Numerical simulation of flows with shocks

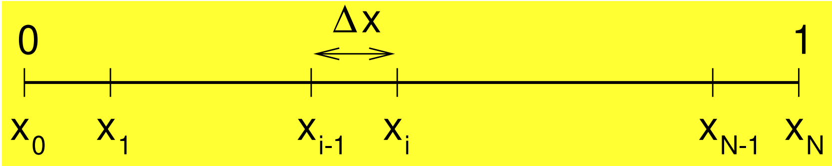
2.1 1D finite difference schemes

2.1.1 Scalar conservation laws

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

$$\frac{\partial u}{\partial t} + f'(u)\,\frac{\partial u}{\partial x} = 0 \qquad \text{with} \qquad f'(u) = \frac{\partial f(u)}{\partial u}$$

- continuous domain $\Rightarrow$ finite number of *grid points*



e.g. equidistant grid: $x_i = i\Delta x = \dfrac{i}{N}$

- point-values of functions $f(x)$ $\Rightarrow$ $\{f_i \equiv f(x_i), i = 0, 1, \ldots, N\}$
- in space and time: $f_i^n \equiv f(x_i, t_n)$

Finite difference approximations

- derivatives $\Rightarrow$ truncated Taylor series expansions, e.g.

$$u_{i\pm1} = u_i \pm \left.\frac{\partial u}{\partial x}\right|_i \Delta x + \left.\frac{\partial^2 u}{\partial x^2}\right|_i \frac{\Delta x^2}{2!} \pm \left.\frac{\partial^3 u}{\partial x^3}\right|_i \frac{\Delta x^3}{3!} + O(\Delta x^2) \qquad \text{[1a] and [1b]}$$

[1a]: $$\left.\frac{\partial u}{\partial x}\right|_i = \frac{u_{i+1}-u_i}{\Delta x} - \frac{\Delta x}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i + O(\Delta x^2)$$ '1st-order forward'

[1b]: $$\left.\frac{\partial u}{\partial x}\right|_i = \frac{u_i-u_{i-1}}{\Delta x} + \frac{\Delta x}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i + O(\Delta x^2)$$ '1st-order backward'

[1a]–[1b]: $$\left.\frac{\partial u}{\partial x}\right|_i = \frac{u_{i+1}-u_{i-1}}{2\Delta x} + 0\,\frac{\Delta x}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i + O(\Delta x^2)$$ '2nd-order central'

- 2nd-order derivatives:

[1a]+[1b]: $$\left.\frac{\partial^2 u}{\partial x^2}\right|_i = \frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2} + O(\Delta x^2)$$ (2nd-order)

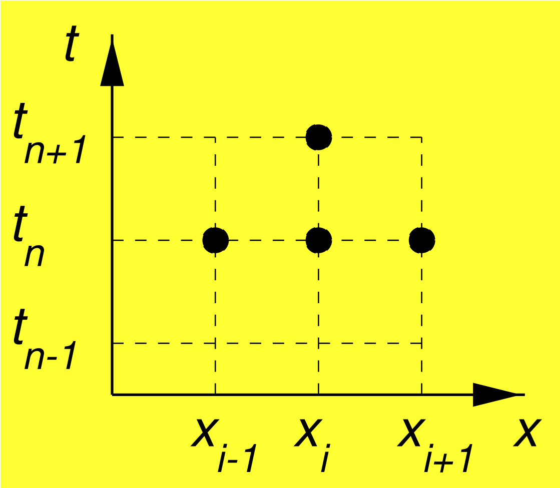
Four schemes for the linear advection equation

$$f(u) = a\,u \quad \Rightarrow \quad \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$ (linear equation; assume $a > 0$)

- 1. Forward Central (FC):

$$\underbrace{\frac{u_i^{n+1} - u_i^n}{\Delta t}}_{\text{1st-order forward FD}} + a\,\underbrace{\frac{u_{i+1}^n - u_{i-1}^n}{2\,\Delta x}}_{\text{2nd-order central FD}}$$

with stencil:



$$\Rightarrow \boxed{u_i^{n+1} = u_i^n - a\,\frac{\Delta t}{\Delta x}\,\frac{u_{i+1}^n - u_{i-1}^n}{2}}$$

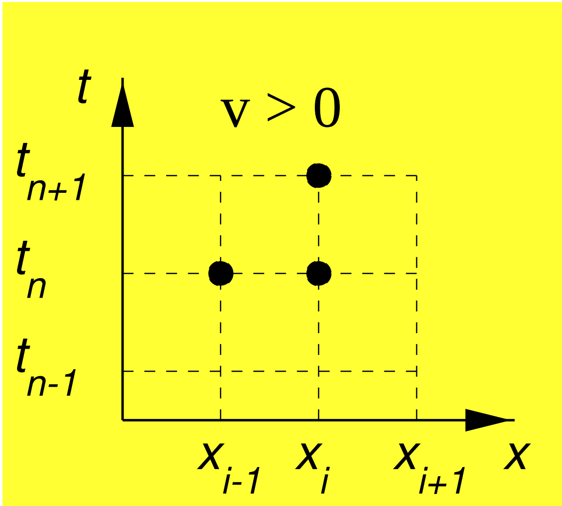
$O(\Delta x^2, \Delta t)$ ($\Rightarrow$ consistent!)

BUT: numerically unstable

- 2. First Order Upwind (FOU):

$$\underbrace{\frac{u_i^{n+1}-u_i^n}{\Delta t}}_{\text{1st-order forward FD}} + a\,\underbrace{\frac{u_i^n-u_{i-1}^n}{\Delta x}}_{\text{1st-order left FD}}$$

with stencil:



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

$O(\Delta x, \Delta t)$ (consistent)

BUT: not accurate – high diffusion

$\Rightarrow$ if the sign of a is not specified in advance:

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

with the definitions: $a^+ = max(0,a)$ and $a^- = min(0,a)$

- 3. Lax-Wendroff (LW):

$$\frac{\partial u}{\partial t} = -a\,\frac{\partial u}{\partial x} \quad \Rightarrow \quad \frac{\partial^2 u}{\partial t^2} = -a\,\frac{\partial^2 u}{\partial x \partial t} = a^2\,\frac{\partial^2 u}{\partial x^2}$$

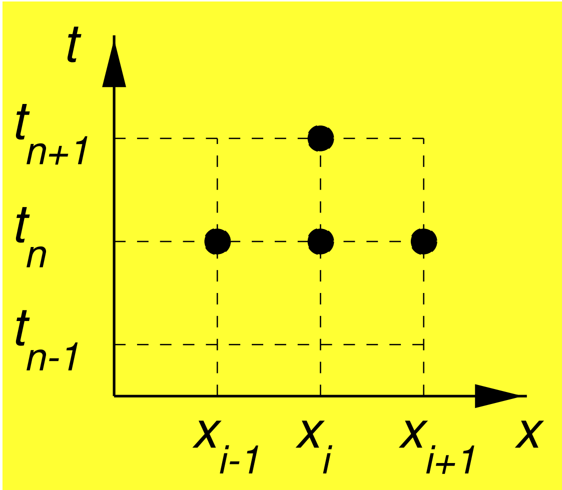
consider Taylor expansion in time:

$$u_i^{n+1} = u_i^n + \left.\frac{\partial u}{\partial t}\right|_i^n \Delta t + \left.\frac{\partial^2 u}{\partial t^2}\right|_i^n \frac{\Delta t^2}{2!} + O(\Delta t^3)$$

$$\Rightarrow u_i^{n+1} = u_i^n - a\left.\frac{\partial u}{\partial x}\right|_i^n \Delta t + a^2\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n \frac{\Delta t^2}{2!} + O(\Delta t^3)$$

$$\boxed{\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\,\frac{u_{i+1}^n - u_{i-1}^n}{2\,\Delta x} - \frac{a^2}{2}\,\Delta t\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = 0} \quad O(\Delta x^2, \Delta t^2)$$

with stencil:



$O(\Delta x^2, \Delta t^2)$ (consistent)

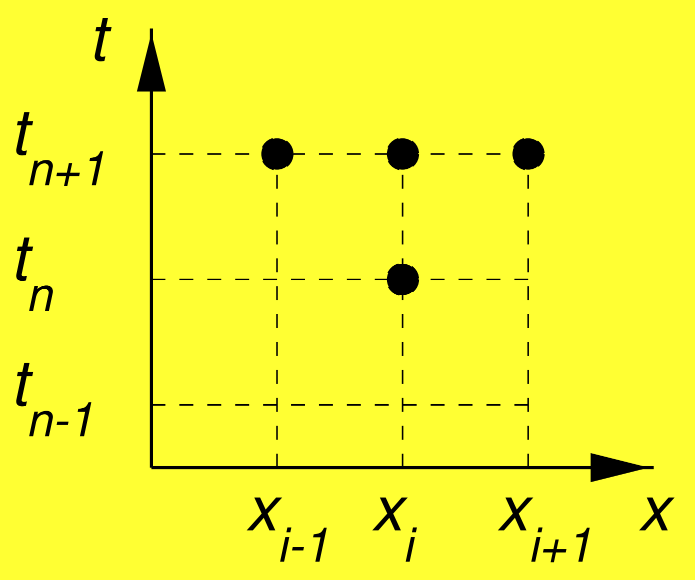
BUT: not positive – oscillations

- 4. Backward Central (BC):

$$\underbrace{\frac{u_i^{n+1}-u_i^n}{\Delta t}}_{\text{1st-order forward FD}} + a\,\underbrace{\frac{u_{i+1}^{n+1}-u_{i-1}^{n+1}}{2\,\Delta x}}_{\text{2nd-order central FD}} \Rightarrow \boxed{u_i^{n+1} = u_i^n - a\,\frac{\Delta t}{\Delta x}\,\frac{u_{i+1}^{n+1}-u_{i-1}^{n+1}}{2}}$$

$O(\Delta x^2, \Delta t)$ (consistent)

with stencil:



IMPLICIT scheme:

- all u_i at time level $n+1$ are coupled

- need matrix inversion in every time step

- schemes 1.–3. are EXPLICIT:

no matrix inversion needed

numerically stable, BUT: not positive – oscillations

$\Rightarrow$ REMARK: schemes 1.–4. are LINEAR in the u_i

Four requirements for a shock-capturing numerical scheme

1. Accurate – low diffusion
2. Numerically stable
3. Positive, no spurious oscillations – low dispersion
4. Conservative – the Lax-Wendroff theorem

1. Accurate – low diffusion

linear advection-diffusion equation: $$\boxed{\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \eta\frac{\partial^2 u}{\partial x^2}}$$

consider discretization of spatial part:

- FC: $$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\,\frac{u_{i+1}^n - u_{i-1}^n}{2\,\Delta x} = 0$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\,\left.\frac{\partial u}{\partial x}\right|_i + 0\,\left.\frac{\partial^2 u}{\partial x^2}\right|_i + a\,\frac{\Delta x^2}{3!}\,\left.\frac{\partial^3 u}{\partial x^3}\right|_i + O(\Delta x^3) = 0$$

$\Rightarrow$ NO DIFFUSION (but: unstable) linear advection-diffusion equation: $$\boxed{\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \eta\frac{\partial^2 u}{\partial x^2}}$$

- FOU: $$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\,\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \qquad (a > 0)$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\left.\frac{\partial u}{\partial x}\right|_i - a\frac{\Delta x}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_i + a\frac{\Delta x^2}{3!}\left.\frac{\partial^3 u}{\partial x^3}\right|_i + O(\Delta x^3) = 0$$

- also: rewrite FOU as

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - \frac{|a|\Delta x}{2}\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = 0$$

- second order accurate discretization of the spatial part of the *linear advection-diffusion equation* with diffusion coefficient $\eta_{num} = |a|\Delta x/2$.
- this *numerical diffusion* vanishes in first order in Δx

$\Rightarrow$ NUMERICAL DIFFUSION: we want to minimize it, but necessary for stability (explicit schemes) and positivity (see below) linear advection-diffusion equation: $\boxed{\dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = \eta\dfrac{\partial^2 u}{\partial x^2}}$

REMARK: interaction of spatial and temporal discretization may lead to cancellation of errors:

- LW: diffusion term, but cancels with first order error in temporal discretization

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\,\frac{u_{i+1}^n - u_{i-1}^n}{2\,\Delta x} - \frac{a^2}{2}\,\Delta t\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = 0 \quad O(\Delta x^2, \Delta t^2)$$

$\Rightarrow$ NO DIFFUSION (second order!), stable

LW is the unique explicit linear 2-level scheme which is second order in space and time

(but: not positive)

linear advection-diffusion equation: $\boxed{\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \eta\frac{\partial^2 u}{\partial x^2}}$

REMARK: dispersion relation: $u(x,t) = \text{exp}(i(kx - \omega t))$

$$-i\,\omega\,u + a\,i\,k\,u = \eta\,(i\,k)^2\,u$$

$$\boxed{\omega - a\,k + \eta\,i\,k^2 = 0}$$

$\Rightarrow$ complex dispersion relation = damped waves

($\eta = 0 \;\Rightarrow$ undamped waves with phase velocity $= \omega/k = a$,

and group velocity $= \partial\omega/\partial k = a$)

2. Numerically stable

numerical solution=exact solution+error

$$\hat{u}_j^n = u_j^n + e_j^n$$

- FOU:

$$\frac{\hat{u}_j^{n+1} - \hat{u}_j^n}{\Delta t} + a\,\frac{\hat{u}_j^n - \hat{u}_{j-1}^n}{\Delta x} = 0$$

$$\Rightarrow \underbrace{\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\frac{u_j^n - u_{j-1}^n}{\Delta x}}_{\text{new error: O}(\Delta x, \Delta t)} + \underbrace{\frac{e_j^{n+1} - e_j^n}{\Delta t} + a\,\frac{e_j^n - e_{j-1}^n}{\Delta x}}_{\text{amplification of error}} = 0 \qquad \text{(linear!)}$$

requirement for numerical stability: existing errors are not amplified

Fourier decomposition of error on grid (mode m): (von Neumann method)

$$e_j^n = \bar{e}_m^n \exp(i\, m\, \pi\, \Delta x\, j) = \bar{e}_m^n \exp(i\theta_m j)$$

in FOU:

$$\frac{\bar{e}_m^{n+1}\,\text{exp}(i\theta_m j) - \bar{e}_m^n\,\text{exp}(i\theta_m j)}{\Delta t} + a\,\bar{e}_m^n\,\text{exp}(i\theta_m j)\frac{1-\text{exp}(-i\theta_m)}{\Delta x} = 0$$

$$\Rightarrow \frac{\bar{e}_m^{n+1}}{\bar{e}_m^n} = 1 - a\,\frac{\Delta t}{\Delta x}\,\underbrace{(1-\text{exp}(-i\theta_m))}_{0\ldots 2}$$

$$\left|\frac{\bar{e}_m^{n+1}}{\bar{e}_m^n}\right| \le 1 \quad \Leftrightarrow \quad |a|\frac{\Delta t}{\Delta x} \le 1$$

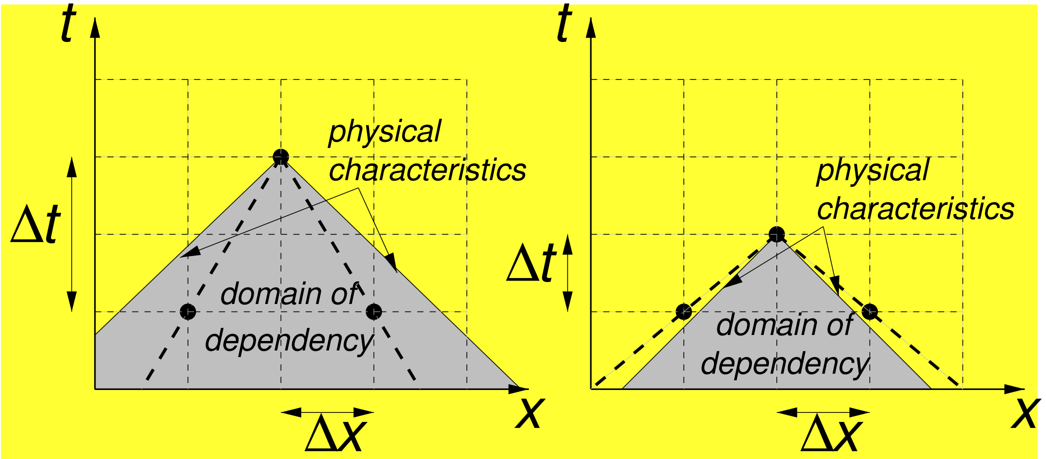
$$\Rightarrow \Delta t \le \frac{\Delta x}{|a|}$$

= CFL condition (Courant-Friedrichs-Levy)

$\Rightarrow$ FOU is conditionally stable

Physical interpretation of CFL condition

- hyperbolic system: information propagates along real characteristic curves $\Rightarrow$ domain of dependence



- for systems: several wave speeds, domain of dependence delineated by characteristics with smallest and largest wave speeds

- slopes of characteristics = wave speeds a_{min} and a_{max}

- $\Delta t \leq \dfrac{\Delta x}{|a|}$ $\Rightarrow$ numerical domain of dependence $\geq$ physical domain of dependence

- FC: (explicit, central in space)

$$\boxed{u_i^{n+1} = u_i^n - a\,\frac{\Delta t}{\Delta x}\,\frac{u_{i+1}^n - u_{i-1}^n}{2}}$$

$$\Rightarrow \frac{\bar{e}_m^{n+1}}{\bar{e}_m^n} = 1 - a\,\frac{\Delta t}{\Delta x}\,\underbrace{\frac{(\exp(i\theta_m) - \exp(-i\theta_m))}{2}}_{isin(\theta_m)}$$

$\Rightarrow$ unconditionally unstable!

- BC: (implicit, central in space)

$$\boxed{u_i^{n+1} = u_i^n - a\,\frac{\Delta t}{\Delta x}\,\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2}}$$

$$\Rightarrow 1 = \frac{\bar{e}_m^n}{\bar{e}_m^{n+1}} - a\,\frac{\Delta t}{\Delta x}\,\underbrace{\frac{(\exp(i\theta_m) - \exp(-i\theta_m))}{2}}_{isin(\theta_m)}$$

$\Rightarrow$ unconditionally stable!

- LW: (explicit)

$\Rightarrow$ conditionally stable under CFL condition

- conclusion: numerical diffusion (FOU) or implicit time integration (BC) can make scheme stable

3. Positive, no spurious oscillations – low dispersion

linear advection equation with dispersive term: $\boxed{\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^3 u}{\partial x^3}}$

dispersion relation: $u(x,t) = \exp(i(kx - \omega t))$

$$-i\,\omega\,u + a\,i\,k\,u = \mu(i\,k)^3\,u$$

$\boxed{-\omega + a\,k = -\mu\,k^3}$ $\boxed{v_{ph} = \frac{\omega}{k} = a + \mu\,k^2}$ $\boxed{v_g = \frac{\partial \omega}{\partial k} = a + 3\,\mu\,k^2}$

$\Rightarrow$ dispersion = Gibbs phenomenon = oscillations at discontinuities

$\Rightarrow$ central space discretization = no diffusion term ($\sim$ second derivative)

leading error term = dispersive term ! ($\sim$ third derivative) $\Rightarrow$ oscillations!

$\Rightarrow$ solution: we need diffusion, but will make schemes first order at discontinuities

Positive schemes

- how can we know where to add diffusion, and how much?

$\Rightarrow$ one answer: consider positive schemes

- $\dfrac{\partial u_j}{\partial t} = c_{j-1}\,u_{j-1} + c_j\,u_j + c_{j+1}\,u_{j+1}$

  consistency ($u(x,t)$ constant is a solution) $\Rightarrow c_{j-1} + c_j + c_{j+1} \equiv 0$

$\Rightarrow \dfrac{\partial u_j}{\partial t} = c_{j-1}\,(u_{j-1} - u_j) + c_{j+1}\,(u_{j+1} - u_j)$

- define positive scheme $\Leftrightarrow c_{j-1} \geq 0$ and $c_{j+1} \geq 0$

$\Rightarrow$ it follows that local extrema ($\sim$ oscillations) are suppressed:

$$\text{if } u_j \text{ is a local maximum } \Rightarrow \frac{\partial u_j}{\partial t} < 0$$

$$\text{if } u_j \text{ is a local minimum } \Rightarrow \frac{\partial u_j}{\partial t} > 0$$

- FOU:

$$\frac{\partial u_j}{\partial t} = \frac{a^+}{\Delta x}(u_{j-1} - u_j) - \frac{a^-}{\Delta x}(u_{j+1} - u_j)$$

$\Rightarrow c_{j-1} \geq 0$ and $c_{j+1} \geq 0$

$\Rightarrow$ FOU is positive! $\Rightarrow$ no spurious oscillations

- positivity is related to concept of Total Variation Diminishing (TVD) schemes, but more easily extendable to multiple spatial dimensions

4. Conservative – the Lax-Wendroff theorem

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

'natural' discretization: CONSERVATIVE

$$\boxed{\frac{\partial u_j}{\partial t} + \frac{\tilde{f}_{j+1/2} - \tilde{f}_{j-1/2}}{\Delta x} = 0}$$

with $\tilde{f}_{j+1/2}$ the numerical flux function at $j + 1/2$

$$\Rightarrow u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\tilde{f}_{j+1/2} - \tilde{f}_{j-1/2})$$

$$\Rightarrow \sum_j u_j^{n+1} = \sum_j u_j^n$$

$\Rightarrow$ exact conservation at the discrete level! (= natural discretization)

- problem: for flows with shocks, some non-conservative schemes converge to solutions with wrong shock speed

  $\Rightarrow$ Lax-Wendroff theorem (in 1D) : for a consistent and conservative scheme:

  if the scheme converges for a given problem, then it will converge to a weak solution (with the right shock speed)

$\Rightarrow$ numerical schemes need to be conservative

BUT: how about 2D and 3D, unstructured grids, some non-conservative schemes seem to give right shock speed, . . . ?

- example: FOU is conservative:

$$\boxed{\frac{\partial u_j}{\partial t} + \frac{\tilde{f}_{j+1/2} - \tilde{f}_{j-1/2}}{\Delta x} = 0}$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a^+ \frac{u_j^n - u_{j-1}^n}{\Delta x} + a^- \frac{u_{j+1}^n - u_j^n}{\Delta x} = 0$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{[a^+ u_j^n + a^- u_{j+1}^n] - [a^+ u_{j-1}^n + a^- u_j^n]}{\Delta x} = 0$$

$$\Rightarrow \tilde{f}_{j+1/2} = a^+ u_j^n + a^- u_{j+1}^n$$

$$\Rightarrow \tilde{f}_{j+1/2} = \underbrace{\frac{a\,u_j + a\,u_{j+1}}{2}}_{\text{central, second order}} - \underbrace{\frac{1}{2}|a|\,(u_{j+1} - u_j)}_{\text{numerical diffusion term}}$$

Nonlinear schemes – the Godunov theorem

- FOU: linear scheme

- 1. Accurate – low diffusion: NO (first order)
- 2. Numerically stable: yes, CFL
- 3. Positive, no spurious oscillations: yes
- 4. Conservative: yes

- BC: linear scheme

- 1. Accurate – low diffusion: yes (second order)
- 2. Numerically stable: yes, unconditionally
- 3. Positive, no spurious oscillations: NO (dispersion)
- 4. Conservative: yes

$\Rightarrow$ Godunov's theorem: a linear positive scheme cannot be second order

$\Rightarrow$ conversely: we need non-linear schemes to achieve both positivity and second order (away from shocks)

Linear reconstruction: the Minmod slope limiter

starting from FOU, construct a second order conservative scheme

FOU ($a > 0$): $\dfrac{u_j^{n+1} - u_j^n}{\Delta t} + a\,\dfrac{u_j^n - u_{j-1}^n}{\Delta x} = 0$

conservative: $u_j^{n+1} = u_j^n - \dfrac{\Delta t}{\Delta x}(\tilde{f}_{j+1/2} - \tilde{f}_{j-1/2})$

$\tilde{f}_{j+1/2} = a\,u_j^n$: first order

$\Rightarrow$ use the gradient to reconstruct u at the cell interface with c_1 and c_2 some constants ($c_1 + c_2 = 1/2$):

$$\tilde{f}_{j+1/2} = a\,(u_j^n + c_1\,(u_{j+1}^n - u_j^n) + c_2\,(u_j^n - u_{j-1}^n))$$

second order, but not positive: oscillations

$\Rightarrow$ use non-linear limiter function Ψ to determine the slope with which to reconstruct

$$\tilde{f}_{j+1/2} = a\,(u_j^n + \frac{1}{2}\,\Psi(\frac{u_{j+1}^n - u_j^n}{u_j^n - u_{j-1}^n})\,(u_j^n - u_{j-1}^n))$$

$\Rightarrow$ conditions on $\Psi(r)$ can be derived such that positivity is satisfied

$$r = \frac{u_{j+1}^n - u_j^n}{u_j^n - u_{j-1}^n}$$

- example: MINMOD limiter function

$$\Psi(r) = \max(0, \min(r, 1))$$

$\Rightarrow$ if slopes same sign ($r > 0$): take smallest slope

$\Rightarrow$ if slopes different sign (oscillation!): take slope 0 $\Rightarrow$ first order at shock

Nonlinear equations

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

$$\frac{\partial u}{\partial t} + f'(u)\,\frac{\partial u}{\partial x} = 0 \qquad \text{with} \quad f'(u) = \frac{\partial f(u)}{\partial u} = a(u)$$

$\Rightarrow$ generalize the first order upwind scheme for linear equation:

$$\tilde{f}_{j+1/2} = \frac{a\,u_j + a\,u_{j+1}}{2} - \frac{1}{2}\,|a|\,(u_{j+1} - u_j)$$

becomes

$$\tilde{f}_{j+1/2} = \frac{f(u_j) + f(u_{j+1})}{2} - \frac{1}{2}\,|\tilde{a}(u_j, u_{j+1})|\,(u_{j+1} - u_j)$$

with, e.g., $\tilde{a}(u_j, u_{j+1}) = a(\frac{u_j + u_{j+1}}{2})$

$\Rightarrow$ this is the (local) Lax-Friedrichs scheme other more sophisticated choices for $\tilde{a}$ are possible

positive second-order schemes can be obtained by taking non-linearly reconstructed values in stead of u_j and u_{j+1}

2.1.2 Systems of conservation laws

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0$$

linear systems

Apply first order upwind to every equation separately:

$$\boxed{\frac{\partial U}{\partial t} + \mathsf{A} \cdot \frac{\partial U}{\partial x} = 0}$$

hyperbolic $\Rightarrow \mathsf{A} = \mathsf{R} \cdot \Lambda \cdot \mathsf{L}$

define $\boxed{W = \mathsf{L} \cdot U : \quad \text{characteristic variables } W}$ $\Rightarrow$ $\boxed{\frac{\partial W}{\partial t} + \Lambda \cdot \frac{\partial W}{\partial x} = 0}$

for every component of W (take, e.g., first component):

$$\frac{w_{1i}^{n+1} - w_{1i}^{n}}{\Delta t} + \lambda_1^{+} \frac{w_{1i}^{n} - w_{1i-1}^{n}}{\Delta x} + \lambda_1^{-} \frac{w_{1i+1}^{n} - w_{1i}^{n}}{\Delta x} = 0$$

$\Rightarrow$ in matrix form: $\dfrac{W_i^{n+1} - W_i^n}{\Delta t} + \Lambda^+ \dfrac{W_i^n - W_{i-1}^n}{\Delta x} + \Lambda^- \dfrac{W_{i+1}^n - W_i^n}{\Delta x} = 0$

$\Rightarrow$ in conserved variables:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + \mathsf{R} \cdot \Lambda^+ \cdot \mathsf{L} \frac{U_i^n - U_{i-1}^n}{\Delta x} + \mathsf{R} \cdot \Lambda^- \cdot \mathsf{L} \frac{U_{i+1}^n - U_i^n}{\Delta x} = 0$$

first order:

$\Rightarrow \tilde{F}_{j+1/2} = \mathsf{A}^+ U_j^n + \mathsf{A}^- U_{j+1}^n$

$$\Rightarrow \tilde{F}_{j+1/2} = \underbrace{\frac{\mathsf{A}\, U_j + \mathsf{A}\, U_{j+1}}{2}}_{\text{central, second order}} - \underbrace{\frac{1}{2} |\mathsf{A}| \left(U_{j+1} - U_j\right)}_{\text{numerical diffusion term}}$$

with $|\mathsf{A}| = \mathsf{R} \cdot |\Lambda| \cdot \mathsf{L}$

second order (away from shocks):

$$\Rightarrow \tilde{F}_{j+1/2} = \mathsf{A}^+ \left(U_j^n + \frac{1}{2} \Psi\left(\frac{U_{j+1}^n - U_j^n}{U_j^n - U_{j-1}^n}\right) \left(U_j^n - U_{j-1}^n\right)\right) + \mathsf{A}^-(\ldots)$$

$\Rightarrow$ this is the Roe scheme

- simplification: give every wave same (maximal) diffusion: Lax-Friedrichs (LF) scheme

$$\Rightarrow \tilde{F}_{j+1/2} = \underbrace{\frac{\mathsf{A}\,U_j + \mathsf{A}\,U_{j+1}}{2}}_{\text{central, second order}} \underbrace{- \frac{1}{2}\,|\lambda|_{max}\,(U_{j+1} - U_j)}_{\text{numerical diffusion term}}$$

with $|\lambda|_{max}$ the largest eigenvalue (in absolute value)

$\Rightarrow$ more simple, more robust $\Rightarrow$ good scheme, but sometimes too diffusive

- the CFL time step limitation becomes (for Roe and LF)

$$\Delta t < \frac{\Delta x}{max_{k,j}(|\lambda^{(k)}_{j+1/2}|)}$$

with k running over the number n of waves in the system

nonlinear systems

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0$$

generalize FOU for linear systems:

$$\Rightarrow \tilde{F}_{j+1/2} = \underbrace{\frac{F(U_j) + F(U_{j+1})}{2}}_{\text{central, second order}} \underbrace{- \frac{1}{2} |\tilde{\mathsf{A}}(U_j, U_{j+1})| (U_{j+1} - U_j)}_{\text{numerical diffusion term}}$$

with, e.g., $\tilde{\mathsf{A}}(U_j, U_{j+1}) = \mathsf{A}(\frac{U_j + U_{j+1}}{2})$

$\Rightarrow$ this is a simplified Roe scheme

other more sophisticated choices for $\tilde{\mathsf{A}}$ are possible (full Roe scheme uses Roe linearization, . . .)

positive second-order schemes can be obtained by taking non-linearly reconstructed values in stead of U_j and U_{j+1}

conclusions:

for 1D problems, using finite differences, the upwind idea, and nonlinear reconstruction, we obtain schemes that are

- 1: accurate: second order (except at shocks)
- 2: numerically stable (CFL for explicit, unconditionally for implicit)
- 3: positive: no oscillations at shocks
- 4: conservative: capture shocks with the right speed

$\Rightarrow$ the Lax-Friedrichs scheme is simple, robust, and thus very useful:

$$\tilde{F}_{j+1/2} = \underbrace{\frac{F(U_j) + F(U_{j+1})}{2}}_{\text{central, second order}} - \underbrace{\frac{1}{2}\,|\tilde{\lambda}|_{max}(U_j, U_{j+1})\,(U_{j+1} - U_j)}_{\text{numerical diffusion term}}$$

with $|\tilde{\lambda}|_{max}(U_j, U_{j+1}) = |\lambda|_{max}(\frac{U_j + U_{j+1}}{2})$

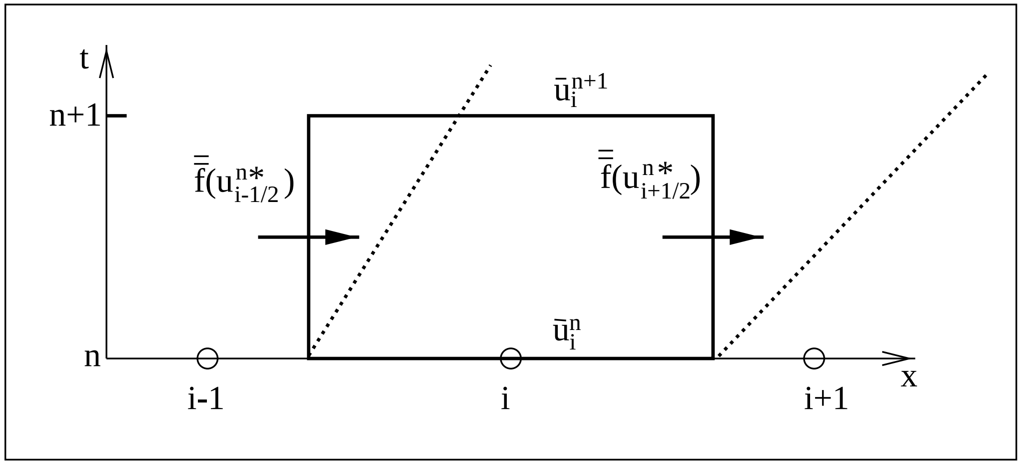
2.2 Finite volume schemes

- Finite Difference (FD) = point values
- Finite Volumes (FV) = cell averages

$\Longrightarrow$ here we will make the link between the two

2.2.1 Spatial discretization

Cell averages define Riemann problems



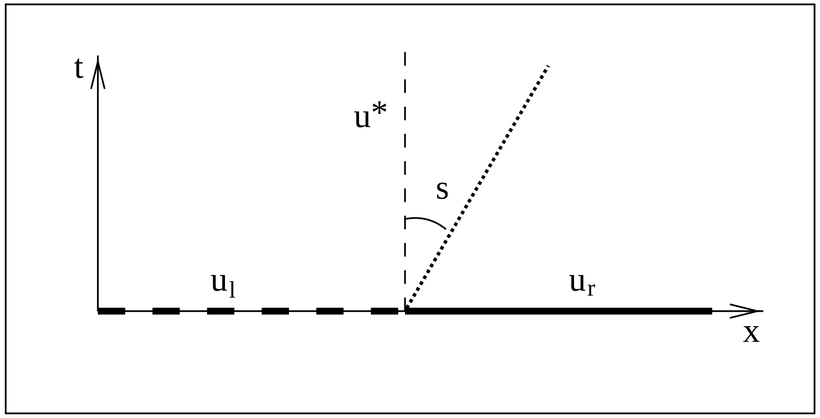
$$\boxed{\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0}$$

We integrate over a finite volume cell and obtain

$$(\overline{u}_i^{n+1} - \overline{u}_i^{n})\,\Delta x + (\overline{\overline{f}}(u^{n*}_{i+1/2}) - \overline{\overline{f}}(u^{n*}_{i-1/2}))\,\Delta t = 0$$

with $\overline{u}(t) = \int_{x_0}^{x_1} u(x,t)dx$ and $\overline{\overline{f}}(x) = \int_{t_0}^{t_1} f(u(x,t))dt$

$\Rightarrow$ at every cell interface there is a Riemann problem: find state u^*



- approximate (linearized) Riemann solver: find an expression for numerical flux function f^* (in stead of u^*)

  e.g., $f^{n*}_{i+1/2} = \frac{f(u^n_{i+1})+f(u^n_i)}{2} - \frac{1}{2}|f'^{n*}_{i+1/2}|(u^n_{i+1} - u^n_i)$

  Lax-Friedrichs, with $f'^{n*}_{i+1/2} = f'(\dfrac{u^n_i + u^n_{i+1}}{2})$
- CFL = waves from neighbouring Riemann problems do not interfere
- FD upwind concept $\sim$ FV scheme with Riemann solver

1D finite volume schemes

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}(U) = 0$$

define cell average stored in cell center of cell i as $\overline{U}_i = \left(\int_{x_{i-1/2}}^{x_{i+1/2}} U(x,t)\, dx \right) / \Delta x$

time evolution equation for this average after integration in space over the *finite volume* cell with label i

$$\frac{\partial \overline{U}_i}{\partial t} + 1/\Delta x \, (F^*_{i+1/2} - F^*_{i-1/2}) = 0$$

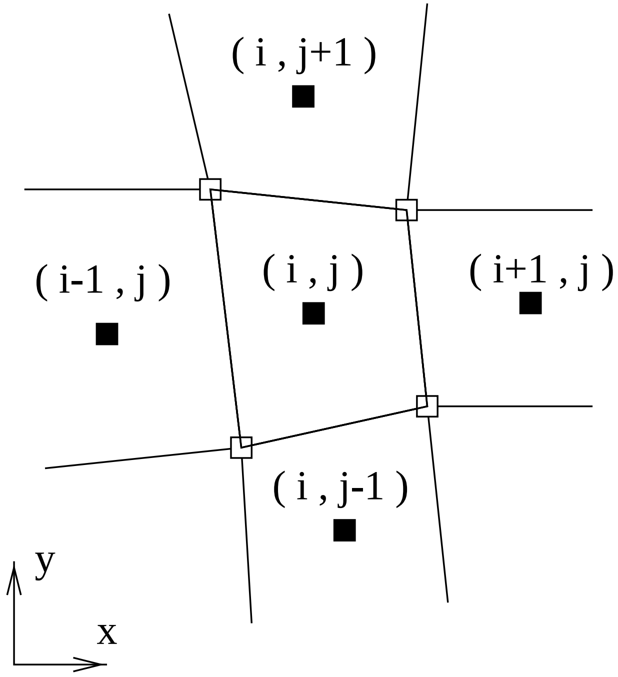
Many numerical flux functions, including Roe and Lax-Friedrichs, can be cast in the following form

$$F^*(U_l, U_r) = \frac{F(U_l) + F(U_r)}{2} + D(U_l, U_r)$$

- second order in space: use linear reconstruction with slope limiter L

$$U_l = \overline{U}_i + 1/2\, L(\overline{U}_i - \overline{U}_{i-1}, \overline{U}_{i+1} - \overline{U}_i)$$

2D finite volume schemes



define cell averages $\overline{U}_{i,j} = \left(\int\int U(x,y,t)\, dx\, dy \right) /$

and obtain for the time evolution

$$\frac{\partial \overline{U}_{i,j}}{\partial t} + 1/\Omega_{i,j} \sum_{k=1}^{4} \vec{F}_k^* \cdot \vec{n}_k \, \Delta l_k = 0$$

- 3D: analogous

Numerical flux functions

- (approximate) Roe:

$$F^*(U_l, U_r) = \frac{F(U_l) + F(U_r)}{2} - |\mathsf{A}^*(U_j, U_{j+1})| \cdot \frac{U_r - U_l}{2}$$

with $\mathsf{A}^*(U_j, U_{j+1}) = \mathsf{A}(\frac{U_j + U_{j+1}}{2})$

- (local) Lax-Friedrichs:

$$F^*(U_l, U_r) = \frac{F(U_l) + F(U_r)}{2} - (|v_n^*| + c_n^{f*})\frac{U_r - U_l}{2}$$

2.2.2 Boundary conditions

Ingoing and outgoing characteristics

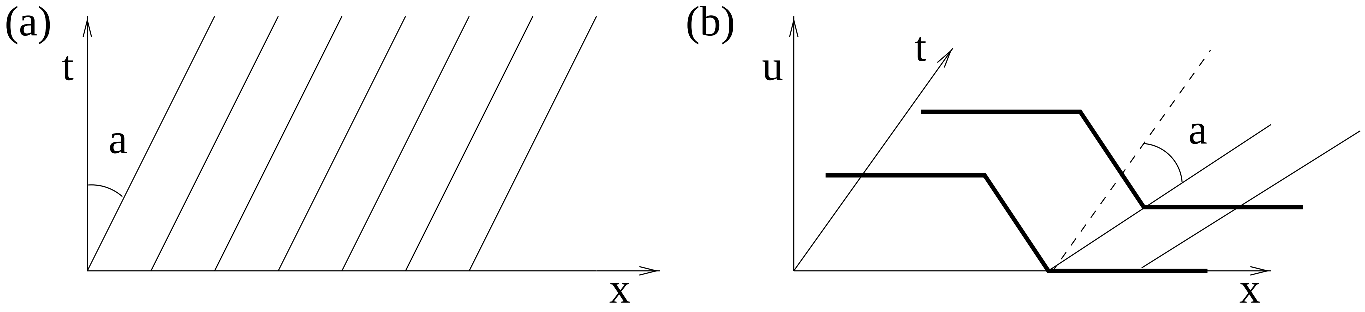
- hyperbolic system: wave perturbations propagate along characteristics

$\Rightarrow$ at domain boundary:

- impose wave perturbations along characteristics going into the domain as boundary conditions

- extrapolate from computational domain wave perturbations along characteristics coming out from the domain

- e.g., linear advection

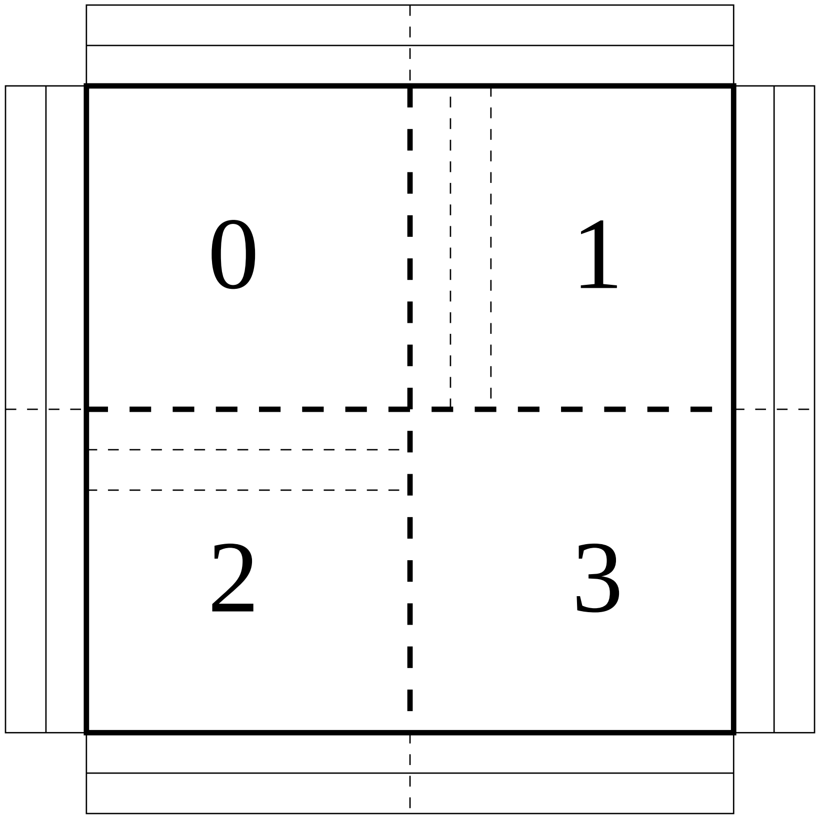


- nonlinear system: complication: nonlinear local relation between wave perturbations and conserved variables (Riemann Invariants do generally not exist)

  $\Rightarrow$ at least take correct number of imposed and extrapolated quantities based on counting ingoing characteristics

  $\Rightarrow$ how to decide which conservative variables to impose: trial and error, guided by physical intuition

Ghost cell approach



2.2.3 Temporal discretization

Two-stage Runge-Kutta scheme

$$\frac{\partial \overline{U}_{i,j}}{\partial t} = R_{i,j}^{(1),(2)}(\overline{U})$$

with $R_{i,j}^{(1),(2)}$ a first or second order accurate discretization of the *residual* in cell (i, j)

- first order in time:

$$\overline{U}_{i,j}^{t+\Delta t} = \overline{U}_{i,j}^{t} + R_{i,j}^{(1)}(\overline{U}^{t})\,\Delta t$$

- second order in time: use two-stage Runge-Kutta:

$$\overline{U}_{i,j}^{*} = \overline{U}_{i,j}^{t} + R_{i,j}^{(2)}(\overline{U}^{t})\,\Delta t/2$$

$$\overline{U}_{i,j}^{t+\Delta t} = \overline{U}_{i,j}^{t} + R_{i,j}^{(2)}(\overline{U}^{*})\,\Delta t$$

- remark: LW: combine temporal and spatial discretization to achieve second order in one step (cancellation of error terms)

- time step Δt is derived from the following CFL-like time step limitation

$$\Delta t = c_{CFL} \min_{i,j}\left[\frac{\Omega_{i,j}}{\sum_{k=1}^{4} \max(0, (\vec{v}_{i,j} \cdot \vec{n}_k + c^f_{k,i,j}))\, \Delta l_k}\right]$$

conclusion

- FV Riemann $\sim$ FD upwind
- $\dfrac{\partial \overline{U}_{i,j}}{\partial t} + 1/\Omega_{i,j} \displaystyle\sum_{k=1}^{4} \vec{F}^*_k \cdot \vec{n}_k \, \Delta l_k = 0$
- good choice of numerical flux function:

  1. accurate – low diffusion (second order away from shocks)
  2. numerically stable (CFL or implicit)
  3. positive – no oscillations (at shocks)
  4. conservative – capture shocks with right speed

2.3 Example: 2D scalar problem on Cartesian grid

The problem – the solution

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{h}(u) = 0$$

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{u^2}{2}}{\partial x} + \frac{\partial u}{\partial y} = 0$$

fluxes $f(u) = u^2/2$ and $g(u) = u$, $\vec{h}(u) = (f(u), g(u))$)

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

$$\frac{\partial u}{\partial t} + (u, 1) \cdot (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = 0$$

- domain: $(x, y) \in [0, 1] \times [0, 1]$
- initial condition: $u(x, y) \equiv 0$

- boundary conditions: impose value where $\vec{\lambda} \cdot \vec{n}$ goes into domain (locally 1D)

  choose bottom: u = 1.5 -> -0.5 linearly

$\Rightarrow$ left: impose u = 1.5 (ingoing characteristic)

$\Rightarrow$ right: impose u = -0.5 (ingoing characteristic)

$\Rightarrow$ top: u is free (outgoing characteristics)

- discretization:

$$\frac{\partial \overline{u}_{i,j}}{\partial t} + 1/\Omega_{i,j} \sum_{k=1}^{4} \vec{h}_k^* \cdot \vec{n}_k \, \Delta l_k = 0$$

  choose regular Cartesian grid (Δx,Δy):

$$\frac{\partial \overline{u}_{i,j}}{\partial t} + 1/\Omega_{i,j} \left((f^*_{i+1/2,j} - f^*_{i-1/2,j})\Delta y + (g^*_{i,j+1/2} - g^*_{i,j-1/2})\Delta x \right) = 0$$

$$\frac{\partial \overline{u}_{i,j}}{\partial t} + \frac{f^*_{i+1/2,j} - f^*_{i-1/2,j}}{\Delta x} + \frac{g^*_{i,j+1/2} - g^*_{i,j-1/2}}{\Delta y} = 0$$

  choose first order Lax Friedrichs flux function:

$$f^*{}_{i+1/2} = \frac{f(u_i) + f(u_{i+1})}{2} - \frac{1}{2}\left|a^*(u_i, u_{i+1})\right|(u_{i+1} - u_i)$$

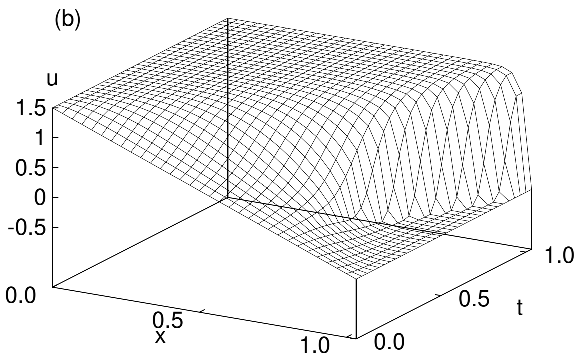
with $a^*(u_i, u_{i+1}) = a(\frac{u_i + u_{i+1}}{2})$

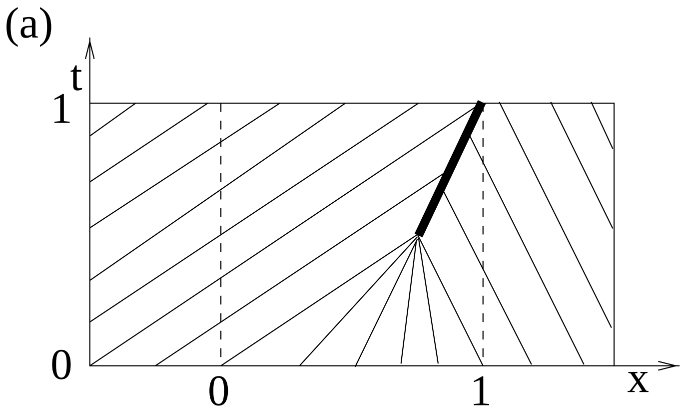
- after time evolution steady state is reached
- interpretation of steady state:

$$\frac{\partial \frac{u^2}{2}}{\partial x} + \frac{\partial u}{\partial y} = 0$$

$\Rightarrow$ this is the Burgers equation (y plays role of t), with $u_l = 1.5$ and $u_r = -0.5$

$\Rightarrow$ compressive wave steepens into shock!





A fortran90 implementation

- $u(-1 : ni + 2, -1 : nj + 2)$

  ni physical cells in i direction $(1 \dots ni)$, nj cells in j direction

  two layers of ghost cells

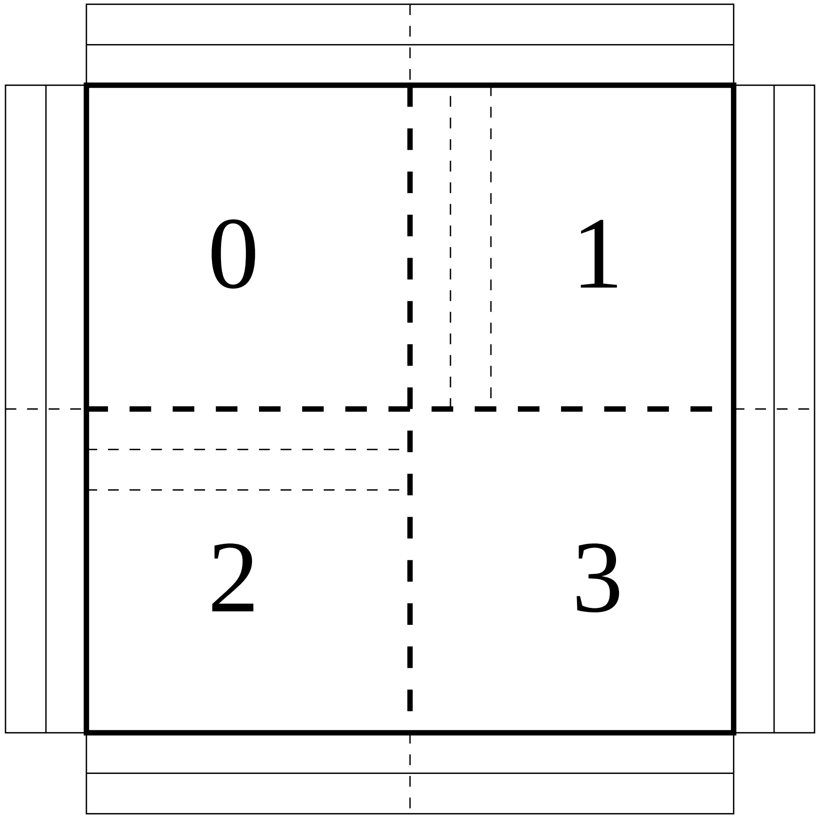
- loop

  - do boundary conditions

  - do update for new timestep

2.4 Implementation on parallel computers using MPI

Domain partitioning



MPI: Message Passing Interface

- single program, multiple processors
- MPI = library of routines (f90 or C)

⇒ commands:

- initialization and termination

```
call MPI_INIT( ierr )
call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
call MPI_ABORT( MPI_COMM_WORLD, 1, ierr)
call MPI_FINALIZE(ierr)
```

- timing

  call MPI_BARRIER(MPI_COMM_WORLD,ierr)

  t1=MPI_WTIME()

- broadcast information (tol0) from processor 0 to all the others: from one to all

  call MPI_BCAST(tol0,1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)

- reduce = from all to all with operation (sum, . . .)

  call MPI_ALLREDUCE(resid,resid0,1,MPI_DOUBLE_PRECISION,MPI_SUM,
    MPI_COMM_WORLD,ierr)

$\Rightarrow$ resid is collected from every processor, summed up, and sent to every processor in resid0

- non-blocking (do something else while waiting) send and receive

```
call MPI_ISEND(leftbufsend,2*nlj,MPI_DOUBLE_PRECISION,nbrleft,0,
  MPI_COMM_WORLD,req(1),ierr)
```

⇒ send leftbufsend of size 2*nlj to nbrleft

```
call MPI_IRECV(leftbufrec,2*nlj,MPI_DOUBLE_PRECISION,nbrleft,0,
  MPI_COMM_WORLD,req(5),ierr)
```

⇒ receive leftbufrec of size 2*nlj from nbrleft

```
call MPI_WAITALL(8,req,status_array,ierr)
```

⇒ wait until all sends and receives (info stored in status arrays) have completed

2.4.1 Parallel performance – scalability

Serial bottlenecks

- execution time

  - 1 processor:

$$T_1 = \underbrace{(1-\alpha)}_{parallel} + \underbrace{\alpha}_{serial} \qquad (=1)$$

  - n processors:

$$T_n = \frac{(1-\alpha)}{n} + \alpha$$

- parallel speedup:

$$S_n = \frac{T_1}{T_n} \qquad (\quad = n \text{ ideally} \quad)$$

$$\Rightarrow S_n = \frac{n}{(1-\alpha) + n\,\alpha} < \frac{1}{\alpha}$$

$$S_n = \frac{n}{(1-\alpha) + n\,\alpha} < \frac{1}{\alpha}$$

- $\alpha = 0 \;\Rightarrow S_n = n$: perfect scaling
- $S_{max} = S_\infty = \dfrac{1}{\alpha}$: speedup curve flattens

e.g.: $\alpha = 0.1 \;\Rightarrow S_{max} = 10$

$\Rightarrow$ serial fraction α acts as bottleneck

- α is very small in our explicit algorithm (local!)
- α is large for standard implicit algorithms (global!)

  $\Rightarrow$ need to adapt algorithm to parallel execution

(Remark: S_n can become larger than n due to hardware effects (e.g. cache))

Communication overhead (for message passing)

- execution time

  - n processors:

  $$T_n = T_n^{calc} + T_n^{comm}$$

  - 1 processor:

  $$T_1 = T_n^{calc}\, n$$

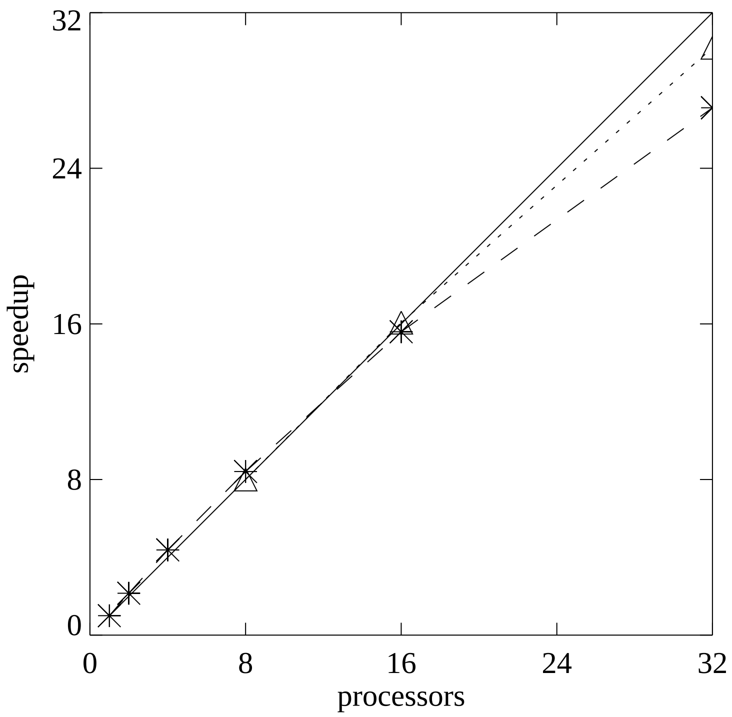
- parallel speedup:

$$S_n = \frac{T_1}{T_n} = \frac{n}{1 + \frac{T_n^{comm}}{T_n^{calc}}} = \frac{n}{1 + f_c}$$

$$f_c = \frac{T_n^{comm}}{T_n^{calc}} \sim \frac{n_x + n_y}{n_x\, n_y} \sim \frac{perimeter}{surface}$$

$\Rightarrow$ large domains (small f_c) are better, speedup curves flatten for large n

- example: 2D MHD code, Lax-Friedrichs, MPI



Speedup for the simulation of a 2D bow shock flow with a parallel MHD code on a relatively coarse grid (80×80, dashed with asterisks) and on a slightly finer grid (160×160, dotted with triangles). The theoretical speedup is given by the solid line.

2.4.2 Parallel computer architectures

- Distributed Memory (DM)
  - every processor has its own memory
  - communication over network (fast hardware or ... internet)
  - IBM SP2
  - scalable (clusters of $\sim$ 4000 Pentiums ...)
- Shared Memory (SM)
  - processors share memory
  - Alfven (sun)
  - limited number of processors, not (directly) scalable
- Hybrid (HY)
  - a cluster of SM machines

- communication over network (fast hardware or ... internet)

2.4.3 Parallellization methods

- MPI: explicit message passing

  - explicit calls to pass the messages

  - 'hard' for programmer

  - efficient (explicit control)

  - standard, portable, omnipresent

  - DM, but also SM, and best for HY

  - (machine-specific message passing libraries exist, slightly more efficient, but not portable (e.g. Cray SHMEM))

- HPF (High Performance Fortran): hidden message passing
  - compiler flags and directives
  - ‘easier’ for programmer
  - potentially much less efficient (less explicit control, ‘hard’ for compiler)
  - standard, portable, but not very widespread
  - DM, but also SM, HY

- application libraries: hidden message passing
  - (Petsc, Aztec, Cactus, ...)
  - higher level directives and calls, use MPI but hidden, for specific PDE applications
  - ‘easier’ for programmer
  - potentially less efficient, but efficient for specific problems
  - standard, portable, free
  - DM, but also SM, HY

- openMP: shared memory programming

    - compiler flags and directives

    - ‘easier’ for programmer

    - may not be very efficient (less explicit control, ‘hard’ for compiler)

    - new standard, portable, not very widespread yet

    - only SM (bad performance on DM, HY), limited number of processors

    - (machine-specific shared memory compilers exist, slightly more efficient, but not portable)