Computational Astrophysics

ASTR 660, Spring 2021
計算天文物理

Lecture 9

PDE: Hyperbolic problems

Instructor: Prof. Kuo-Chuan Pan
kuochuan.pan@gapp.nthu.edu.tw

Class website

https://kuochuanpan.github.io/courses/109ASTR660_CA/

Plan for today

- Partial Differential Equations
- Hyperbolic PDEs
- Lab: Advecting Fluids

Partial Differential Equations

Partial Differential Equations

- Partial Differential Equations (PDEs) involve partial derivatives with respect to more than one independent variable
- Independent variables typically include
  - One or more space dimensions
  - Possibly time dimension as well
- More dimensions complicate problem formulation: we can have:
  - Pure initial value problem (IVP)
  - Pure boundary value problem (BVP)
  - Or mixture of both
- Equation and boundary data may be defined over irregular domain

Continuous Phenomena in nature

- Many of the basic laws of nature are expressed as PDEs, including
  - Maxwell's equations (EM fields)
  - Navier-Stokes equations (fluid)
  - Elasticity equations (vibrations in solid states)
  - Schrodinger's equations (quantum behaviors)
  - Einstein's equation of GR (space-time evolution)
  - …(more)

Order of PDE

- Order of PDE is order of highest-order partial derivative appearing in equation
- For example, advection equation is first order
- Important second-order PDEs include
  - Heat equation: $u_t = u_{xx}$
  - Wave equation: $u_{tt} = u_{xx}$
  - Laplace equation: $u_{xx} + u_{yy} = 0$

Classification of PDEs

- Second-order linear PDEs of general form

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0$$

- $b^2 - 4ac > 0$ Hyperbolic (e.g.wave, hydro eqs)
- $b^2 - 4ac = 0$ Parabolic (e.g.heat equation)
- $b^2 - 4ac < 0$ Elliptic (e.g. Laplace, Poisson eqs.)

Classification of PDEs

- Classification of more general PDEs is not so clean and simple, but roughly speaking
  - Hyperbolic PDEs describe time-dependent, conservative physical processes, such as convection, that are not evolving toward steady state (hydrodynamics)
  - Parabolic PDEs describe time-dependent, dissipative physical process, such as diffusion, that are evolving toward steady state (diffusion)
  - Elliptic PDEs describe process that have already reached steady state, and hence are time-independent (gravity)

Single PDEs

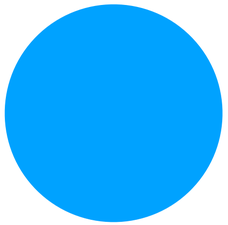
- For simplicity, we start from single PDEs (as opposed to systems of several PDEs) with only two independent variables, either:
  - Two space variables (x and y) or
  - One space variable (x) and one time variable (t)
- Partial derivatives with respect to independent variables are denoted by subscripts, for example
  - $u_t = \partial u / \partial t$
  - $u_{xy} = \partial^2 u / \partial x \partial y$

Example: Advection Equation

- Advection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\boldsymbol{c}) = 0$$

$$u_t = -cu_x \qquad \text{1D}$$

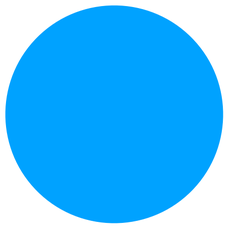


Example: Advection Equation

- Advection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\boldsymbol{c}) = 0$$

$$u_t = -cu_x \qquad \text{1D}$$



Example: Advection Equation

- Advection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\boldsymbol{c}) = 0 \qquad\qquad u_t = -cu_x$$

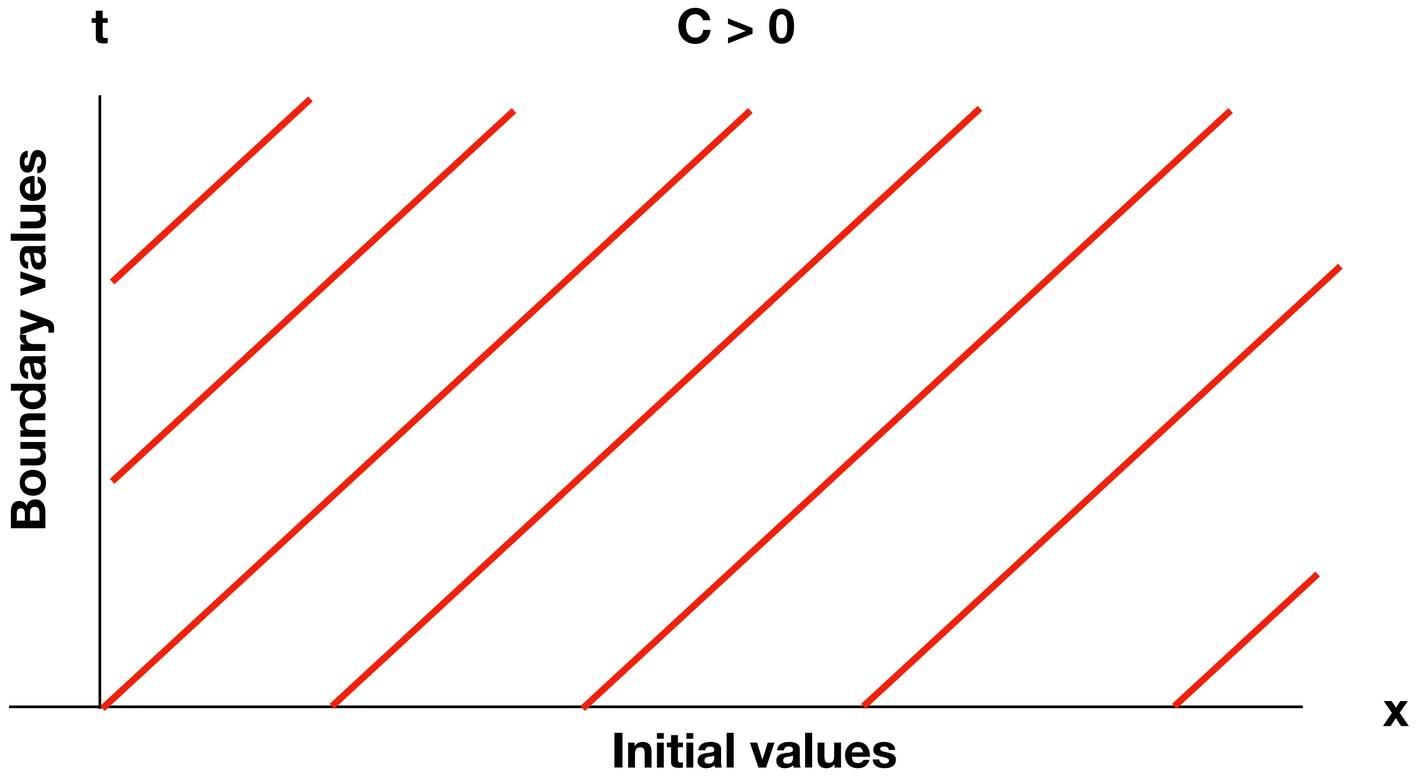
- Unique solution is determined by initial condition

$$u(0, x) = u_0(x),$$

- Seek solution u(t,x) for t >= 0
- Solution is initial function u_0 shifted by c t to right (if c > 0)

Characteristics

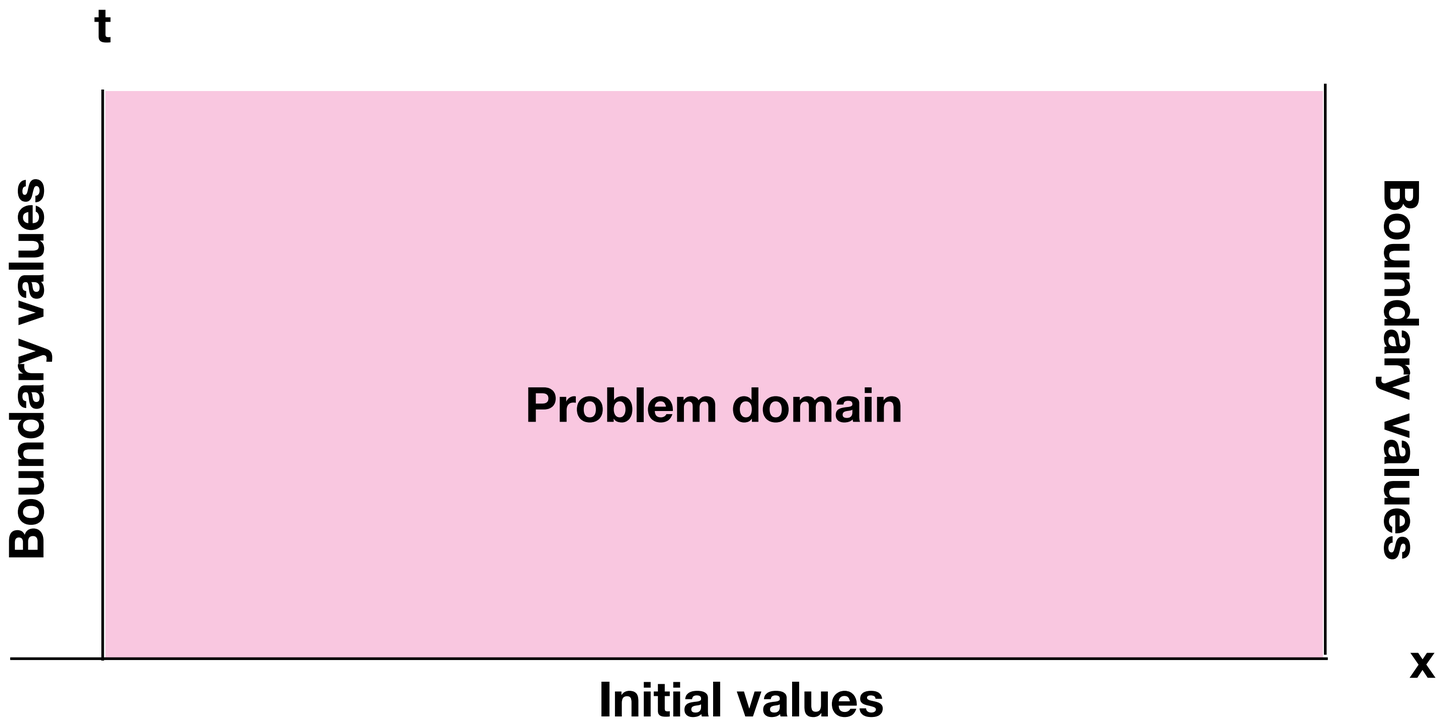
- Characteristics for PDE are level curves of solution
- For advection equation



Hyperbolic PDEs: Semi-Discrete Methods

Hyperbolic PDEs

- Hyperbolic (Time-dependent) PDEs usually involve both initial values and boundary values



Hyperbolic PDEs

- Hyperbolic (Time-dependent) PDEs usually involve both initial values and boundary values
- One way to solve time-dependent PDE numerically is to discretize in space but leave time variable continuous
- Result is systems of ODEs that can be solved by methods previously discussed
- For example, consider heat equation

$$u_t = u_{xx}$$

Initial condition: $u(0, x) = f(x) \qquad 0 \leq x \leq 1$

Boundary condition: $u(t, 0) = 0, u(t, 1) = 0, t \geq 0$

Semi-discrete Methods

$$u_t = u_{xx}$$

Initial condition: $u(0,x) = f(x) \qquad 0 \le x \le 1$

Boundary condition: $u(t,0) = 0, u(t,1) = 0, t \ge 0$

- Define spatial mesh points $x_i = i\ dx$
- Replace u_{xx} by finite difference approximation

$$u_{xx}(t,x_i) \sim \frac{u(t,x_{i+1}) - 2u(t,x_i) + u(t,x_{i-1})}{(\Delta x)^2}$$

Semi-discrete Methods

$$u_t = u_{xx}$$

Initial condition: $u(0,x) = f(x) \qquad 0 \le x \le 1$

Boundary condition: $u(t,0) = 0, u(t,1) = 0, t \ge 0$

- Rewrite in matrix form

$$\boldsymbol{y}' = \frac{c}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & .... & 0 & 1 & -2 \end{bmatrix} \boldsymbol{y} = \boldsymbol{A}\boldsymbol{y}$$

Hyperbolic PDEs: Fully-Discrete Methods

Fully-discrete Methods

- Fully discrete methods for time-dependent PDEs discretize in both time and space dimensions
- We could
  - replace continuous domain of equation by discrete mesh points
  - replace all derivatives in PDE by finite difference approximations
  - Seek numerical solution as table of approximate values at selected points in space and time

Fully-discrete Methods

- Resulting approximate solution values represent points on solution surface over problem domain in space-time plane.
- Accuracy of approximate solution depends on step sizes in both space and time
- Discrete system may be linear or nonlinear, depending on underlying PDE

Fully-discrete Methods

- Solution is obtained by starting with initial values along boundary of problem domain and marching forward in time step by step.
- Time-stepping procedure may be explicit or implicit, depending on whether formula for solution values at next time step involves only past information
- We might expect to obtain arbitrarily good accuracy by taking sufficiently small step sizes in time and space

Example: Advection Equation

$$u_t = -cu_x$$

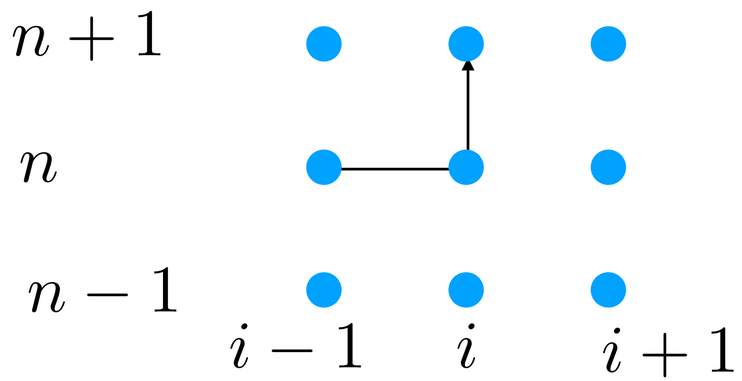
- Rewrite to

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\frac{u_i^n - u_{i-1}^n}{\Delta x}$$

or

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

Stencil



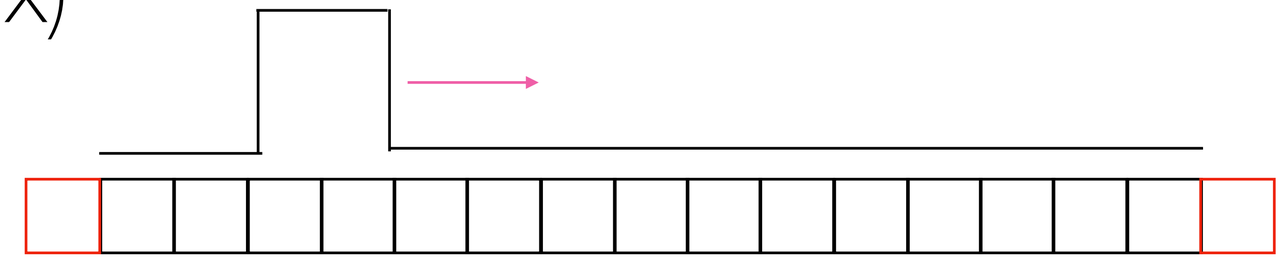
an explicit scheme

Example: Advection Equation

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

- Initial condition u(0,x) = f(x)

u (0,x)



- Boundary condition: periodic B.C.

See `1_fdm`

CFL Condition

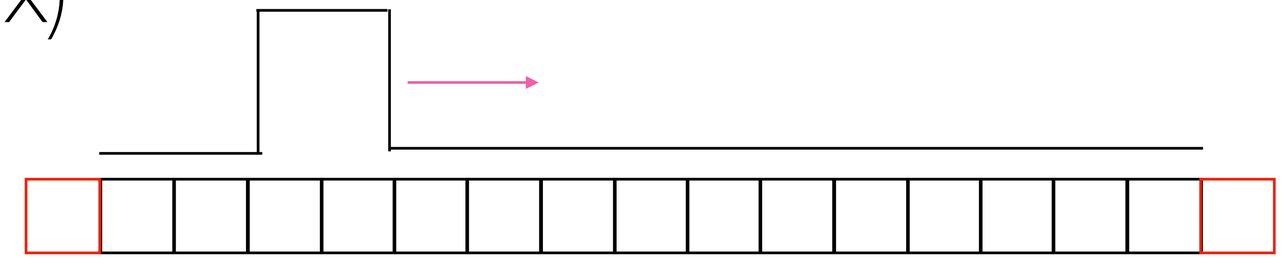
- Courant, Friedrichs, and Lewy (1928)
- Domain of dependence of PDE is portion of problem domain that influences solution at given point, which depends on characteristics of PDE
- Domain of dependence of difference scheme is set of all other mesh point that affect approximate solution at given mesh point
- CFL condition: necessary condition for explicit finite difference shame for hyperbolic PDE to be stable.

Example: Advection Equation

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

- Initial condition u(0,x) = f(x)

u (0,x)

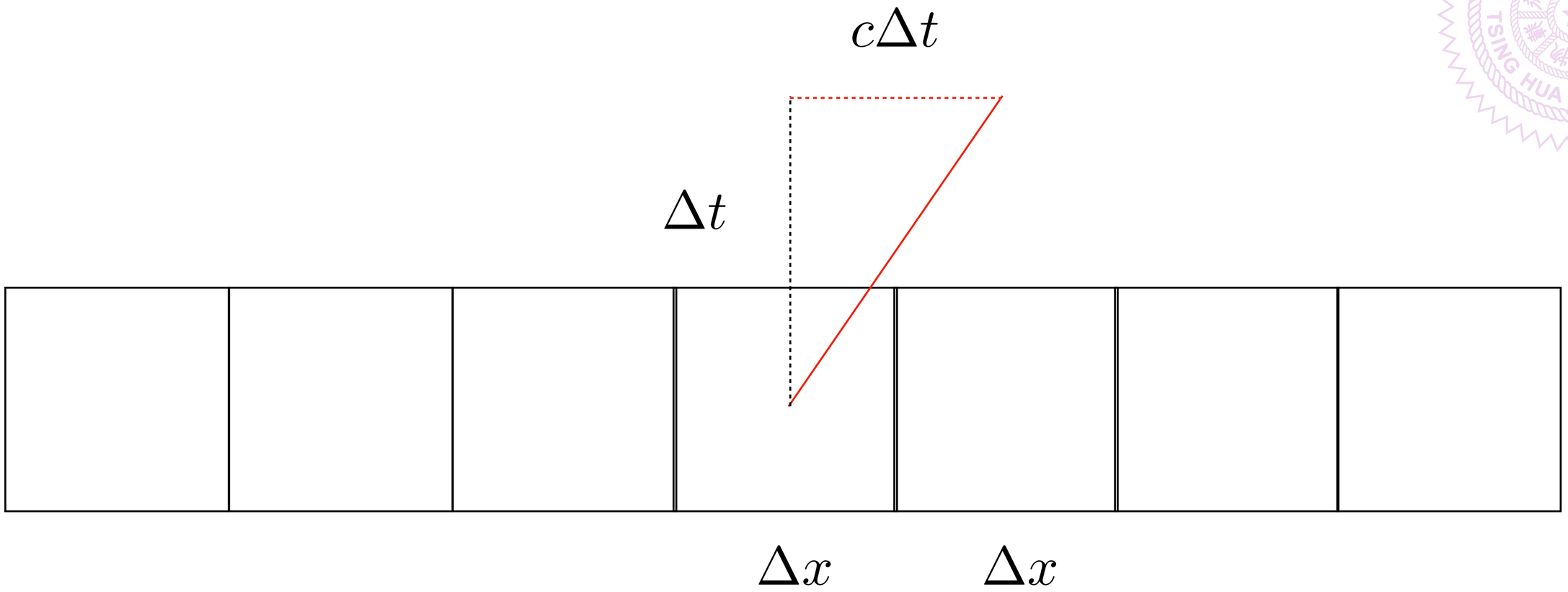


$$\Delta t \leq \mathrm{CFL}\frac{\Delta x}{c}$$

- Boundary condition: periodic B.C.

See `1_fdm`

Example: Advection Equation



$$\Delta t \leq \mathrm{CFL}\frac{\Delta x}{c}$$

See `1_fdm`

Example: Advection Equation

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

$$\Delta t \leq \mathrm{CFL}\frac{\Delta x}{c}$$

- Boundary condition: periodic B.C.
- Initial condition: (assume c=1)

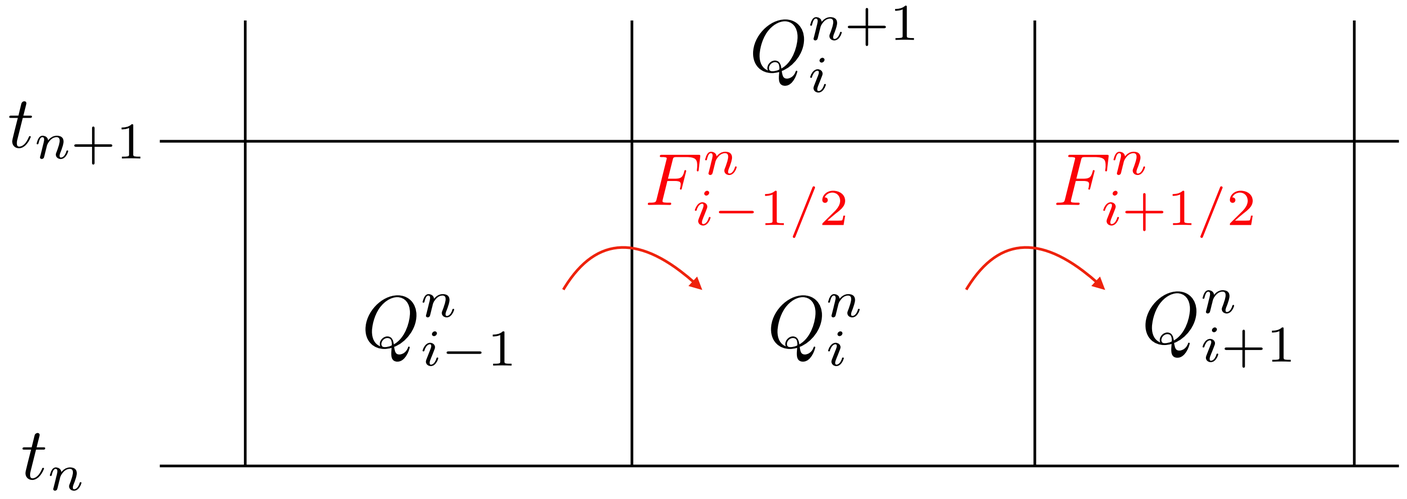
$$u(0,x) = \begin{cases} 1 & \text{if } 0.1 \leq x \leq 0.2 \\ 0.01 & \text{otherwise} \end{cases}$$

See `1_fdm`

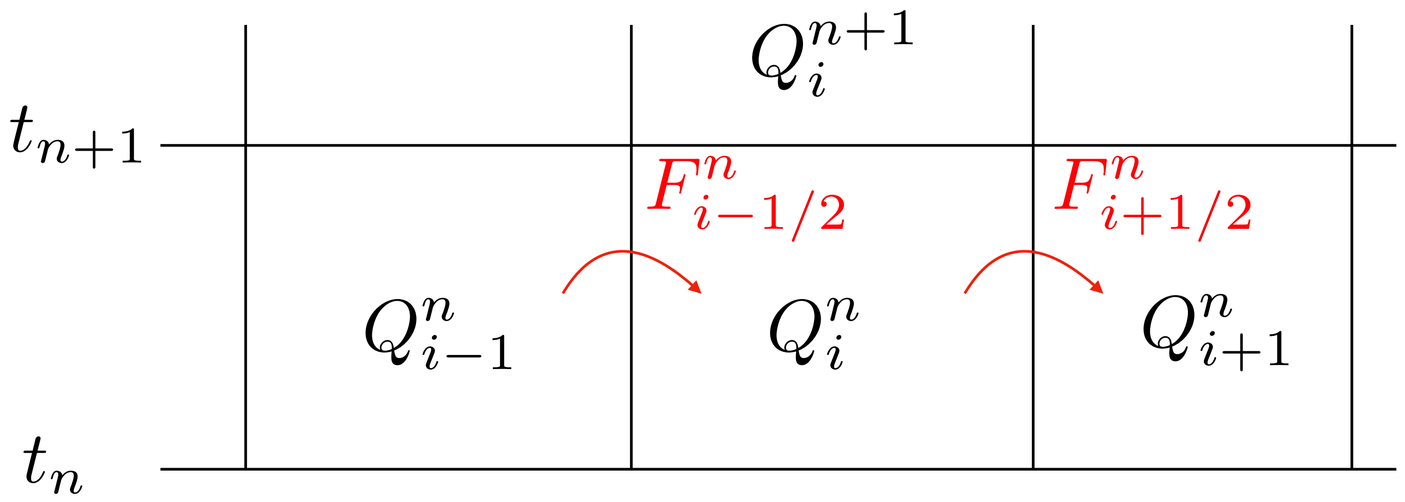
Hyperbolic PDEs: Finite Volume Methods

Finite Volume Methods

- Mesh points become “finite volumes” (grid cells)
- U(t,x) become cell averaged values instead of cell centered values
- Update cell values by evaluating fluxes at cell edges



Finite Volume Methods



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

- Problem -> How to approximate the flux at cell edge

Example: Finite Volume Methods

$$u_t = -cu_x$$

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

- The simplest approach

$$F_{i-1/2}^n = \frac{1}{2}\left[f(Q_{i-1}^n) + f(Q_i^n)\right]$$

Unstable!!!

See `2_fvm`

Example: Finite Volume Methods

- The simplest approach

$$F^n_{i-1/2} = \frac{1}{2}\left[f(Q^n_{i-1}) + f(Q^n_i)\right]$$ Unstable!!!

- The Lax-Friedrichs Method

$$F^n_{i-1/2} = \frac{1}{2}\left[f(Q^n_{i-1}) + f(Q^n_i)\right] - \frac{\Delta x}{2\Delta t}(Q^n_i - Q^n_{i-1})$$

- The upwind method

$$F^n_{i-1/2} = f(Q^n_{i-1})$$

See `2_fvm`

Godunov's Method

- Algorithm (REA)
- Reconstruct a piecewise polynomial function q(i,n) from the cell averages Q(n,i).
- Evolve the hyperbolic equation exactly with this initial data q(i,n) a time step dt later.
- Average this ruction over each grid cell to obtain new cell averages Q(n+1,i)

High Resolution Methods

The Lax-Wendroff Method

- The Lax-Wendroff method is based on Tayler series expansion

$$q_t + cq_x = 0$$

$$q(x, t_{n+1}) = q(x, t_n) + \Delta t q_t(x, t_n) + \frac{1}{2}\Delta t^2 q_{tt}(x, t_n) + ...$$

$$q_{tt} = -cq_{xt} = c^2 q_{xx}$$

$$q(x, t_{n+1}) = q(x, t_n) + \Delta t q_t(x, t_n) + \frac{1}{2}\Delta t^2 c^2 q_{xx}(x, t_n) + ...$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{2\Delta x} c(Q_{i+1}^n - Q_{i-1}^n) + \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 c^2 (Q_{i-1}^n - 2Q_i^n + Q_{i+1}^n)$$

Exercise: The Lax-Wendroff Method

$$q_t + cq_x = 0$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{2\Delta x} c(Q_{i+1}^n - Q_{i-1}^n) + \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 c^2(Q_{i-1}^n - 2Q_i^n + Q_{i+1}^n)$$

or

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

See 2_fvm

Piecewise Linear Reconstruction

- Recall the REA algorithm, we can construct a piecewise linear function of the form

$$q(x, t_n) = Q_i^n + \sigma_i^n (x - x_i)$$

$$\text{for } x_{i-1/2} \leq x \leq x_{i+1/2}$$

Piecewise Linear Reconstruction

- Choice of slopes
  - Centered slope: $\sigma_i^n = \dfrac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x}$
  - Upwind slope: $\sigma_i^n = \dfrac{Q_i^n - Q_{i-1}^n}{\Delta x}$
  - Downwind slop: $\sigma_i^n = \dfrac{Q_{i+1}^n - Q_i^n}{\Delta x}$

Piecewise Linear Reconstruction

- Slope limiter (avoid oscillation)

$$\sigma_i^n = \text{minmod}\left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x}, \frac{Q_{i+1}^n - Q_i^n}{\Delta x}\right)$$

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

Piecewise Linear Reconstruction

- Flux can be calculated by

$$
\begin{aligned}
F^n_{i-1/2} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} cq^n(x_{i-1/2},t)dt \\
&= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} cq^n(x_{i-1/2}-c(t-t_n),t_n)dt \\
&= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} c\left[Q^n_{i-1}+(x_{i-1/2}-c(t-t_n)-x_{i-1})\sigma^n_{i-1}\right]dt \\
&= cQ^n_{i-1}+\frac{1}{2}c(\Delta x - c\Delta t)\sigma^n_{i-1}
\end{aligned}
$$

Exercise: Piecewise Linear Reconstruction

- Flux can be calculated by

$$
\begin{aligned}
F^n_{i-1/2} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} cq^n(x_{i-1/2},t)dt \\
&= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} cq^n(x_{i-1/2}-c(t-t_n),t_n)dt \\
&= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} c\left[Q^n_{i-1}+(x_{i-1/2}-c(t-t_n)-x_{i-1})\sigma^n_{i-1}\right]dt
\end{aligned}
$$

$$
= cQ^n_{i-1}+\frac{1}{2}c(\Delta x - c\Delta t)\sigma^n_{i-1}
$$

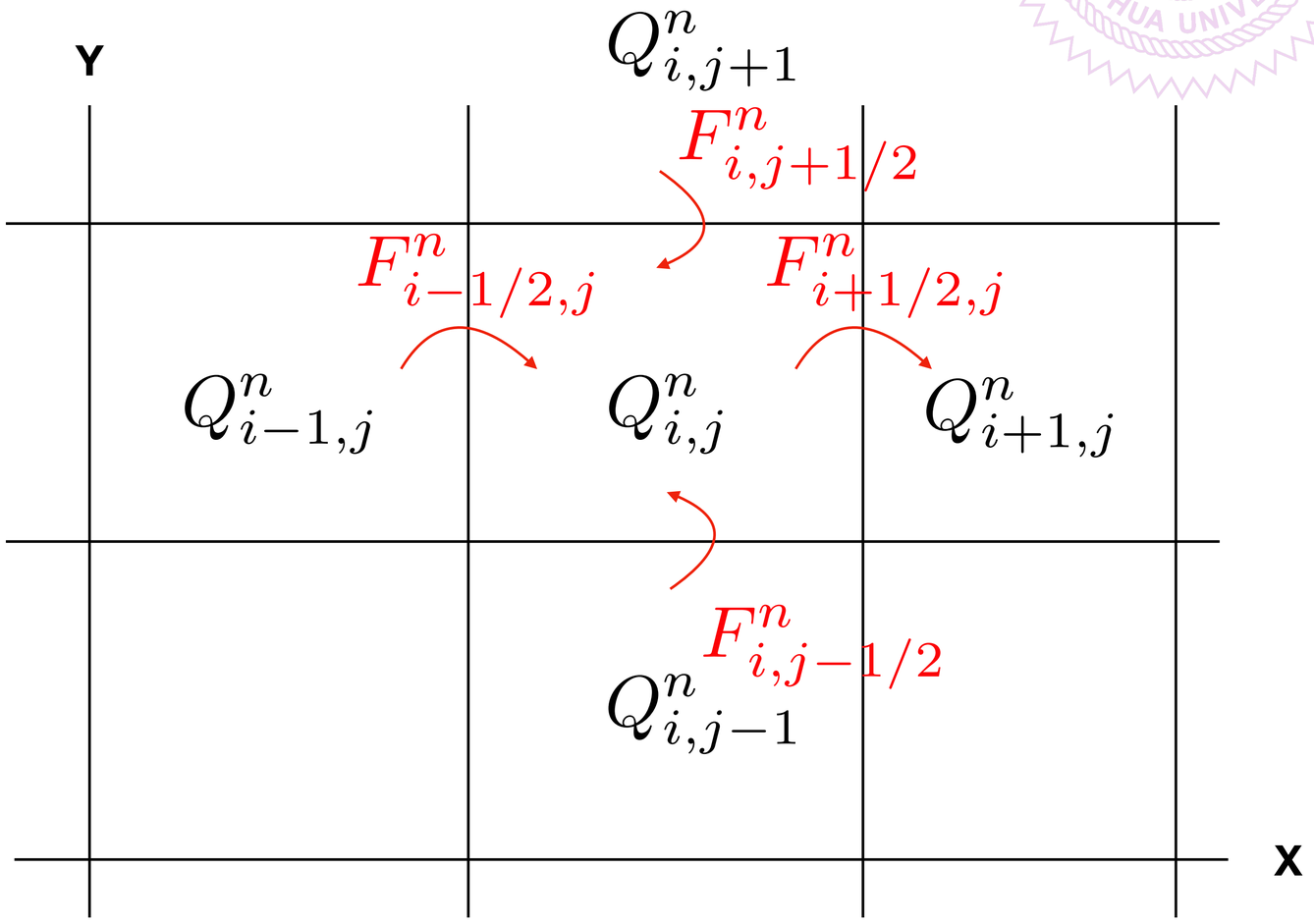
modify `2_fvm`

Multi-D Simulations

Multi-dimensional Advection

- 2D advection equation becomes

$$u_t = -c_x u_x - c_y u_y$$



Multi-dimensional Advection

- 2D advection equation becomes

$$u_t = -c_x u_x - c_y u_y$$

- Dimensionally split

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}\right) - \frac{\Delta t}{\Delta y}\left(F_{i,j+1/2}^{n} - F_{i,j-1/2}^{n}\right)$$

see 3_fvm2d

Problem Set 8

https://kuochuanpan.github.io/courses/109ASTR660_CA/

Next lecture

- Introduction to Fortran