Introduction to finite difference methods for hyperbolic equations

Luca Bonaventura

POLITECNICO MILANO 1863

DIPARTIMENTO DI MATEMATICA

Keio University, 9.7.15

MOX

Outline of the lecture

Key concepts introduced in the third lecture

- An approach to time discretization: the method of lines
- General derivation of finite difference operators
- Finite difference approximations of the advection equation
- Stability of finite difference approximations
- Finite difference approximations of nonlinear conservation laws

Time discretization approach: the method of lines

Semi-discretization in space

General approximation procedure: semi-discretization in space

- **General approximation procedure: semi-discretization in space**
- **PDE solution: continuous space-time variables** $c(x, t)$
- **Discretize in space (1): introduce discretization mesh with nodes** $x_i, i = 1, \ldots, N$ **and functions** $c_i(t) \approx c(x_i, t)$ **of continuous time variable**
- **Discretize in space (2): replace PDE by large ODE system**

$$\frac{\partial c}{\partial t} + a\frac{\partial c}{\partial x} = 0 \qquad \rightarrow \qquad \frac{d}{dt}\mathbf{c}(t) = \mathbf{A}\mathbf{c}(t) \quad \mathbf{c}(t) = [c_1(t), \ldots, c_N(t)]^T$$

Solution of the resulting ODE systems

- Application of standard methods for systems of ODEs
- Very large systems result: numerical weather prediction $O(10^7)$ variables, earthquake simulation $O(10^8)$ variables
- Choice of the time step is related to the choice of the spatial mesh: numerical stability requirements
- Stiff ODE systems often arise: special solvers are needed

Polynomial interpolation and finite differences

Polynomial interpolation of a function

Given distinct nodes $x_0, \ldots, x_n$ and values $f(x_i) \quad i = 0, \ldots, n$, there is only one polynomial of degree n such that $\Pi_n(f)(x_i) = f(x_i) \quad i = 0, \ldots, n:$

$$P_n(f)(x) = \sum_{i=0}^{n} f(x_i) l_i(x) \qquad l_i(x) = \Pi_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

- $l_i(x)$: **Lagrange basis functions,** $l_i(x_i) = 1, l_i(x_j) = 0 \ i \neq j$
- **Only one of many representations of the interpolating polynomial: theoretically equivalent, not computationally**
- **Essentially all methods for numerical approximation of PDE are based on polynomial interpolation**

Error in polynomial interpolation of a function

If $f \in \mathcal{C}^{n+1}$, there is $\xi \in \mathbf{R}$ such that

$$P_n(f)(x) - f(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi_{j=0}^{n}(x - x_j)$$

- **Good approximation if $|f^{(n+1)}|$ bounded by a reasonable number (f regular or smooth) and $|(x - x_j)|$ reasonably small**
- **Bad approximation if $|f^{(n+1)}|$ very large (f irregular or non smooth) or $|(x - x_j)|$ not sufficiently small**

Derivation of interpolating polynomials

$$\frac{d}{dx}P_n(f)(x) = \sum_{i=0}^{n} f(x_i) l_i'(x)$$

$$f'(x) - \frac{d}{dx}P_n(f)(x) \approx \frac{f^{(n+1)}}{(n+1)!}\frac{d}{dx}\sum_{j=0}^{n}\Pi_{k\neq j}(x - x_k)$$

- **Good approximation if $|f^{(n+1)}|$ bounded by a reasonable number and $|(x - x_j)|$ reasonably small**
- **Bad approximation if $|f^{(n+1)}|$ very large or $|(x - x_j)|$ not sufficiently small**

Finite difference approximation

Compute $dP_n(f)/dx$ at one of the interpolation nodes

$$\frac{d}{dx}P_n(f)(x_k) = \sum_{i=0}^{n} f(x_i)l_i'(x_k)$$

- **Can be generalized to higher order derivatives if $p \geq n$**

$$\frac{d^p}{dx^p}P_n(f)(x_k) = \sum_{i=0}^{n} f(x_i)\frac{d^p}{dx^p}l_i(x_k)$$

- **Usually on uniform meshes $x_i = i\Delta x$, but can be generalized to non uniform 1D and tensor product multidimensional meshes**
- **On a uniform mesh, for $f \in \mathcal{C}^{n+1}([0, L])$ and $n+1$ nodes $x_i = i\Delta x, \quad i = 0, \dots, n, \quad \Delta x = L/n$, one has**

$$f'(x_k) - \frac{d}{dx}P_n(f)(x_k) = O(\Delta x^n)$$

Basic finite difference operators (1)

All following definitions on uniform mesh

$x_i = i\Delta x, \quad i = 0, \dots, N, \qquad \Delta x = L/N,$ **with** $f_i = f(x_i)$

Forward and backward finite differences

$$\delta^+ f_i = \frac{f_{i+1} - f_i}{\Delta x} \qquad \delta^- f_i = \frac{f_i - f_i - 1}{\Delta x}$$

- **First order approximations:**

$$\delta^{\pm} f_i - f'(x_i) = O(\Delta x)$$

- **Also used in many time discretization methods: forward and backward Euler methods**

Basic finite difference operators (2)

Centered finite differences

$$\delta^{(0,2)} f_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x}$$

$$\delta^{(0,4)} f_i = \frac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12\Delta x}$$

- **Higher order approximations:**

$$\delta^{(0,2)} f_i - f'(x_i) = O(\Delta x^2) \qquad \delta^{(0,4)} f_i - f'(x_i) = O(\Delta x^4)$$

- **More accurate only if the function is sufficiently smooth, error constant depends on $|f^{(3)}|, |f^{(5)}|$, respectively**

Basic finite difference operators (3)

One sided second order operator

$$\delta^{++} f_i = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2\Delta x}$$

- **Second order approximation**

$$\delta^{++} f_i = f^{(1)}(x_i) + O(\Delta x^2)$$

- **Useful at boundaries of the domain and to develop higher order upwind methods**

Basic finite difference operators (4)

Second finite differences

$$\delta^{(2)} f_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2}$$

- **Second order approximation of second derivative:**

$$\delta^{(2)} f_i - f^{(2)}(x_i) = O(\Delta x^2)$$

- **Important to discretize advection diffusion equation and conservation law with viscosity term**

Finite differences approximations of the advection equation

The linear advection equation

Initial and boundary value problem

$$\frac{\partial c}{\partial t} + a\frac{\partial c}{\partial x} = 0 \qquad a > 0 \qquad x \in [0, L], \quad t \in [0, T]$$

- **Solution on a bounded interval $x \in [0, L]$: some information is needed about $c(0, t)$ for $t \in [0, T]$**
- **Mathematical formulation as initial and boundary value problem: assuming that $c(x, 0) = c_0(x)$ is known for $t = 0$ and $c(0, t) = g(t)$ is known for $t \in [0, T]$, determine $c(x, t)$ at later times**

General finite difference approximation

- **Introduce a uniform mesh**

$$x_i = i\Delta x, \quad i = 0, \dots, N, \qquad \Delta x = L/N$$

- **Introduce quantities** $c_i(t) \approx c(x_i, t)$ **and assume that** $c_i(t)$ **satisfy**

$$\frac{d}{dt} c_i + a\ \delta c_i = 0 \qquad i = 1, \dots, N-1$$

- **Use boundary conditions to define** $c_0(t) = c(0, t) = g_0(t)$ **or** $c_N(t) = c(L, t) = g_L(t)$ **depending on the sign of** a
- **Solve the resulting ODE system**

$$\frac{d}{dt}\mathbf{c}(t) = \mathbf{A}\mathbf{c}(t) + \mathbf{g}(t)$$

Accuracy of finite difference approximation (1)

- **Let $\mathbf{c}_{ex}(t) = [c(x_1, t), \ldots, c(x_N, t)]^T$ be values of exact solution**
- **Spatial truncation error**

$$\tau(\Delta x) = \left\| \frac{d}{dt}\mathbf{c}_{ex}(t) - \mathbf{A}\mathbf{c}_{ex}(t) - \mathbf{g}(t) \right\| \neq 0$$

- **Spatial discretization is consistent of order p if for sufficiently small Δx, uniformly in $t \in [0, T]$**

$$\tau(\Delta x) \leq K \Delta x^p$$

- **Spatial discretization is convergent of order p if for sufficiently small Δx, uniformly in $t \in [0, T]$**

$$\|\mathbf{c}_{ex}(t) - \mathbf{c}(t)\| \leq K \Delta x^p$$

Accuracy of finite difference approximation (2)

- Consistency is a necessary condition for convergence
- However, for simplicity often is the only one checked
- The order of convergence of the finite difference operators employed determines the order of consistency and convergence of the method
- Total accuracy of numerical solutions depends on space and time discretizations: order of convergence of space and time method should be related

Basic methods (1)

First order upwind method

$$\frac{d}{dt}c_i + a\delta^- c_i = \frac{d}{dt}c_i + a\frac{c_i - c_{i-1}}{\Delta x} = 0 \qquad i = 1, \ldots, N$$

- **First order method obtained from backward finite difference in space**
- **Most often coupled to forward finite difference in time:**
  $t^n = n\Delta t, \quad n = 0, \ldots, M \quad \Delta t = T/M \quad c_i^n \approx c_i(t^n)$

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + a\frac{c_i^n - c_{i-1}^n}{\Delta x} = 0 \qquad i = 1, \ldots, N-1$$

Basic methods (2)

Second order centered method

$$\frac{d}{dt}c_i + a\delta^{(0,2)}c_i = \frac{d}{dt}c_i + a\frac{c_{i+1} - c_{i-1}}{2\Delta x} = 0 \qquad i = 1, \dots, N-1$$

- **Second order method obtained from centered finite difference in space**
- **Often coupled to centered finite difference in time:**
  $t^n = n\Delta t, \quad n = 0, \dots, M \quad \Delta t = T/M \quad c_i^n \approx c_i(t^n)$

$$\frac{c_i^{n+1} - c_i^{n-1}}{2\Delta t} + a\frac{c_{i+1}^n - c_{i-1}^n}{2\Delta x} = 0 \qquad i = 1, \dots, N-1$$

- **Does not work with simple forward finite differences in time: must be coupled to Runge Kutta of order 3 or higher**

MOX

Basic methods (3)

Fourth order centered method

$$\frac{d}{dt}c_i + a\frac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12\Delta x} = 0 \qquad i = 2, \dots, N-2$$

- Fourth order method obtained from centered finite difference
- Does not work with forward finite differences in time: must be coupled to Runge Kutta of order 3 or higher

Basic methods (4)

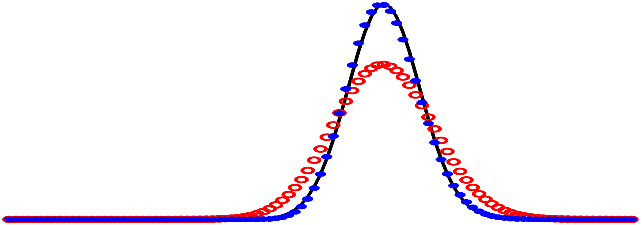
Second order upwind method

$$\frac{d}{dt}c_i + a\delta^{++}c_i = \frac{d}{dt}c_i + a\frac{3c_i - 4c_{i-1} + c_{i-2}}{2\Delta x} = 0 \qquad i = 2, \dots, N$$

- **Second order method obtained from second order upwind finite difference**
- **Coupled to forward finite difference in time:**
  $t^n = n\Delta t, \quad n = 0, \dots, M \quad \Delta t = T/M \quad c_i^n \approx c_i(t^n)$

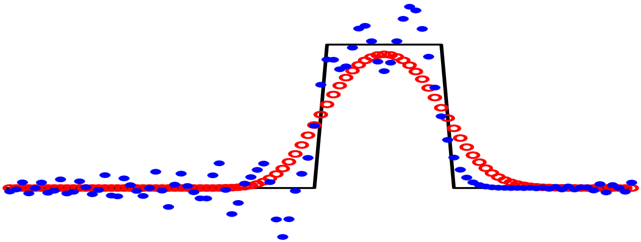
$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + a\frac{3c_i - 4c_{i-1} + c_{i-2}}{2\Delta x} = 0 \qquad i = 1, \dots, N-1$$

Let's have a look: smooth solutions



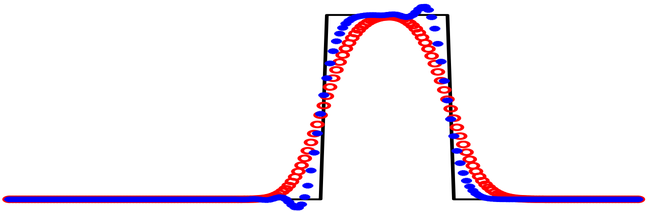
- First order upwind finite difference method and second order centered finite difference method, coupled to Runge Kutta method of order 3 in time
- First order method displays numerical diffusion, second order method is more accurate but with some phase error

Let's have a look: non smooth solutions



- **First order upwind finite difference method and second order centered finite difference method, coupled to Runge Kutta method of order 3 in time**
- **First order method displays numerical diffusion, but no spurious oscillations, second order method is not more accurate**

Let's have a look: non smooth solutions with numerical diffusion



- First order upwind finite difference method and second order centered finite difference method, coupled to Runge Kutta method of order 3 in time
- Numerical (artificial) diffusion added to centered finite difference method

Stability of finite difference approximations

Stability of space time approximation

- Errors due to space and time discretizations interact
- Space discretization errors can in some cases accumulate catastrophically when applying repeatedly time discretization
- Property that identifies this behaviour: stability
- Difficult mathematical analysis, we will only consider simple example

Stability of explicit upwind method (1)

- **Introduce a uniform mesh in space and time**

$$x_i = i\Delta x, \qquad \Delta x = L/N \qquad t^n = n\Delta t, \qquad \Delta t = T/M$$

- **Consider upwind approximation of the advection equation with** $a > 0$

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + a\frac{c_i^n - c_{i-1}^n}{\Delta x} = 0$$

- **Rewrite as**

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

Stability of explicit upwind method (2)

- **Stability means that bounded approximations remain bounded: we want to guarantee that** $\max |c_i^{n+1}| \leq \max |c_i^n|$
- **Considering**

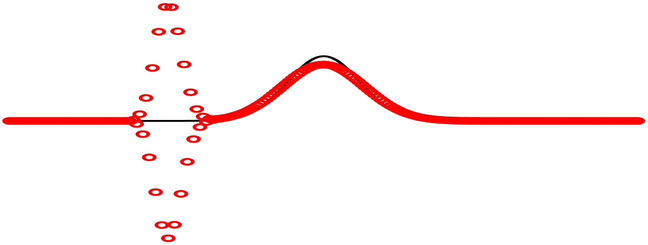
$$c_i^{n+1} = \left(1 - a\frac{\Delta t}{\Delta x}\right) c_i^n + a\frac{\Delta t}{\Delta x} c_{i-1}^n$$

it is clear that sufficient condition for stability is

$$a\frac{\Delta t}{\Delta x} \leq 1$$

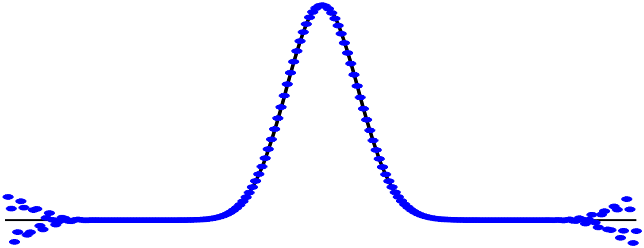
- **Courant Friedrichs Lewy (1928!!!) stability condition on the Courant number** $a\Delta t/\Delta x$

Let's have a look: numerical instability (1)



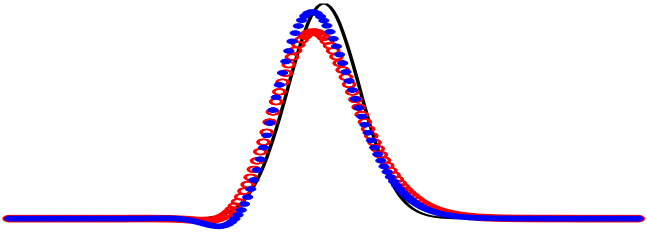
- **First order upwind finite difference method coupled to first order explicit Euler method in time, Courant number larger than 1: unphysical oscillations that are not related to regularity of the solution**
- **This is early stage of instability, total nonsense follows...**

Let's have a look: numerical instability (2)



- **Second order centered finite difference method coupled to Runge Kutta method of order 3 in time, Courant number larger than 1: unphysical oscillations that are not related to regularity of the solution**
- **This is early stage of instability, total nonsense follows...**

Let's have a look: numerical instability (3)



- First order upwind and second order centered finite difference method coupled to Implicit Crank Nicolson method, Courant number larger than 1
- Phase error, but no unphysical oscillation
- Each step of the numerical solution is computationally more expensive, but less steps can be employed

Finite differences approximations of nonlinear conservation laws

Nonlinear conservation laws as nonlinear advection

Advective form of conservation law

$$\frac{\partial c}{\partial t} + f'(c)\frac{\partial c}{\partial x} = 0$$

- Finite difference approximations usually start from advective form of nonlinear conservation laws
- Usually they do not guarantee discrete conservation

General finite difference approximation

- **Introduce a uniform mesh**

$$x_i = i\Delta x, \quad i = 0, \dots, N, \qquad \Delta x = L/N$$

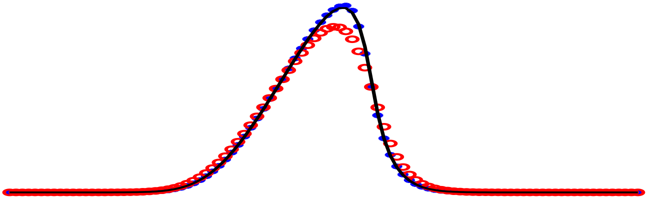
- **Introduce quantities $c_i(t) \approx c(x_i, t)$ and assume that $c_i(t)$ satisfy**

$$\frac{d}{dt}c_i + f'(c_i)\ \delta c_i = 0 \qquad i = 1, \dots, N-1$$

- **Use boundary conditions to define $c_0(t) = c(0, t) = g_0(t)$ or $c_N(t) = c(L, t) = g_L(t)$ depending on the sign of $f'(c_i)$ close to the boundary**
- **Solve the resulting ODE system**

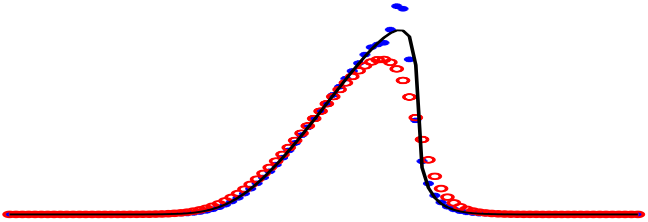
$$\frac{d}{dt}\mathbf{c}(t) = \mathbf{f}(\mathbf{c}(t), t)$$

Let's have a look: smooth solutions



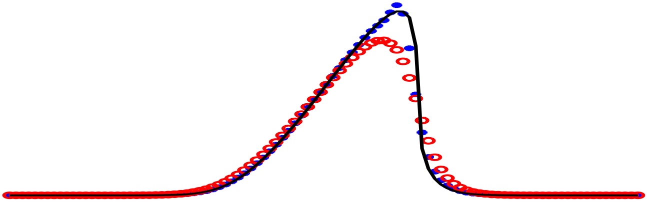
- **First order upwind and second order centered finite difference method coupled to Runge Kutta method of order 3 in time**
- **Good approximation of solution, diffusion error with first order upwind method**

Let's have a look: less smooth solutions



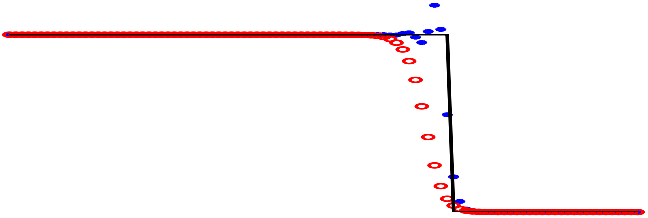
- **First order upwind and second order centered finite difference method coupled to Runge Kutta method of order 3 in time**
- **Spurious oscillations arise with second order method where solution is less regular**

Let's have a look: add numerical diffusion



- **First order upwind and second order centered finite difference method with artificial numerical diffusion, coupled to Runge Kutta method of order 3 in time**
- **Spurious oscillations are damped but will not vanish completely if numerical diffusion is not really large**

Let's have a look: non smooth solutions



- **Shock wave solution of Riemann problem for Burgers equation**
- **First order upwind and second order centered finite difference method coupled to Runge Kutta method of order 3 in time**
- **Spurious oscillations arise with second order method around shock**

Let's have a look: upwind methods

- Shock wave solution of Riemann problem for Burgers equation
- First order upwind and second order upwind finite difference method coupled to Runge Kutta method of order 3 in time
- Less spurious oscillations arise with second order upwind method around shock, but non conservative upwind methods tend to produce wrong shock speed

Summary of properties of finite difference approximations

- Easy to define for arbitrary meshes in 1D, uniform tensor product meshes in multiple dimensions
- Approximation depends on the regularity of the solution
- Accurate approximation for regular solutions
- Spurious oscillations if the solution has large derivatives, the only simple cure is adding numerical diffusion terms
- Not conservative, wrong speed for shock waves
- Cannot be easily generalized to unstructured meshes in multiple dimensions