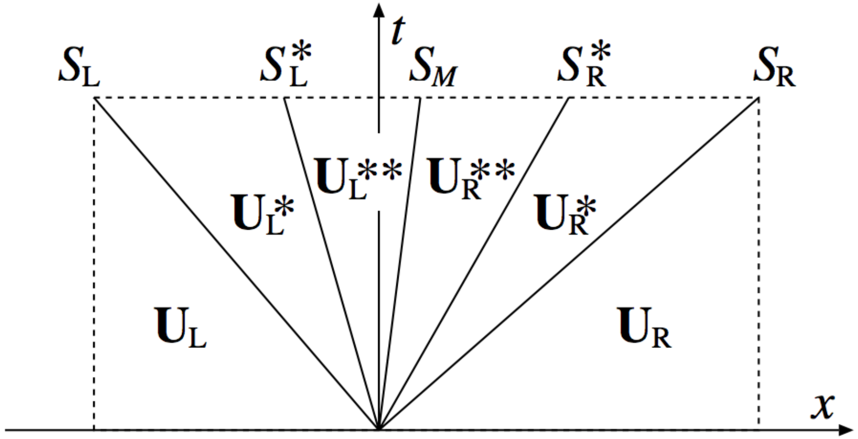
The HLLC solver (hydro):

contact discontinuity



The HLLD solver (MHD):

(all) discontinuity



(Miyoshi & Kasano, 2005)

5-wave Riemann solver with 4 intermediate states (for MHD): resolves fast, Alfven waves and the contact discontinuity.

Reasonably simple and efficient, guarantees positivity in 1D, better resolution at contact discontinuities.

Primitive vs. conserved variables

It is necessary to convert conserved variables ***U*** to primitive variables ***W*** in various stages of the computation.

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ M_x \\ M_y \\ M_z \\ E \\ B_x \\ B_y \\ B_z \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \rho \\ v_x \\ v_y \\ v_z \\ P \\ B_x \\ B_y \\ B_z \end{bmatrix},$$

Caveat: Due to the approximate nature of the Riemann solver, one might get negative density after one step of integration.

Similarly, with

$$E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho v^2 + \frac{B^2}{8\pi}$$

one might obtain negative pressure following conversion from conserved to primitive variables.

These issues can be more severe in relativistic MHD.

Solution:

1). Add density/pressure floors.

2). Use a more diffusive solver.

MHD integrator

- Godunov’s original method (1st order)

Step 1: Donor-cell reconstruction to obtain interface L/R states.

Step 2: Use an MHD Riemann solver to compute 1st order fluxes.

Step 3: Update the system for a full time step using 1st order fluxes.

Robust, but very diffusive.

MHD integrator

- Second-order accuracy can be achieved using predictor-corrector type method (with a number of varieties).

Step 1: Donor-cell reconstruction to obtain interface L/R states.

Step 2: Use a Riemann solver to compute 1st order fluxes F^n.

Step 3: Advance the system for ½ time step (predict step).

$$U_i^{n+1/2} = U_i^n - \frac{\Delta t}{2\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n)$$

Step 4: Use the second-order (piecewise-linear) reconstruction to compute the L/R states from $U^{n+1/2}$.

Step 5: Use a Riemann solver to compute 2nd order fluxes $F^{n+1/2}$.

Step 6: Update the system for a full time step.

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2})$$

This is one algorithm adopted in Athena++, following Falle (1991), modified from the MUSCL-Hancock (“van Leer”) schemes.

Multi-dimension MHD

- MHD equations in conservative form in 3D:

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}+\frac{\partial \boldsymbol{G}}{\partial y}+\frac{\partial \boldsymbol{H}}{\partial z}=0,$$

- Traditionally, multi-D methods are constructed using directional spliting (mainly for hydrodynamics):

1.Solve $U_t=F_x$ as in 1D MHD.

2.Solve $U_t=G_y$, with G constructed from result of the x-update.

3.Solve $U_t=H_z$, with H constructed from result of the y-update.

Pros: easy to implement.

Cons: symmetry is not preserved, incompatible with constrained transport.

Directionally unsplit approach

- MHD equations in conservative form in 3D:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} + \frac{\partial \boldsymbol{H}}{\partial z} = 0,$$

- Directionally unsplit method:

Solve $U_t=F_x$, $U_t=G_y$, $U_t=H_z$ simultaneously as in 1D MHD.

In other words, in every MHD (sub)timestep:

1. Use a reconstruction method to compute the L/R states at all interfaces.
2. Use a Riemann solver to compute the fluxes across all faces.
3. Construct EMFs at all cell edges (see in the next few slides).
4. Advance the solution by taking flux divergences and applying CT.

The Courant-Friedrichs-Lewy (CFL) condition

- Numerical timestep Δt must be sufficiently small so that information propagates no more than one grid point per timestep.

For MHD equations, taking the fastest speed as:

$$V_{x,y,z} = \max[\mathrm{abs}(v_{x,y,z} \pm v_{f,\{x,y,z\}})]$$

where *v* is flow speed, v_f is the fast magnetosonic speed, in each direction.

The timestep is given by:

taken across the entire mesh

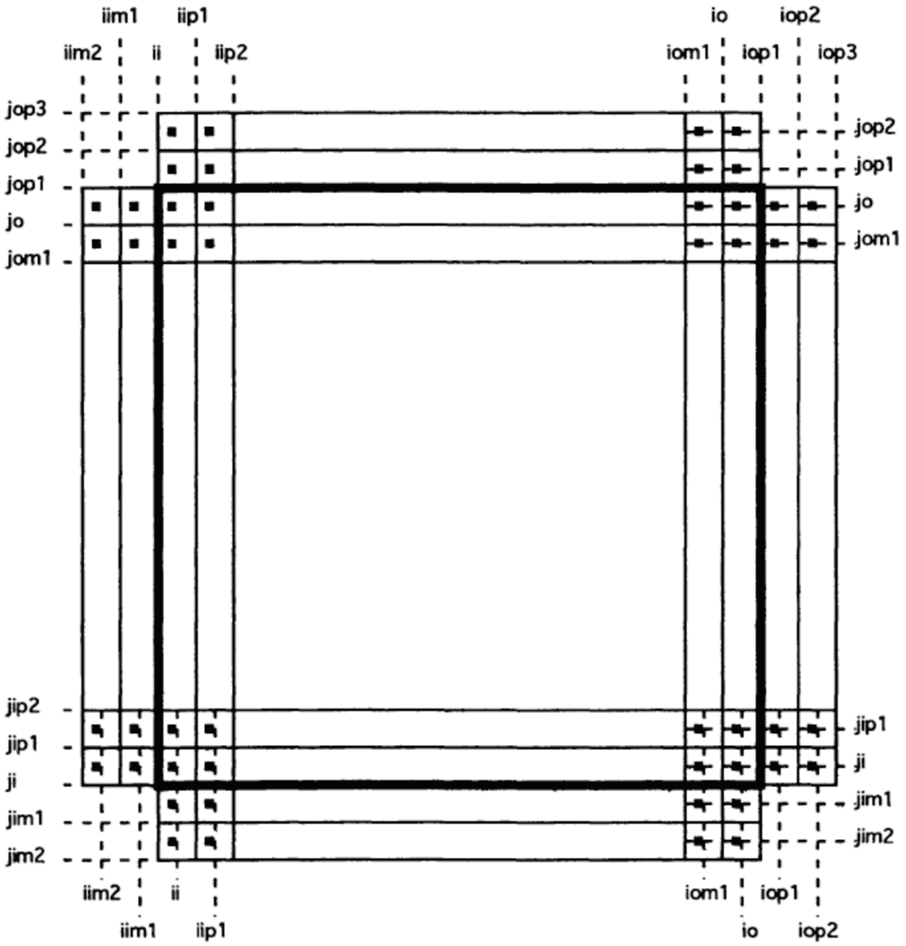
$$\Delta t = \mathrm{CFL} \cdot \min\left(\frac{\Delta x}{V_x}, \frac{\Delta y}{V_y}, \frac{\Delta z}{V_z}\right)$$

Different MHD integrators may have different requirement on the CFL number, which can also depend on the dimension of the problem.

e.g., the van Leer integrator in Athena++ requires the CFL # to < 1 for 1D and <0.5 for 2D/3D problems.

Boundary conditions (BCs)

Boundary conditions are implemented by properly filling the ghost zones.



Needed only to help determine the L/R states at the boundaries (and reconstruct EMFs for CT).

# of ghost zones depends on the order of reconstruction and numerical implementation (2 in Athena++).

Periodic and reflecting/conducting BCs are straightforward.

“zeroth-order extrapolation” generally gives best results serving for outflow/absorbing BCs

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Importance to preserve divergence of B

- For multi-dimensions numerical schemes, there is no guarantee that divergence of B is kept zero, due to truncation error.

Consequence:

spurious parallel acceleration

$$\boldsymbol{J} \times \boldsymbol{B} = -\nabla \cdot \left(\frac{B^2}{8\pi}\mathbf{I} - \frac{\boldsymbol{B}B}{4\pi} \right) - \frac{(\nabla \cdot \boldsymbol{B})\boldsymbol{B}}{4\pi}$$

Divergence error can accumulate, leading to inconsistent results over long term.

In some cases, it can lead to numerical instabilities and make the code crash…

Techniques to preserve divergence of B

- Divergence cleaning:

Powell’s 8-wave scheme (Powell, 1999):

Add source terms to momentum/induction equations to advect magnetic monopoles away. But: can give the wrong shock jump conditions.

Projection method (Brackbil & Bams, 1980):

Solve a Poisson equation for the “magnetic charge”: $\Delta\Phi = \nabla \cdot \boldsymbol{B}$

Then clean the divergence field: $\boldsymbol{B} \rightarrow \boldsymbol{B} - \nabla\Phi$

But: very expensive to solve elliptic PDE, and may smooth discontinuities in B.

Dedner’s scheme (Dedner et al. 2002): introducing a general Lagrangian multiplier, transporting div(B) errors away. Reasonably robust in most cases.

- Use vector potential (usually used in finite-difference codes, e.g., Pencil)

$$\frac{\partial \boldsymbol{A}}{\partial t} = \boldsymbol{v} \times \boldsymbol{B} \ , \quad \boldsymbol{B} = \nabla \times \boldsymbol{A}$$

Div(B)=0 by construction, but need hyper-resistivity for stabilization.

Constrained transport (CT)

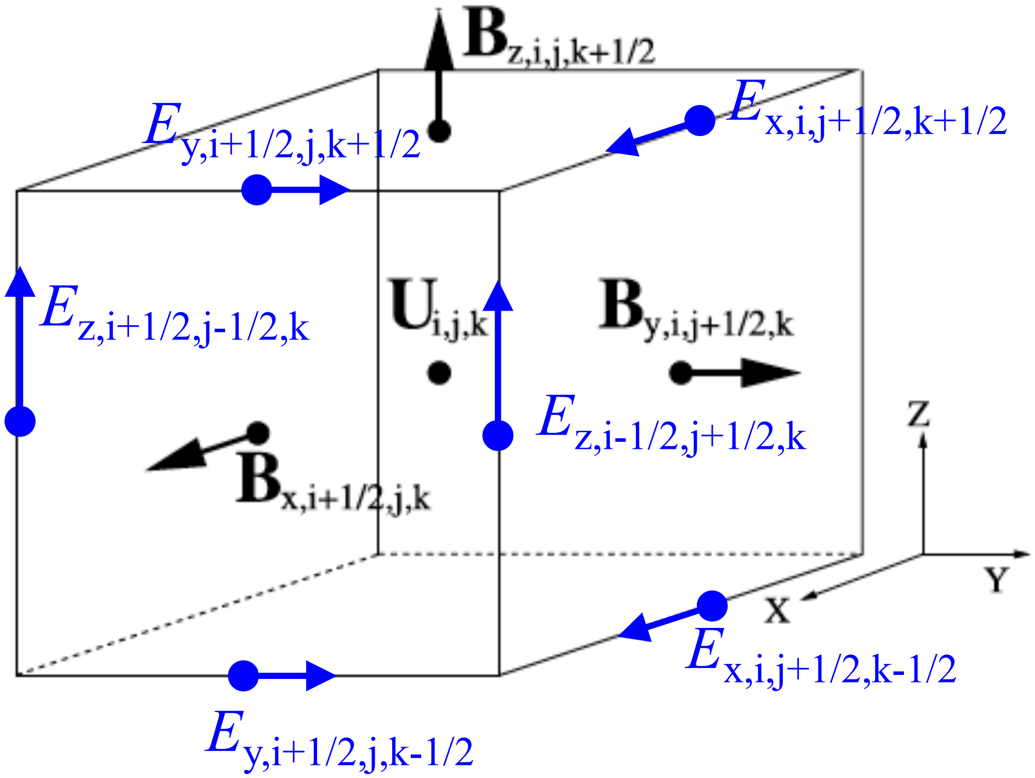
(Evans & Hawley, 1988)

Magnetic fields defined at face-center, area-averaged:

$$(B_x)_{i+1/2,j,k} = \frac{1}{\Delta y \Delta z} \int_S B_x(y,z)\, dydz$$

Electromotive forces (vxB) defined at edges, line-averaged:

$$(E_x)_{i,j+1/2,k-1/2} = \frac{1}{\Delta x \Delta t} \int E_x(x)\, dxdt$$



Evolve magnetic field via Stoke's law:

$$\frac{\partial}{\partial t} \int_S \boldsymbol{B} \cdot d\boldsymbol{S} = -\int_L \boldsymbol{E} \cdot d\boldsymbol{l}$$

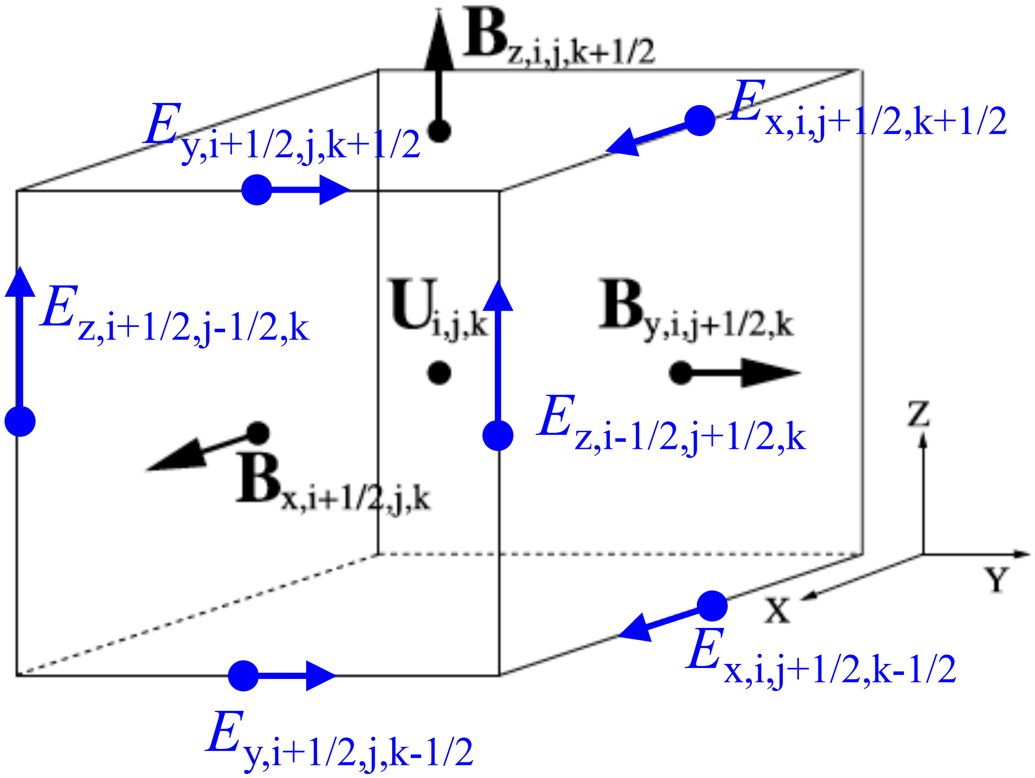
$$B^{n+1}_{x,i+1/2,j,k} = B^{n}_{x,i+1/2,j,k} - \frac{\Delta t}{\Delta y}(E^{n+1/2}_{z,i-1/2,j+1/2,k} - E^{n+1/2}_{z,i-1/2,j-1/2,k}) + \frac{\Delta t}{\Delta z}(E^{n+1/2}_{y,i-1/2,j,k+1/2} - E^{n+1/2}_{z,i-1/2,j,k-1/2})$$

These equations are exact: no approximations.

Constrained transport (CT)

Div (B)=0 is preserved to machine accuracy:

$$\nabla \cdot \boldsymbol{B} = \frac{B_{x,i+1/2,j,k} - B_{x,i-1/2,j,k}}{\Delta x} + \frac{B_{y,i,j+1/2,k} - B_{x,i,j-1/2,k}}{\Delta y} + \frac{B_{z,i,j,k+1/2} - B_{z,i,j,k-1/2}}{\Delta z}$$



Updates in Div(B) corresponds to differences in the EMFs that cancel exactly.

Main challenge: construct electric fields at cell edges (3D) or corners (2D).

By arithmetic averaging the EMFs returned from the Riemann solvers (at face centers), the EMFs are not properly upwinded.

Need to reconstruct the EMF at the corners (Gardiner & Stone, 2005).

Other methods for computational MHD

- Finite-difference method

  Can achieve very high order accuracy for smooth flows, but requires artificial and hyper viscosity/resistivity to stabilize the code. Poor performance at strong shocks.

  Example: Pencil code

- Spectral method

  Usually for incompressible/anelastic flow (filter out sound waves). Convergence is exponential. Main application: atmosphere, stellar interior, (occasionally) accretion disks.

  Example: Snoopy, Dedalus

- Smoothed particle (magneto-)hydrodynamics

  Mesh-free Lagrangian method, mostly for hydrodynamic applications, but recent development include magnetic fields with divergence cleaning. Very flexible to handle flows with large dynamical range, but has issues dealing with shocks and turbulence.

  Example: Phantom code

- Moving-mesh/meshless MHD

  Computation based on unstructured Lagrangian points. Partition the volume and use Riemann solvers (fully conservative). Implementing CT is possible but very difficult, mostly use divergence cleaning. Reduced advection error but enhanced grid noise.

  Example: Arepo, Gizmo

Outline

- Hyperbolic PDEs and conservation laws
- Nonlinear problems and discontinuities
- Solving the linear advection equation
- Finite volume methods (for scalar conservation law)
- Godunov method for solving MHD equations
- Preserving divergence-free B field
- Adding source terms

Add more physics

- Depending on the problem, adding more physics can require just small changes, or a complete rewrite of the algorithm.

1). Simple changes:

Adding local source terms (e.g., cooling, thermal relaxation).

2). Modest changes:

Adding flux-divergence terms (e.g., viscosity, resistivity).

Add terms requiring elliptic solvers (e.g., self-gravity).

3). Complete re-write:

Adding new dynamical equations (e.g., special/general relativity, particles, radiation).

Add more physics

- Simple source terms (cases 1, 2) are usually added via operator splitting:

For equation: $$\frac{\partial U}{\partial t} + \nabla \cdot \boldsymbol{F} = S$$ or $$\frac{\partial U}{\partial t} + \nabla \cdot \boldsymbol{F} = \nabla \cdot \boldsymbol{F}_s$$

Solve it by sequentially solving two separate equations:

$$\frac{\partial U}{\partial t} + \nabla \cdot \boldsymbol{F} = 0$$ and $$\frac{\partial U}{\partial t} = S$$

Formally, operate splitting makes the scheme first order in time.

Higher-order accuracy can be achieved using multi-step methods (e.g., RK), and can be embedded to the MHD integrator.

- New dynamical equations (case 3) are solved separately, which then supply source terms to the MHD equations. They can be handled either by operator splitting or multi-step methods.

Example: optically thin cooling

Add source terms to the energy equation:

$$\frac{\partial E}{\partial t} = \underbrace{-\rho^2 \Lambda(T)}_{\text{cooling rate}} + \underbrace{\rho H}_{\text{heating rate}}$$

Cooling terms can be added directly to the integrator in Godunov methods.

The cooling rate can be stiff: explicit methods require very small Δt.

Solution: use implicit method

$$\rho c_V \frac{T^{n+1} - T^n}{\Delta t} = -\rho^2 \Lambda(T^{n+1})$$

which can be implemented locally once the cooling function is known.

Diffusive source terms

Consider the prototype of a diffusion equation: $\partial_t u = D\partial_{xx} u$

Simplest approach: FTCS

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = D u_{xx}^n = D\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}\right)$$

von Neumann stability analysis:

Setting $u_i^n = a e^{\mathrm{i}k(i\Delta x)}$ and $u_i^{n+1} = A_k u_i^n$, we find

$$A_k = 1 + \frac{D\Delta t}{\Delta x^2}[2\cos(k\Delta x) - 2] = 1 - \frac{4D\Delta t}{\Delta x^2}\sin^2\left(\frac{k\Delta x}{2}\right)$$

➡ $\Delta t \leq \dfrac{\Delta x^2}{2D}$ for stability.

Diffusive source terms

In multi-D, von Neumann stability analysis:

$$A_{\boldsymbol{k}} = 1 - \frac{4D\Delta t}{\Delta x^2}\sin^2\left(\frac{k_x \Delta x}{2}\right) - \frac{4D\Delta t}{\Delta y^2}\sin^2\left(\frac{k_y \Delta y}{2}\right) - \frac{4D\Delta t}{\Delta z^2}\sin^2\left(\frac{k_z \Delta z}{2}\right)$$

⇨ $\Delta t \leq \dfrac{\Delta x^2_{\mathrm{min}}}{2ND}$ for stability (N: # of dimensions).

The fact that $\Delta t \propto \Delta x^2$ places severe constraints on the timestepping.

What to do?

1. Do nothing. This is physics, you can not avoid.

2. Sub-cycling or super timestepping.

   Multiple diffusion steps per MHD step. Can accelerate by a factor of several.

3. Use implicit method.

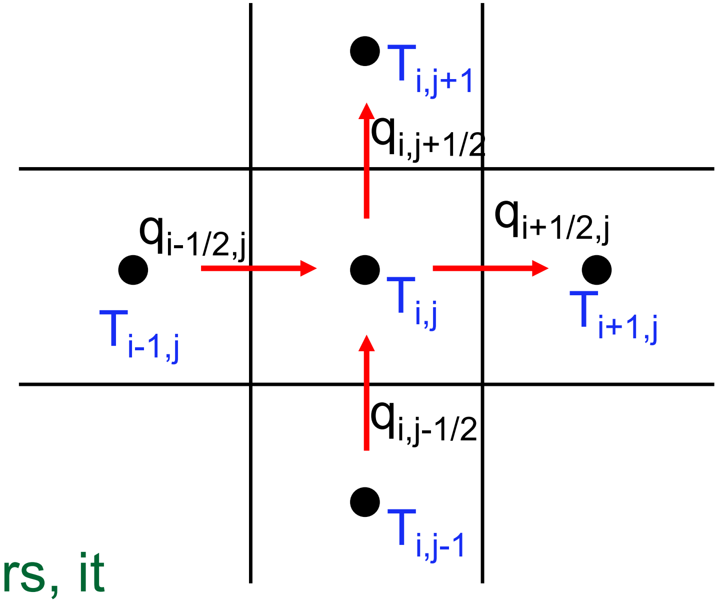
   Eliminate the timestep constraint, at the cost of needing global communication.

Example: thermal conduction

Heat flux: $$\boldsymbol{q} = -\kappa \nabla T$$

Energy update: $$\frac{\partial \epsilon}{\partial t} = -\nabla \cdot \boldsymbol{q} = \nabla \cdot (\kappa \nabla T)$$

Thermal conduction should be implemented in the framework of energy conservation.



Since temperature is defined at cell centers, it is natural to define heat flux at cell interfaces, and update the energy by directly differencing the heat flux.

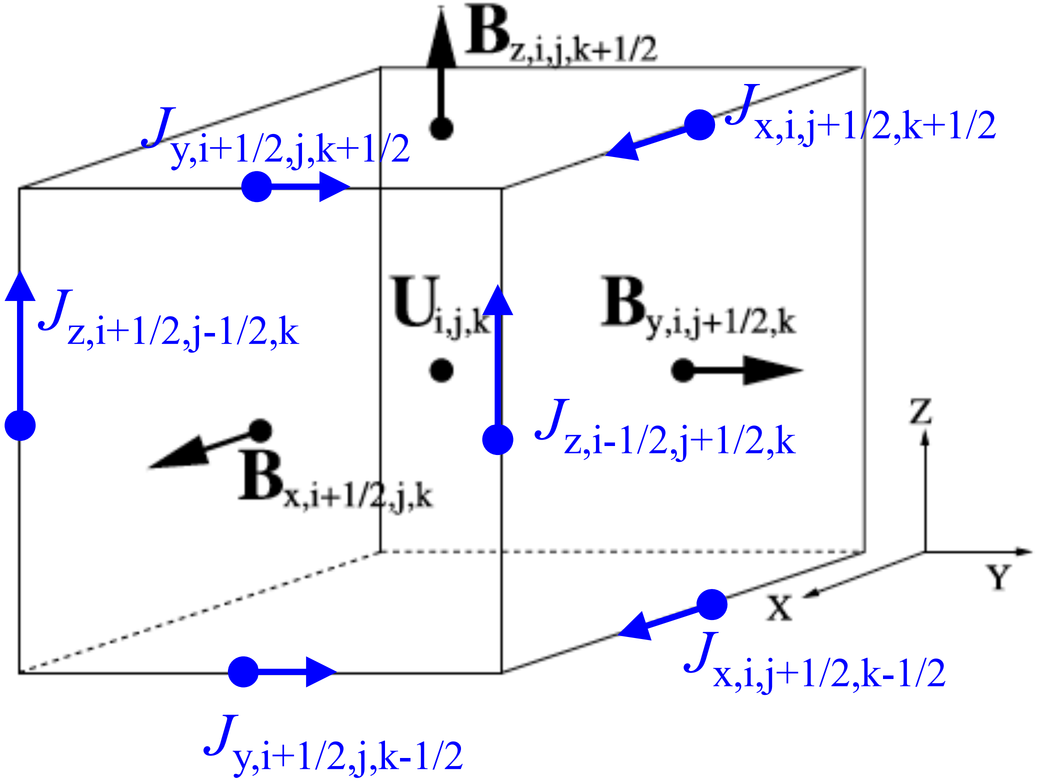
Example: resistivity

Resistive term:

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times (\eta \boldsymbol{J})$$

The overriding concern to keep div(B)=0 suggests CT differencing:

Define $\boldsymbol{J}$ at cell edges => resistive EMF = $\eta\boldsymbol{J}$.



J can be obtained by taking the curl face-centered B field.

Additionally, if energy eq. is included, one needs to further compute the Poynting flux at cell interfaces (by interpolation).

MHD integrator with source terms

Steps in a predictor-corrector type (van Leer) MHD integrator with source terms:

Step 1: Donor-cell reconstruction from U^n to obtain interface L/R states.

Step 2: Use a Riemann solver to compute 1st order fluxes F^n.

Step 3: Calculate EMFs at cell edges based on F^n and U^n.

Step 4: Calculate new physics source terms.

Step 5: Advance the system for ½ time step (predict step), including new physics source terms.

Step 6: Use the second-order (piecewise-linear) reconstruction to compute the L/R states from $U^{n+1/2}$.

Step 7: Use a Riemann solver to compute 2nd order fluxes $F^{n+1/2}$.

Step 8: Compute new EMFs at cell edges based on $F^{n+1/2}$ and $U^{n+1/2}$.

Step 9: Calculate new physics source terms.

Step 10: Update the system for a full time step, including new physics source terms.

Summary

- Computational MHD is an important tool to study a wide range of astrophysical plasma phenomena.
- MHD equations are hyperbolic conservation laws
- Godunov method: fully conservative

  Main idea: reconstruct-evolve-average, with proper upwinding.

  Shock capturing using Riemann solvers.
- Preserving div(B)=0 is crucial:

  Use divergence cleaning or constrained transport.
- Source terms with different complexities can be added, exercising care in preserving the conservation law, and issues with timestepping and stability.