

Chapter 8: Extensions to Balance

Laws and multi-D

8.1 Source terms

Consider a hyperbolic system of the type

$$(I) \quad u_t + f(u)_x = S(u)$$

↑ geometric and/or physical source terms

Two approaches:

1) unsplit methods: direct discretization of

(I) → numerical schemes can/must be directly tailored to the specific equation

→ less general
(will not discuss further)

2) split / "operator split" / "fractional-step methods":

Split problem into two subproblems

$$(II) \left\{ \begin{array}{l} u_t + f(u)_x = 0 \\ \frac{d}{dt} u = S(u) \end{array} \right.$$

that can be solved independently in an alternating manner

→ can use optimal, existing schemes for each subproblem

→ can be easily applied to a large class and more complicated problems

and widely used in hydrodynamics codes!

$$\text{Idea: } \left. \begin{array}{l} u_t + f(u)_x = 0 \\ \text{IC: } u(x, t^n) = u^n(x) \end{array} \right\} \Rightarrow \bar{u}^{n+1}(x)$$

intermediate advection update

$$\left. \begin{array}{l} \frac{d}{dt} u = S(u) \\ \text{IC: } \bar{u}^{n+1}(x) \end{array} \right\} \Rightarrow u^{n+1}(x)$$

final update

8.1.1 Accuracy of split methods

Consider linear system and write (I) or (II) as

$$u_t = (A + S)u$$

and assume A & S do not depend on time t .

Example: $A = A \partial_x$, $S = S(x)$

Note: $\partial_t^{\frac{1}{2}} u = (A+S)^{\frac{1}{2}} u$

Theorem: The splitting error will be $\Theta(\Delta t^2)$

in general and vanish if A and S commute,
i.e. $AS = SA$.

Proof: First Taylor expand actual (exact) solution
to (I):

$$\begin{aligned} \underline{u(x, \Delta t)} &= u(x, 0) + \Delta t (A + S)u(x, 0) \\ &\quad + \frac{1}{2} (\Delta t)^2 (A + S)^2 u(x, 0) + \dots \\ &= (1 + \Delta t (A + S) + \frac{1}{2} \Delta t^2 (A + S)^2 + \dots) u(x, 0) \end{aligned}$$

$$= \sum_{k=0}^{\infty} \frac{\Delta t^k}{k!} (A+S)^k u(x,0)$$

$$= e^{\Delta t(A+S)} u(x,0)$$

Split method:

$$\bar{u}(x, \Delta t) = e^{\Delta t A} u(x, 0)$$

$$u^S(x, \Delta t) = e^{\Delta t S} \bar{u}(x, \Delta t) = e^{\Delta t S} e^{\Delta t A} u(x, 0)$$

and splitting error:

$$u(x, \Delta t) - u^S(x, \Delta t) = (e^{\Delta t(A+S)} - e^{\Delta t S} e^{\Delta t A}) u(x, 0)$$

$$\underline{u^S(x, \Delta t)} \stackrel{\text{Taylor}}{=} (1 + \Delta t S + \frac{1}{2}(\Delta t)^2 S^2 + \dots) (1 + \Delta t A + (\Delta t)^2 A^2 + \dots) \\ \times u(x, 0)$$

$$= (1 + \Delta t(A+S) + \frac{1}{2}(\Delta t)^2(A^2 + 2SA + S^2) + \dots) u(x, 0)$$

$$\Rightarrow \underline{\text{Splitting Error}}: \quad \left(\begin{array}{l} \text{Note } (A+S)^2 = (A+S)(A+S) \\ \qquad \qquad \qquad = A^2 + AS + SA + S^2 \end{array} \right)$$

$$\Delta \varepsilon^S \equiv u(x, \Delta t) - u^S(x, \Delta t)$$

$$= \frac{1}{2}(\Delta t)^2 (AS - SA) u(x, 0) + O(\Delta t^3)$$

□

- Remarks:
- 1) split methods are exact for linear problems if A & S commute (can show all higher error terms cancel)
 - 2) Even if the subproblems are solved exactly or to some high order $O(\Delta t)^k$, the split error introduces $\mathcal{O}(\Delta t) = \frac{1}{\Delta t} O(\Delta t)^2$ after $\frac{I}{\Delta t}$ time steps to reach some specified time T .
 - 3) Even though formally only first-order accurate (see 2)), often the coefficients of the 2nd order terms resulting from the discretization of e^{tA} and e^{tS} dominate and essentially 2nd-order accuracy is observed (in smooth regions, not for shocks where a lower order is generally observed)
 - 4) Through a slight modification, a higher-order split scheme can be obtained
 → "Strong splitting"

Strong SIAM J. Num. Anal. 5, 506 (1968)

Idea: update with

$$e^{\Delta t S} e^{\Delta t A} \rightarrow e^{\frac{1}{2}\Delta t A} e^{\Delta t S} e^{\frac{1}{2}\Delta t A}$$

to cancel $\mathcal{O}((\Delta t)^2)$ error term:

$$\begin{aligned} e^{\frac{1}{2}\Delta t A} e^{\Delta t S} e^{\frac{1}{2}\Delta t A} &= \left(\mathbb{1} + \frac{1}{2}\Delta t A + \frac{1}{8}(\Delta t)^2 A^2 + \dots \right) \\ &\quad \times \left(\mathbb{1} + \Delta t S + \frac{1}{2}(\Delta t)^2 S^2 + \dots \right) \\ &\quad \times \left(\mathbb{1} + \frac{1}{2}\Delta t A + \frac{1}{8}(\Delta t)^2 A^2 + \dots \right) \\ &= \mathbb{1} + \Delta t (A + S) + \frac{1}{2}(\Delta t)^2 (A^2 + AS + SA + S^2) \\ &\quad + \mathcal{O}((\Delta t)^3) \\ \Rightarrow \Delta \varepsilon &= \mathcal{O}((\Delta t)^3) \end{aligned}$$

8.1.2 ODE solver and timestepping

$$\frac{d}{dt} u = S(u)$$

first-order, non-linear
system (in general)

Note: 1) generally want to use a numerical scheme
that is at least 2nd-order accurate, in order
to preserve accuracy of split method (see Sec. 8.1.1)

2) Choice of numerical method depends on properties of S :

stability: the ODE system is called asymptotically stable if $|u_1(t) - u_2(t)| \xrightarrow{t \rightarrow \infty} 0$ for $u_1(0)$ and $u_2(0)$ sufficiently close. This is the case for

$$\operatorname{Re}(\lambda_k) < 0 \quad \forall k = 1, \dots, m$$

where $\lambda_1, \dots, \lambda_m$ are the eigenvalues of the Jacobian

$$J(u) = \frac{\partial S}{\partial u} = \begin{pmatrix} \frac{\partial S_1}{\partial u_1} & \cdots & \frac{\partial S_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial S_m}{\partial u_1} & \cdots & \frac{\partial S_m}{\partial u_m} \end{pmatrix}$$

Stiffness: disparate time scales
 \rightarrow rapid and slow variations

Def: (i) $\operatorname{Re}(\lambda_k) < 0 \quad \forall k = 1, \dots, m$

(ii) $\lambda_{\max} \equiv \max_k |\operatorname{Re}(\lambda_k)| \gg \lambda_{\min} \equiv \min_k |\operatorname{Re}(\lambda_k)|$

Stiffness ratio of $R_{\text{stiff}} = \frac{\lambda_{\max}}{\lambda_{\min}} \gtrsim 20$ can

already be problematic for explicit schemes.

Example: $u'(t) = \lambda u(t)$

Euler method: $u^{n+1} = u^n + \Delta t S(t^n, u^n)$ (explicit)

Trapezoidal method: $u^{n+1} = u^n + \frac{1}{2} \Delta t [S(t^n, u^n) + S(t^{n+1}, u^{n+1})]$
(implicit)

\Rightarrow Euler : $u^{n+1} = (1 + \lambda \Delta t) u^n$

stability requires $J = \frac{\partial S}{\partial u} = \lambda < 0$

and (solution remains bounded) $(1 + \lambda \Delta t) \leq 1$

$$\Rightarrow \Delta t \leq \frac{2}{|\lambda|}$$

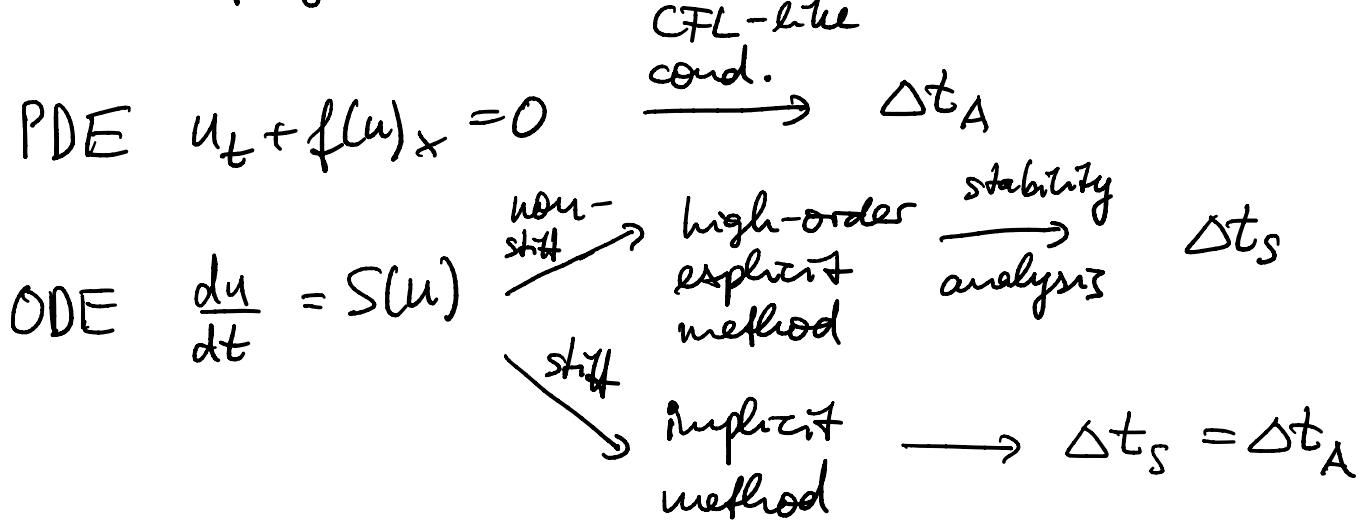
\Rightarrow for stiff ODE (large $|\lambda|$), Δt becomes impractically small!

Trapezoidal: $u^{n+1} = \frac{(1 + \frac{1}{2} \Delta t \lambda)}{1 - \frac{1}{2} \Delta t \lambda} u^n$
 ≤ 1

for $\lambda \leq 0$

\Rightarrow unconditionally stable for any Δt
whenever ODE itself is stable

3) Timestepping:



If $\Delta t_S \geq \Delta t_A \Rightarrow$ set $\Delta t = \Delta t_A = \Delta t_S$
and evolve both subsystems
with Δt

$\Delta t_S < \Delta t_A$; advance ODE in l timesteps

$$\Delta t_l = \frac{\Delta t_A}{l} < \Delta t_S$$

advance PDE with Δt_A

4) One-step methods: note that in general one can
not use multi-step ODE methods, as u^{n-1} or
 \bar{u}^{n-1} of the previous timestep are not suitable
to use because \bar{u}^{n-1} was obtained from u^{n-1}
by use of a different equation, i.e. the conservation
law (!)

→ only use high-order multistage one-step methods (generate intermediate values as needed to construct higher-order approximations)

e.g. explicit Runge-Kutta methods

$$\text{rk4: } K_1 = \Delta t S(t^n, u^n)$$

$$K_2 = \Delta t S\left(t^n + \frac{1}{2}\Delta t, u^n + \frac{1}{2}K_1\right)$$

$$K_3 = \Delta t S\left(t^n + \frac{1}{2}\Delta t, u^n + \frac{1}{2}K_2\right)$$

$$K_4 = \Delta t S\left(t^n + \Delta t, u^n + K_3\right)$$

$$u^{n+1} = u^n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

→ only makes use of u^n !

More on RK methods: Leveque Finite Difference Methods for ODEs & PDEs, chapter 5.7

8.2 Extension to multi-D

Here we consider the multi-dimensional hyperbolic system

$$u_t + \sum_i \vec{f}^i(u)_{x_i} = 0$$

For Euler equations (cf. Chap. 2):

$$\left. \begin{aligned} u &= \begin{pmatrix} s \\ sv_1 \\ sv_2 \\ sv_3 \\ E \end{pmatrix} \\ \vec{f}^1 &= \begin{cases} s\vec{v}, & f^1 = \begin{pmatrix} sv_1 \\ sv_1^2 + p \\ sv_1 v_2 \\ sv_1 v_3 \\ v_1(E+p) \end{pmatrix} \\ f^2 = \begin{pmatrix} sv_2 \\ sv_2 v_1 \\ sv_2^2 + p \\ sv_2 v_3 \\ v_2(E+p) \end{pmatrix}, & f^3 = \begin{pmatrix} sv_3 \\ sv_3 v_1 \\ sv_3 v_2 \\ sv_3^2 + p \\ v_3(E+p) \end{pmatrix} \\ f^i = \begin{pmatrix} sv_i \\ sv_i v_j + \delta_{ij} p \\ v_i(E+p) \end{pmatrix} & j=1,2,3 \end{cases} \end{aligned} \right\}$$

Two basic approaches:

- Dimensional Splitting
- unsplit finite volume methods ("flux differencing")

8.2.1 Dimensional splitting

Idea: introduce dimensional splitting in analogy
to operator splitting for source terms (Chap. 8.1)

$$\text{wD PDE: } u_t + f^1(u)_{x_1} = 0 \quad \left. \begin{array}{l} \\ u^n \end{array} \right\} \xrightarrow[\substack{\Delta t \\ n \text{ ID}}]{\text{evolve}} "u^{n+1/3}$$

↓

$$\text{IC: } u^n$$

$$\text{PDE: } u_t + f^2(u)_{x_2} = 0 \quad \left. \begin{array}{l} \\ u^{n+1/3} \end{array} \right\} \xrightarrow{\Delta t} "u^{n+2/3}"$$

↓

$$\text{IC: } u^{n+1/3}$$

$$\text{PDE: } u_t + f^3(u)_{x_3} = 0 \quad \left. \begin{array}{l} \\ u^{n+2/3} \end{array} \right\} \xrightarrow{\Delta t} "u^{n+1}"$$

↓

$$\text{IC: } u^{n+2/3}$$

For an explicit conservative method:

$$1) \quad u_{i,j,k}^{n+1/3} = u_{i,j,k}^n + \frac{\Delta t}{\Delta x_1} \left[\left(g^1 \right)_{i-\frac{1}{2},j,k}^n - \left(g^1 \right)_{i+\frac{1}{2},j,k}^n \right] \quad \forall j,k$$

numerical

$$\left(g^1 \right)_{i+\frac{1}{2},j,k}^n = g^1(u_{i,j,k}^n, u_{i+1,j,k}^n) \quad \begin{array}{l} \text{flux at cell} \\ \text{interface } x_{i+\frac{1}{2}} \end{array}$$

$$2) \quad u_{i,j,k}^{n+2/3} = u_{i,j,k}^{n+1/3} + \frac{\Delta t}{\Delta x_2} \left[\left(g^2 \right)_{i,j-\frac{1}{2},k}^{n+1/3} - \left(g^2 \right)_{i,j+\frac{1}{2},k}^{n+1/3} \right] \quad \forall i,k$$

$$\left(\frac{g^2}{\delta}\right)_{i,j+\frac{1}{2},k}^{n+\frac{4}{3}} = g^2 \left(u_{i,j,k}^{n+\frac{1}{3}}, u_{i,j+1,k}^{n+\frac{1}{3}} \right)$$

$$3) u_{i,j,k}^{n+1} = u_{i,j,k}^{n+\frac{2}{3}} + \frac{\Delta t}{\Delta x_3} \left[\left(\frac{g^3}{\delta}\right)_{i,j,k-\frac{1}{2}}^{n+\frac{2}{3}} - \left(\frac{g^3}{\delta}\right)_{i,j,k+\frac{1}{2}}^{n+\frac{2}{3}} \right] \quad \forall i, j$$

Implementation: have one routine that does it all

- x sweep: call update_1D($S, v_1, v_2, v_3, p, \Delta t, \Delta x_1, \dots$)
- y sweep: call update_1D($S, v_2, v_1, v_3, p, \Delta t, \Delta x_2, \dots$)
- z sweep: call update_1D($S, v_3, v_2, v_1, p, \Delta t, \Delta x_3, \dots$)

Remark: special care for intercell fluxes
must be taken when using flux-limiter
type schemes (WAF, PLIC, ...), see Toro Sec. 16.3.3

Time-stepping:

$$\Delta t = C_{CFL} \times \min_{i,j,k} \left\{ \frac{(\Delta x_1)_{i,j,k}}{S_{i,j,k}^{n,x_1}}, \frac{(\Delta x_2)_{i,j,k}}{S_{i,j,k}^{n,x_2}}, \frac{(\Delta x_3)_{i,j,k}}{S_{i,j,k}^{n,x_3}} \right\}$$

$S_{i,j,k}^{n,x_i}$: largest signal speed at $t=t^n$, $x = (x_i, x_j, x_k)$
in direction x_i

For Euler eqns: $S_{i,j,k}^{n,x_i} = |(v_i)_{i,j,k}^n| + (c_s)_{i,j,k}^n$
fluid velocity in x_i -direction

In terms of "augmented" 1D evolution operators:

$$u^{n+1} = X_3^{(\Delta t)} X_2^{(\Delta t)} X_1^{(\Delta t)} u^n \quad (\text{1st order accurate})$$

$$u^{n+1} = X_1^{\left(\frac{\Delta t}{2}\right)} X_2^{\left(\frac{\Delta t}{2}\right)} X_3^{(\Delta t)} X_2^{\left(\frac{\Delta t}{2}\right)} X_1^{\left(\frac{\Delta t}{2}\right)} u^n \quad (\text{2nd order accurate})$$

Remark: Accuracy estimates rely on Taylor series expansion (see also Chap. 8.1)

→ Schemes are strictly valid only for smooth solutions

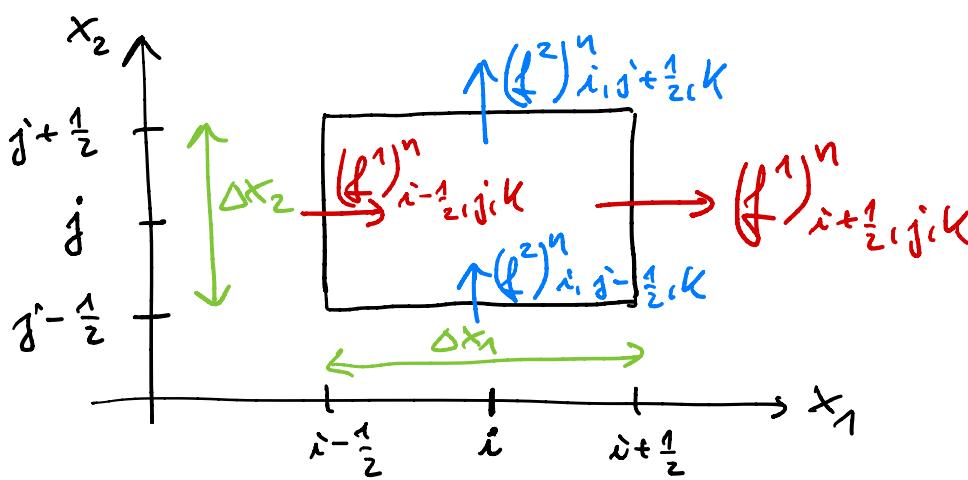
Disadvantage: For discontinuous solutions, splitting schemes may lead to erroneous results → need schemes that tend to "smear out" discontinuities

8.2.2 Unsplit finite volume schemes

and account for combined flux contributions in a single step

Consider explicit 3D finite volume scheme:

projection onto $x_1 - x_2$ plane



Finite volume
discretization
of Cartesian
domain

$$\Delta V_{i,j,k} = (\Delta x_1)_{i,j,k} \times (\Delta x_2)_{i,j,k} \times (\Delta x_3)_{i,j,k}$$

$$U_{i,j,k}^{n+1} = U_{i,j,k}^n + \frac{\Delta t}{\Delta x_1} \left[g^1_{i-1/2,j,k} - g^1_{i+1/2,j,k} \right] + \frac{\Delta t}{\Delta x_2} \left[g^2_{i,j-1/2,k} - g^2_{i,j+1/2,k} \right] \\ + \frac{\Delta t}{\Delta x_3} \left[g^3_{i,j,k-1/2} - g^3_{i,j,k+1/2} \right]$$

$U_{i,j,k}^n$: cell average, often assigned to centre of cell no "cell-centred methods"

Simplest scheme: Godunov (1st order)

and apply fluxes across each intercell boundary
in exact same way as for 1D problems

$$u_{i,j,k}^n \approx \frac{1}{\Delta V_{i,j,k}} \int u(x_1, x_2, x_3, t) dx_1 dx_2 dx_3$$

$$\Delta V_{i,j,k}$$

$$g_{i+\frac{1}{2}, j, k}^1 = f(u_{i+\frac{1}{2}, j, k}(0)), \quad g_{i, j+\frac{1}{2}, k}^2 = f^2(u_{i, j+\frac{1}{2}, k}(0))$$

$$g_{i, j, k+\frac{1}{2}}^3 = f^3(u_{i, j, k+\frac{1}{2}}(0)), \text{ where}$$

- $u_{i+\frac{1}{2}, j, k}(\frac{x_1}{\Delta})$ is the solution to the Riemann problem

$$u_t + f'(u)_{x_1} = 0$$

$$u(x_1, 0) = \begin{cases} u_{i,j,k}^n & , x_1 < 0 \\ u_{i+1,j,k}^n & , x_1 > 0 \end{cases}$$

- $u_{i, j+\frac{1}{2}, k}(\frac{x_2}{\Delta})$ is the solution to the Riemann problem

$$u_t + f^2(u)_{x_2} = 0$$

$$u(x_2, 0) = \begin{cases} u_{i,j,k}^n & , x_2 < 0 \\ u_{i,j+1,k}^n & , x_2 > 0 \end{cases}$$

- $u_{i,j,k+\frac{1}{2}}(\frac{x_3}{t})$ is the solution to the Riemann problem

$$u_t + f'(u)_{x_3} = 0$$

$$u(x_3, 0) = \begin{cases} u_{i,j,k}^n & , x_3 < 0 \\ u_{i,j,k+1}^n & , x_3 > 0 \end{cases}$$

Remark: As in 1D, approximate Riemann solvers can be used for the above Riemann problems.

8.2.3 Higher-order unsplit schemes:

MUSCL-Hancock methods

Van Leer / Hancock 1977:

J. Comp. Phys. 23, 263 (1977)

modification of Godunov's

" 23, 276 (1977)

method to achieve higher accuracy

" 32, 101 (1979)

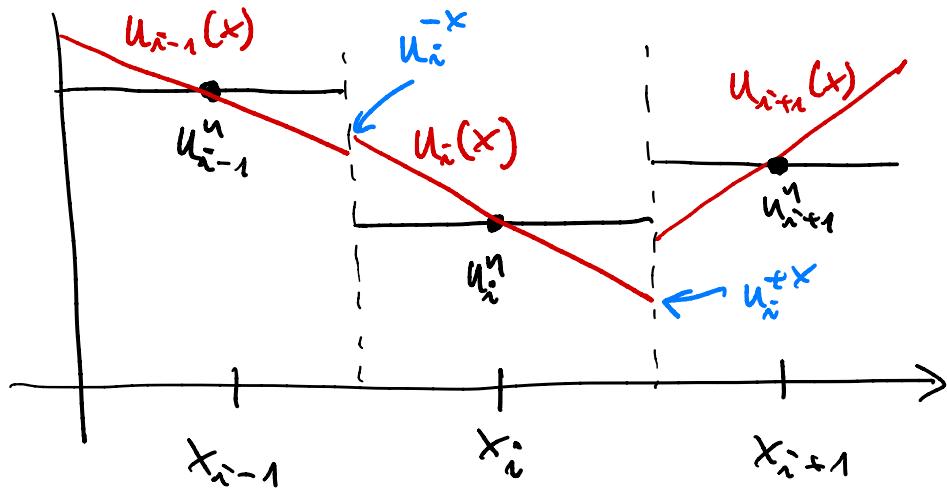
→ Variable extrapolation approach

(Monotone Upstream-centred Scheme for Conservation Laws)

Idea: piece-wise linear (or higher-order) local reconstruction

$$u_i(x) = u_i^n + \frac{(x - x_i)}{\Delta x} \Delta_i, \quad x \in [0, \Delta x]$$

↑ suitably chosen slope of
 $u_i(x)$ in cell $[x_i, x_i + \Delta x]$



REA algorithm:

(I) Reconstruction

reconstruct cell averages $u_{i,j,k}^n$ independently in x, y, z directions, select appropriate slopes $\Delta_{i,j,k}$, and obtain boundary extrapolated values (BEV)

$$u_{i,j,k}^{-x} = u_{i,j,k}^n - \frac{1}{2} \Delta_i, \quad u_{i,j,k}^{+x} = u_{i,j,k}^n + \frac{1}{2} \Delta_i$$

$$u_{i,j,k}^{-y} = u_{i,j,k}^n - \frac{1}{2} \Delta_j, \quad u_{i,j,k}^{+y} = u_{i,j,k}^n + \frac{1}{2} \Delta_j$$

$$u_{i,j,k}^{-z} = u_{i,j,k}^n - \frac{1}{2} \Delta_k, \quad u_{i,j,k}^{+z} = u_{i,j,k}^n + \frac{1}{2} \Delta_k$$

(II)(a) Evolution of BEV by half a timestep $\frac{\Delta t}{2}$:

$$\begin{aligned}\hat{u}_{i,j,k}^l &= u_{i,j,k}^l + \frac{1}{2} \frac{\Delta t}{\Delta x} \left[f^1(u_{i,j,k}^{-x}) - f^1(u_{i,j,k}^{+x}) \right] \\ &\quad + \frac{1}{2} \frac{\Delta t}{\Delta y} \left[f^2(u_{i,j,k}^{-y}) - f^2(u_{i,j,k}^{+y}) \right] \\ &\quad + \frac{1}{2} \frac{\Delta t}{\Delta z} \left[f^3(u_{i,j,k}^{-z}) - f^3(u_{i,j,k}^{+z}) \right]\end{aligned}$$

for $l = -x, +x, -y, +y, -z, +z$

(II)(b) Solution of piece-wise constant data

Riemann problem:

at each cell interface $(i+\frac{1}{2}, j)$ solve the
x/y/z-split 1D Riemann problem

$$u_t + f(u)_x = 0$$

$$u(x, 0) = \begin{cases} \hat{u}_{i,j,k}^{+x} & x < 0 \\ \hat{u}_{i+1,j,k}^{-x} & x > 0 \end{cases}$$

to find $u_{i+\frac{1}{2},j,k}(\frac{x}{t})$ (analogous for y, z).

The corresponding intercell fluxes are found as in the 1D Godunov method:

$$g_{i+\frac{1}{2}, j, k}^1 = f^1(u_{i+\frac{1}{2}, j, k}(0))$$

$$g_{i, j+\frac{1}{2}, k}^2 = f^2(u_{i, j+\frac{1}{2}, k}(0))$$

$$g_{i, j, k+\frac{1}{2}}^3 = f^3(u_{i, j, k+\frac{1}{2}}(0))$$

(III) Average $u_{i,j,k}^{n+1} = \frac{1}{\Delta x \Delta y \Delta z} \int v^n(x, t^{n+1}) dx dy dz$

Remarks: 1) This scheme is 2nd order

accurate. As for Godunov method, numerical fluxes in (II)(b) can be found using exact or approximate Riemann solvers

- 2) In order to avoid/control spurious oscillations 1D slope limiters are employed to replace $\Delta u_{i,j,k}$ by limited slopes (as cf. 1D TVD schemes)
- 3) Note that in order to obtain higher accuracy, we did not have to solve the so-called generalized Riemann problem (in split version)

$$u_t + f(u)_x = 0$$

$$u(x, 0) = \begin{cases} \hat{u}_{i,j,k}^{+x}(x), & x < 0 \\ \hat{u}_{i+1,j,k}^{-x}(x), & x > 0 \end{cases}$$

$$v^n(x, t) = u_{i,j,k}^n(\frac{x}{t})$$

Instead, the generalized RP is ignored but the BEVs are evolved by $\frac{\Delta t}{2}$, which cancels the error introduced here to linear order.

Also, one can show that the resulting scheme without the $\frac{\Delta t}{2}$ advance would be unstable.

→ essentially, one can view this as

"the reconstruction method leads to higher accuracy"

- 4) Higher-order schemes can be devised using higher-order reconstruction.

For example: piece-wise quadratz

$$u_{i,j,k}(x) = u_{i,j,k}(x) + \frac{(x-x_i)}{\Delta x} \Delta_i^{(1)} + \frac{3x}{2(\Delta x)^2} \left[\frac{(x-x_i)^2 - (\Delta x)^2}{12} \right] \Delta_i^{(2)}$$

$\Delta_i^{(1)}, \Delta_i^{(2)}$ estimates for 1st & 2nd derivative

$K = \frac{1}{3}$: 3rd order accuracy

→ the Piece-wise Parabolic Method (PPM)

(Colella & Woodward J. Comp. Phys. 54, 174
(1984))

based on similar reconstruction & 54, 115
(3rd order accurate) (1984))

Need to make sure that reconstruction is conservative, i.e. that integral of reconstruction recovers total values of the conserved variables in each cell.

Yet higher-order reconstruction using higher-order polynomials in conservative way are possible

→ e.g. ENO, WENO schemes

→ but need more cells to robustly determine coefficients of the reconstruction

→ i.e. larger stencil size required

(use least-squares fitting etc.)