Higher Order Methods

A. Schwarz, C.-D. Munz
Institute of Aerodynamics and Gasdynamics (IAG), University of Stuttgart

Preliminary: Order of Accuracy p

$$\frac{E_1}{E_2} = \left(\frac{\Delta x_1}{\Delta x_2}\right)^p \quad \Rightarrow \quad p = \frac{\log(\frac{E_1}{E_2})}{\log(\frac{\Delta x_1}{\Delta x_2})}$$

- Example: Halving grid size divides error by factor of two for $p = 1$ (O1); four for O2
- Valid for smooth problems in the limit $\Delta x \to 0$
- Mostly: Polynomial approximation of solution; $p = N + 1$ with polynomial degree N

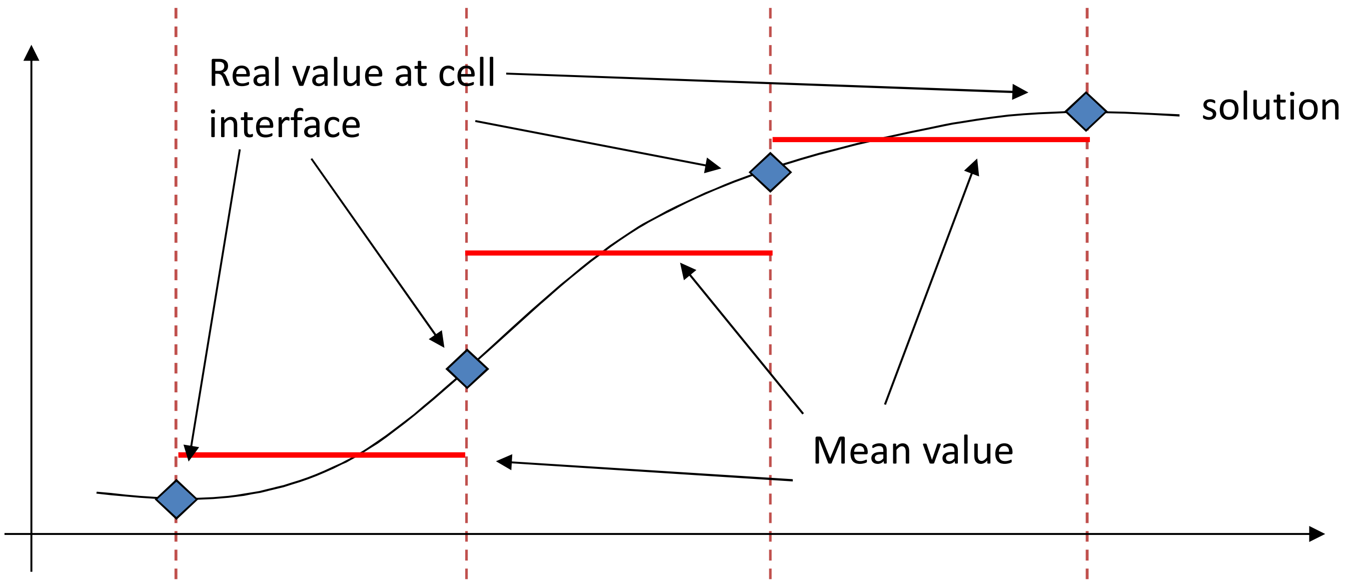
Outline

1. Motivation
2. 2nd order Finite Volume: Reconstruction, Limiting
3. Higher-order Finite Volume: ENO, WENO
4. Discontinuous Galerkin

Outline

1. Motivation
2. 2nd order Finite Volume: Reconstruction, Limiting
3. Higher-order Finite Volume: ENO, WENO
4. Discontinuous Galerkin

FV-Discretization (Smooth Problem)

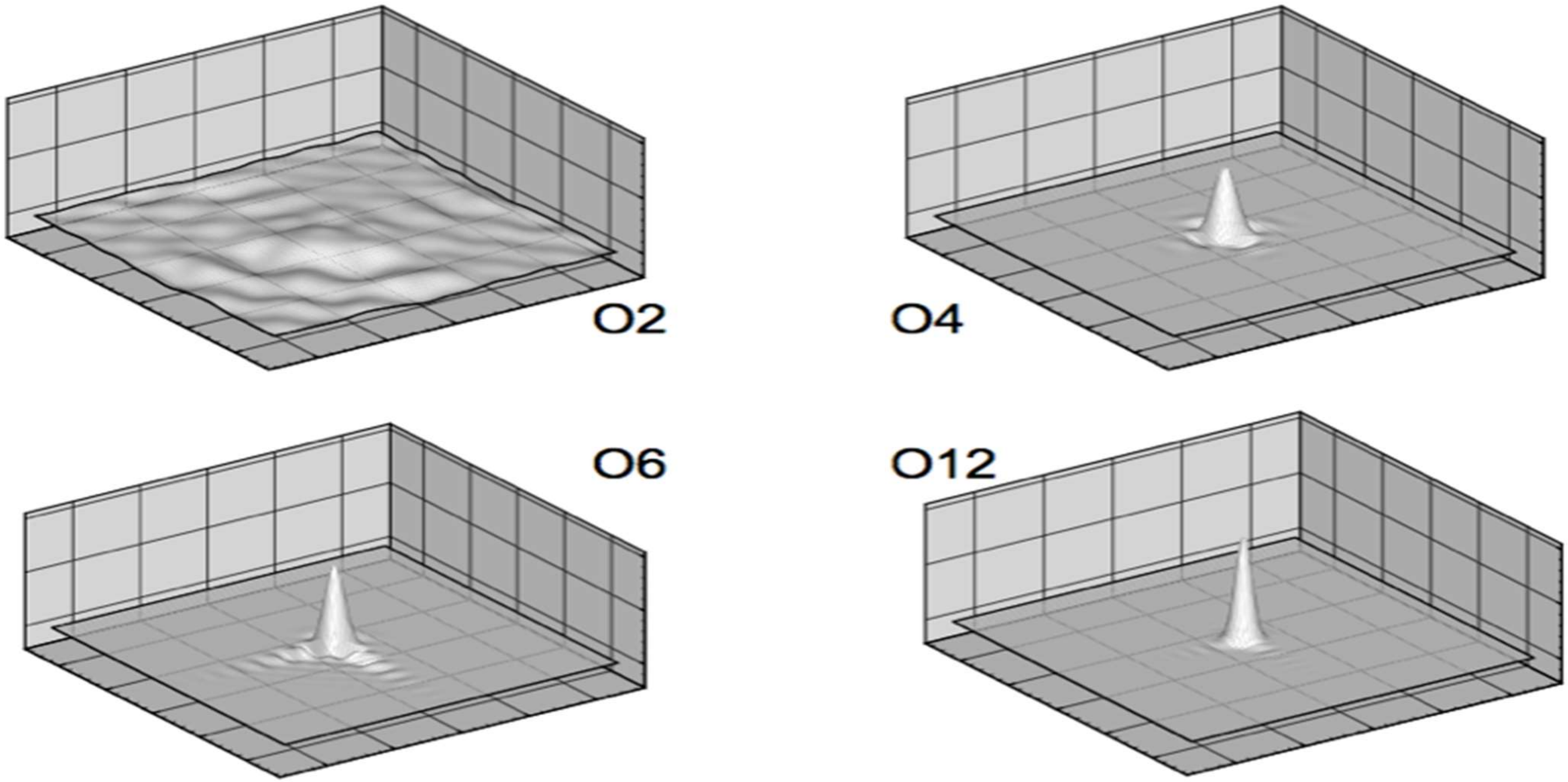


Problem: Values at cell interfaces differ from the real solution.

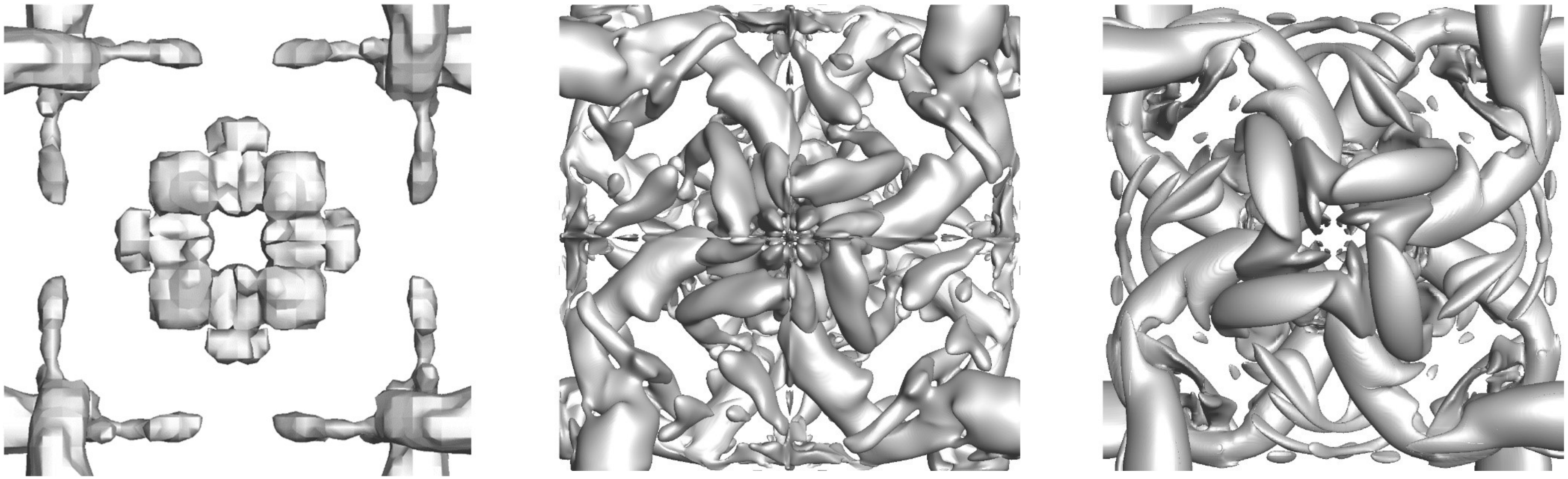
Fundamental Challenge for High Order Methods:

Get accurate/smooth solution approximation in smooth regions while capturing/retaining physical discontinuities.

Motivation-I : High Resolution/Order/Accuracy



Motivation-I : High Resolution/Order/Accuracy

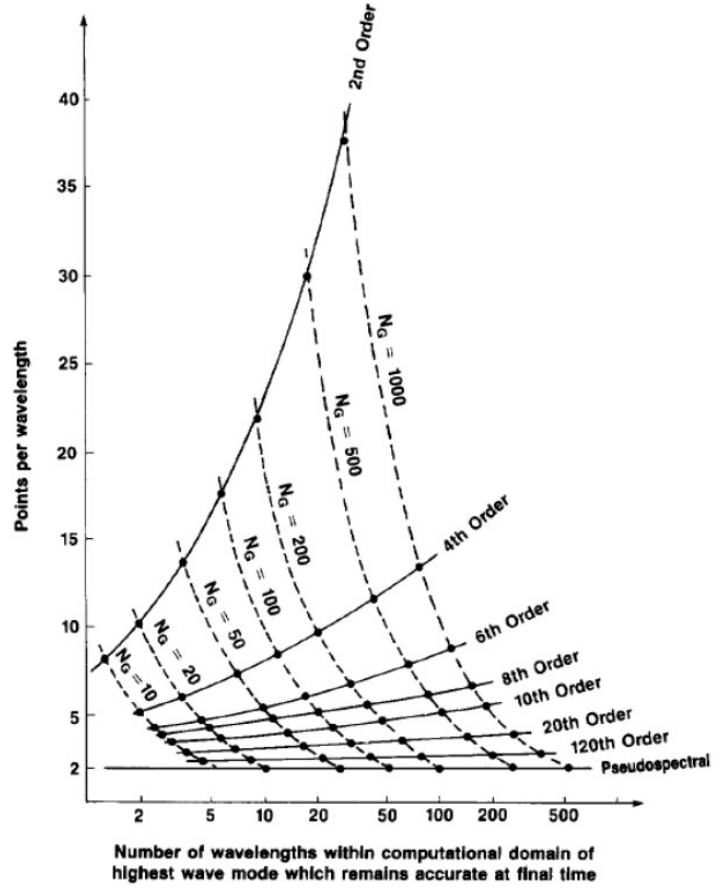


O2 – 64^3 DOF

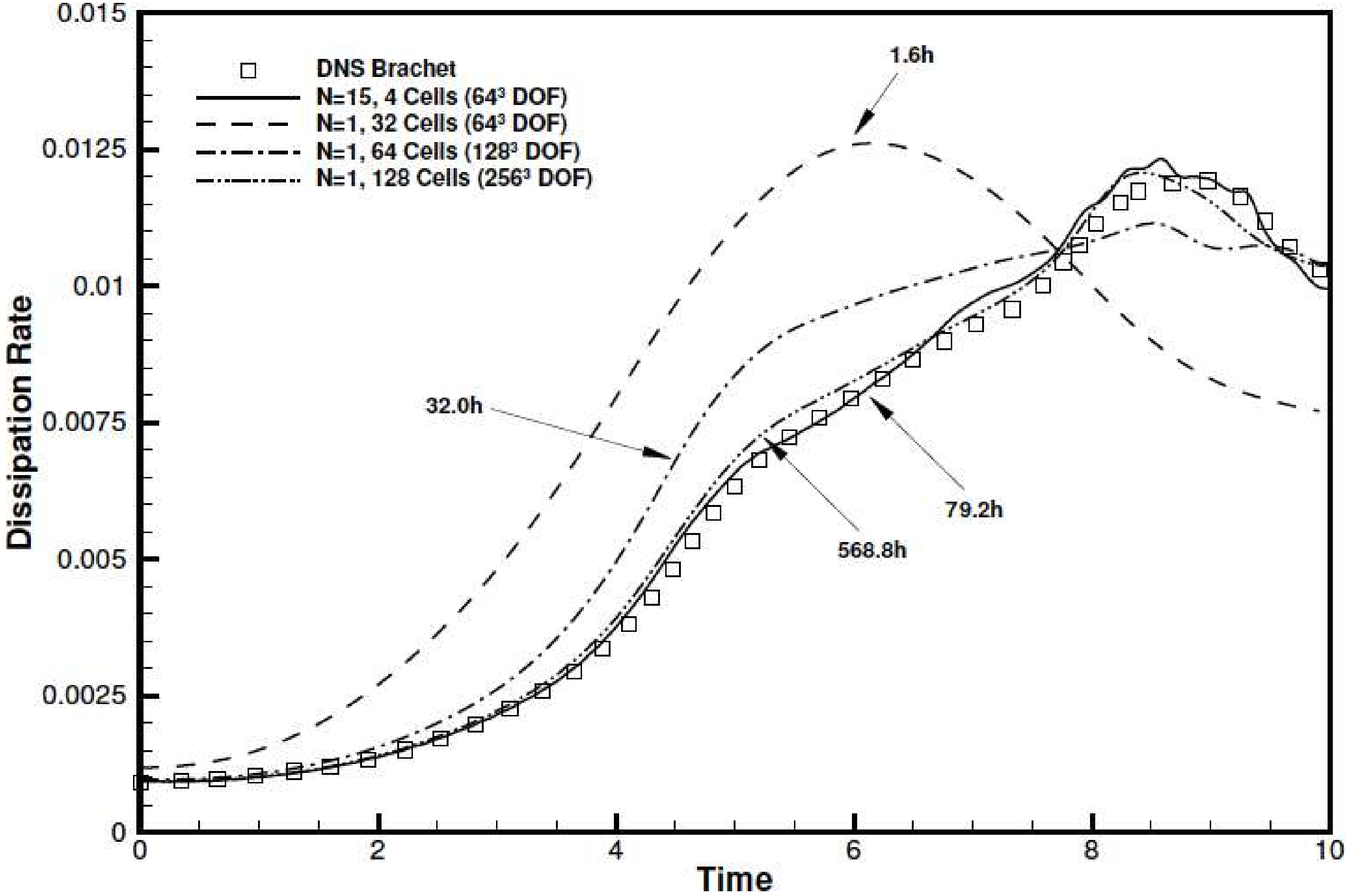
O16 – 64^3 DOF

DNS –512^3 DOF

Motivation I : High Resolution/Order/Accuracy



Motivation-II : Efficiency

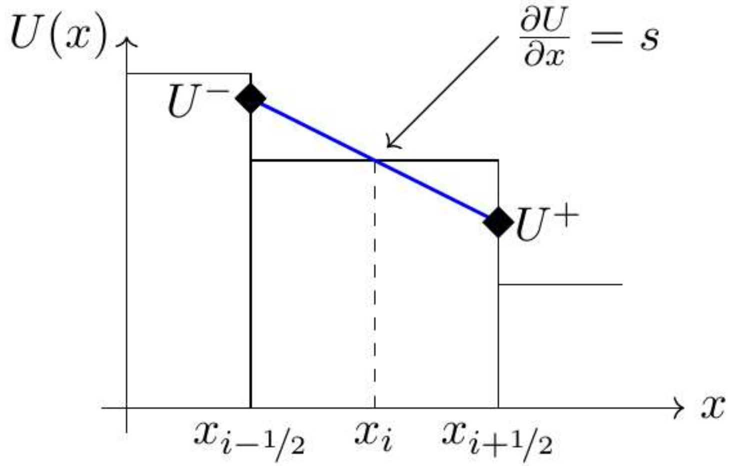


Outline

1. Motivation
2. 2nd order Finite Volume: Reconstruction, Limiting
3. Higher-order Finite Volume: ENO, WENO
4. Discontinuous Galerkin

Reconstruction in 1D

- **Ansatz:** Instead of a constant approximation in each cell, a linear distribution is used. The integral value must be preserved.



$$U(x) = U_i + s(x - x_i)$$

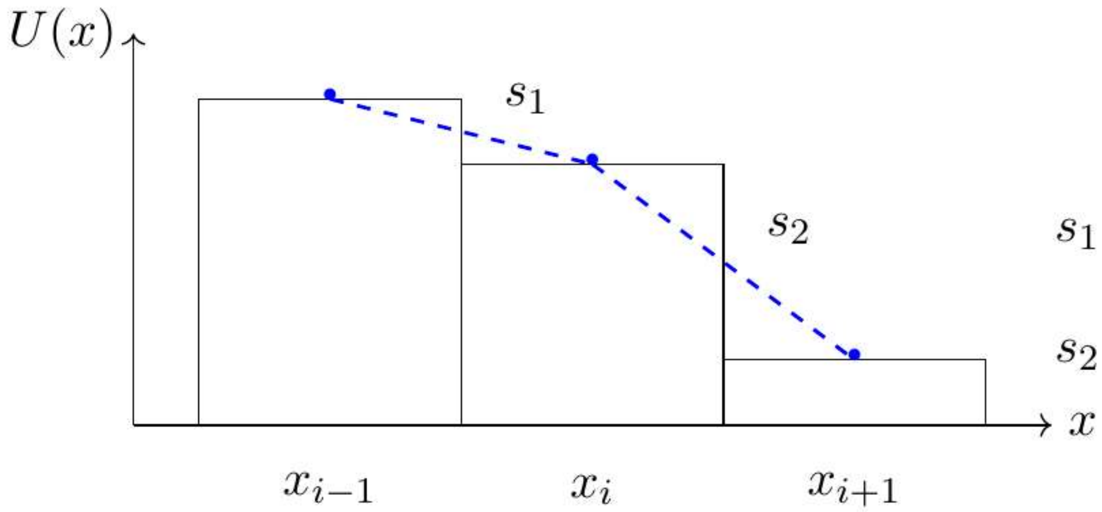
$$U_i^+ = U_i + s\frac{\Delta x}{2}$$

$$U_i^- = U_i - s\frac{\Delta x}{2}$$

- **Problem:** The FV method has no possibility to save interior cell information beside the mean value.

Reconstruction in Space: Slope Calculation

- **Process**: Only cell mean values are saved. The slope in each cell is calculated by using adjoined cells. Two neighbors allow the computation of two gradients (s_1,s_2).



$$s_1 = \frac{U_i - U_{i-1}}{\Delta x}$$

$$s_2 = \frac{U_{i+1} - U_i}{\Delta x}$$

- **Problem**: Which one is the correct gradient to reconstruct?

Reconstruction in Space: TVD-Property

- TVD-Property (Total Variation Diminishing)
- Mathematical theory for scalar conservation law in 1D
- Over time, no new extrema may be generated:

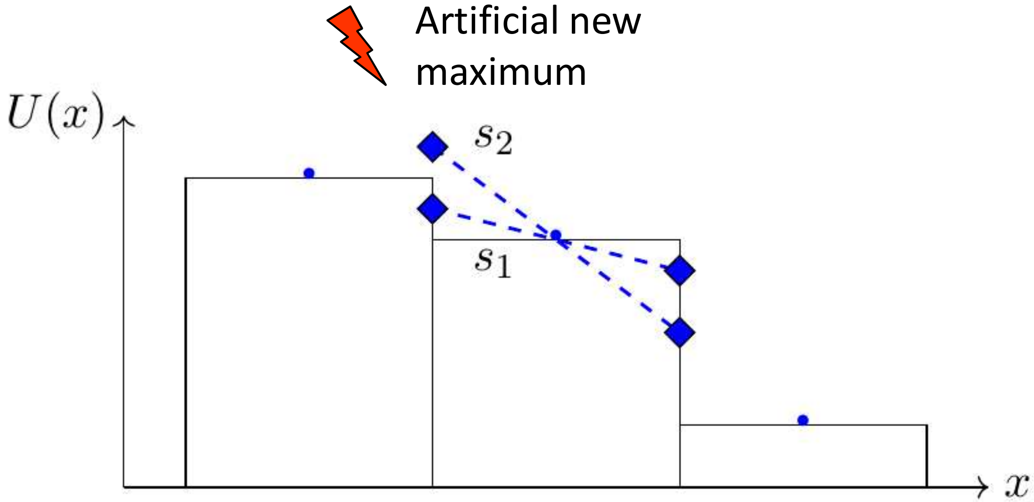
$$\sum_i \left|U_{i+1}^n - U_i^n\right| \leq \sum_i \left|U_{i+1}^0 - U_i^0\right|$$

- Sufficient condition after A. Harten (1983):

$$0 \leq \left\{ \frac{\Delta x s_i}{U_i - U_{i-1}}, \frac{\Delta x s_i}{U_{i+1} - U_i} \right\} \leq 2$$

Reconstruction in Space: TVD-Property

- The reconstructed slopes have to be limited --> Limiter



1D Limiter Functions: Examples

1. Minmod limiter (Roe, 1986):

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

$$\text{minmod}(a,b) = \begin{cases} a \text{ für } |a| < |b|, ab > 0 \\ b \text{ für } |a| \geq |b|, ab > 0 \\ 0 \text{ sonst.} \end{cases}$$

2. Sweby limiter (Sweby, 1984):

$$s(a,b;k) = \operatorname{sign}(a) \max \left\{ |\text{minmod}(a,kb)|, |\text{minmod}(ka,b)| \right\}$$

mit $1 \leq k \leq 2$.

Reconstruction in 2D / 3D

- Cartesian Grid:
  Each dimension is independent of the other. The 1D-scheme can be applied for each dimension.

- Unstructured Grid:
  More complex. Dimensions cannot be separated. More complex slope and limiter calculation.

Reconstruction on Unstructured Grids

- Reconstruction in space

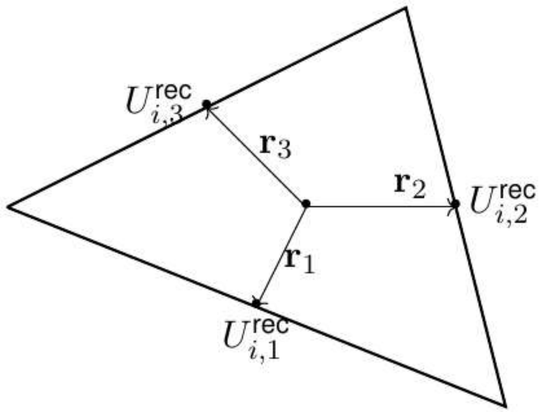
$$U_{i,m} = U_i + \nabla U_i \cdot \mathbf{r}_m$$

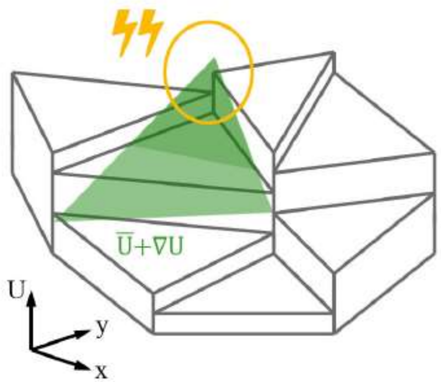
- A limiter is necessary

$$U_{i,m}^{\text{rec}} = U_i + \Psi_i \nabla U_i \cdot \vec{r}_m, \; \Psi_i \in [0,1]$$

- Barth & Jespersen limiter

$$\Psi_k = \min\left(1, \frac{\Delta_{\text{max},k}}{\Delta_{R,k}}\right), \; \Psi_k(\Delta_{R,k} = 0) = 1$$





Methods of 2nd order in Time

Two different methods are possible:

- Method of lines:
  - Separation of space and time integration
  - Time integration can be easily exchanged
  - Time and space order are generally independent
  - Easy to implement
- Space-time-expansion:
  - Taylor expansion in time
  - Less flexible, equation-dependent

Method of Lines

- The time discretization is independent of the spatial discretization.
- Calculate the spatial operator with an arbitrary scheme

$$\int_{t^n}^{t^{n+1}} u_t dt = \int_{t^n}^{t^{n+1}} -\frac{1}{V} \oint_{\partial V} F^C(U) \cdot \vec{n}\, dS dt$$
$$= \int_{t^n}^{t^{n+1}} R(u) dt$$

- Results in an ordinary differential equation, which can be integrated in time by a method for initial value problem.

Method of Lines - Implementation

- Typical methods are explicit Runge-Kutta methods

$$
\begin{aligned}
u^0 &= u^n \\
u^1 &= u^0 - \alpha_1 \Delta t R(u^n) \\
u^2 &= u^0 - \alpha_2 \Delta t R(u^1) \\
&\vdots \\
u^{n+1} &= u^0 - \alpha_m \Delta t R(u^{m-1})
\end{aligned}
$$

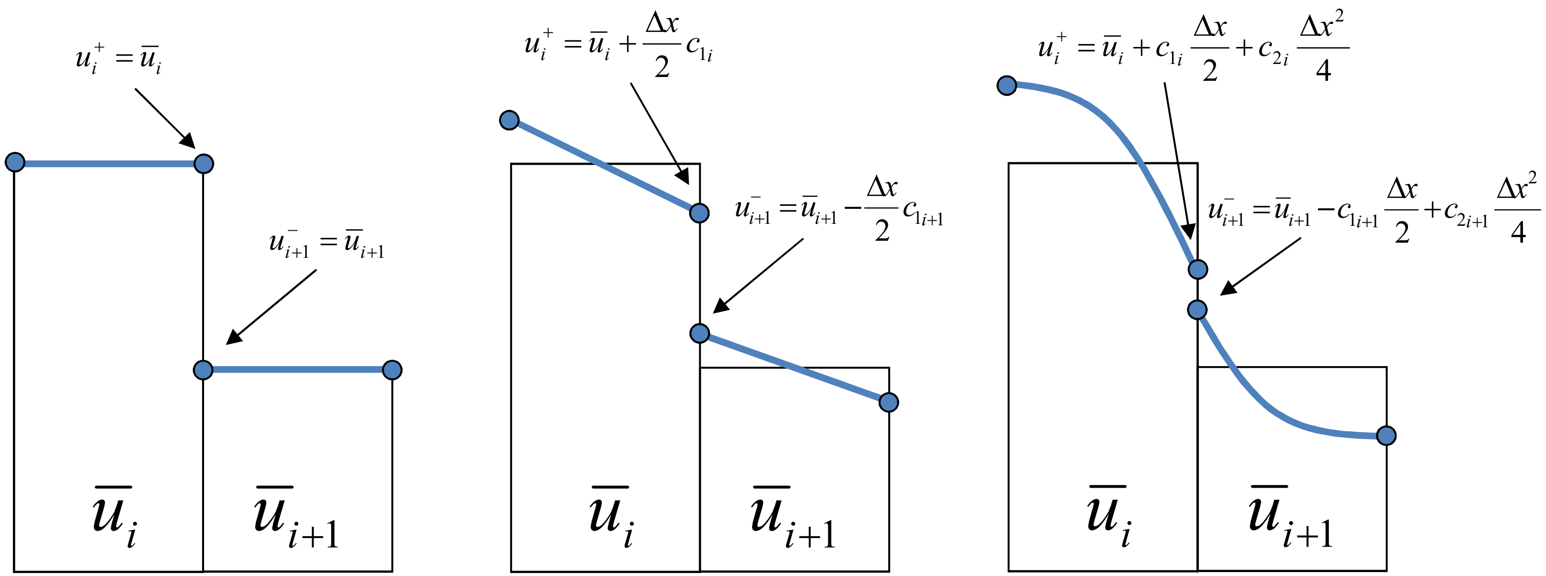
- or implicit BDF methods

$$
\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} = -R(u^{n+1}).
$$

Outline

1. Motivation
2. 2nd order Finite Volume: Reconstruction, Limiting
3. Higher-order Finite Volume: ENO, WENO
4. Discontinuous Galerkin

How to get better reconstructions?



1st order
piecewise linear

$$u_i(x) = \bar{u}_i$$

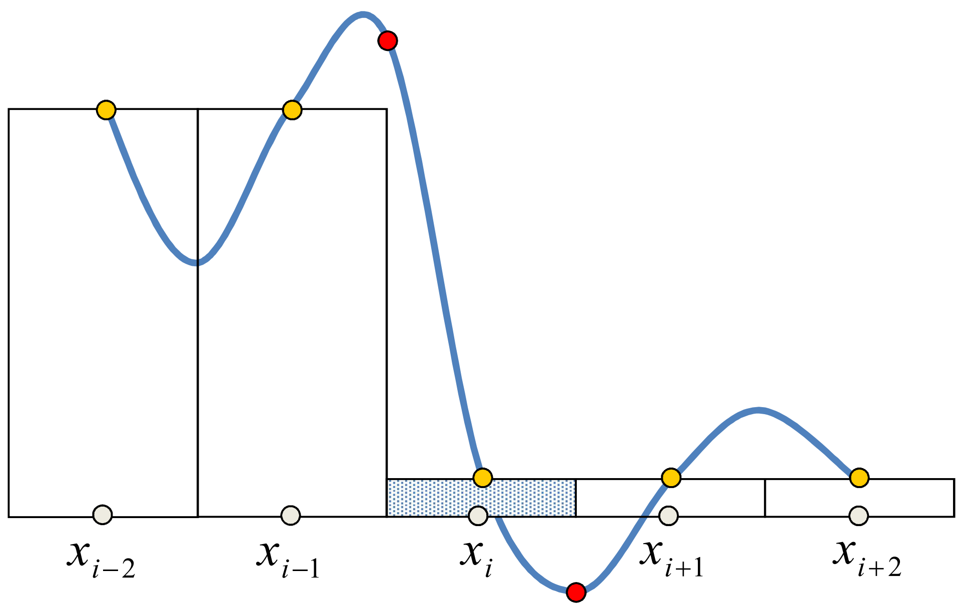
2nd order
piecewise linear

$$u_i(x) = \bar{u}_i + c_{1i}(x - x_i)$$

3rd order
piecewise quadratic

$$u_i(x) = \bar{u}_i + c_{1i}(x - x_i) + c_{2i}(x - x_i)^2$$

How Do We Get Non-Oscillating Polynomials at a Discontinuity?

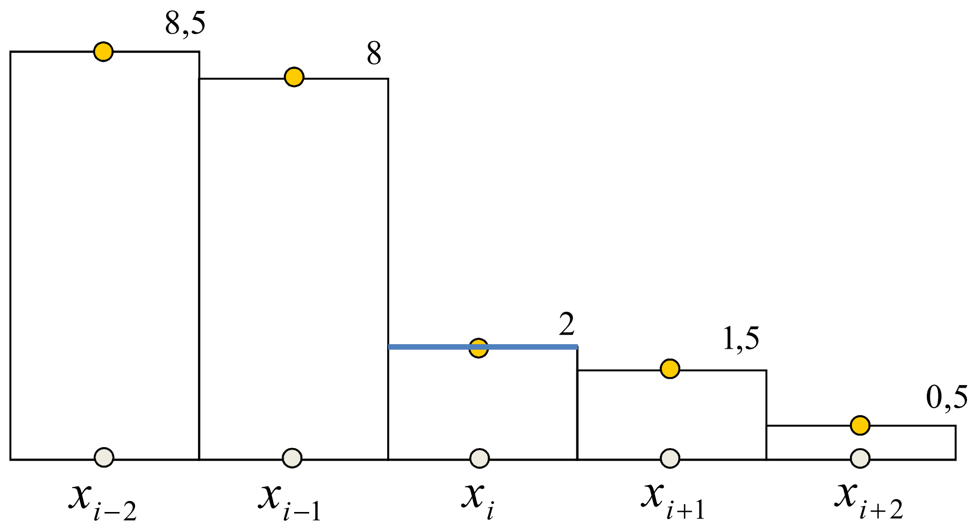


Essentially Non-Oscillatory (ENO) Reconstruction

- Add neighboring interpolation points one by one
- Thus increase the polynomial degree
- For each new point, choose between left and right:
  – Which yields the smallest change of the polynomial?
  – Which yields the smoothest part of the solution? (do not include discontinuity in the stencil)

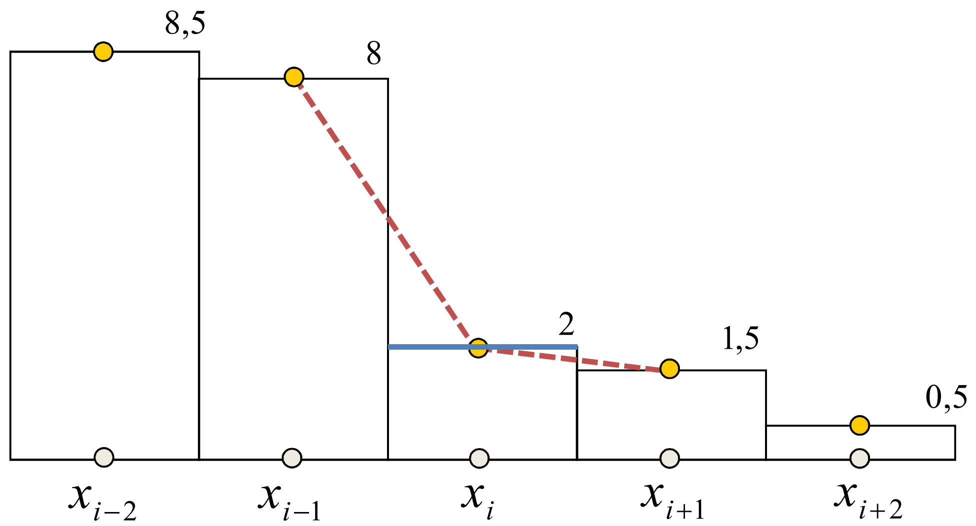
Example: ENO Reconstruction

- Constant solution u_i is 0^{th} degree polynomial



Example: ENO Reconstruction - 1st Step

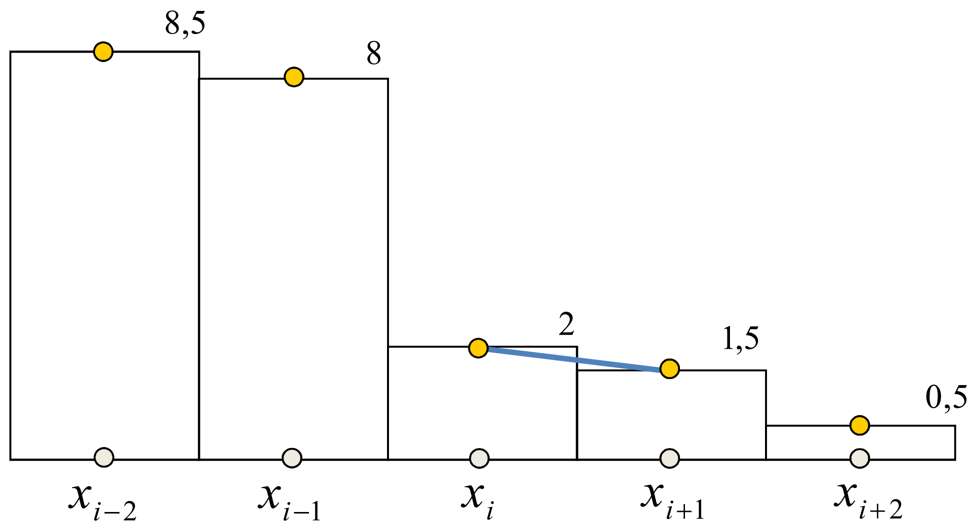
- Candidates: left or right neighbor x_{i-1} and x_{i+1}



- x_{i+1} yields smaller change compared to previous (0^{th} degree) polynomial

Example: ENO Reconstruction - 1st Step

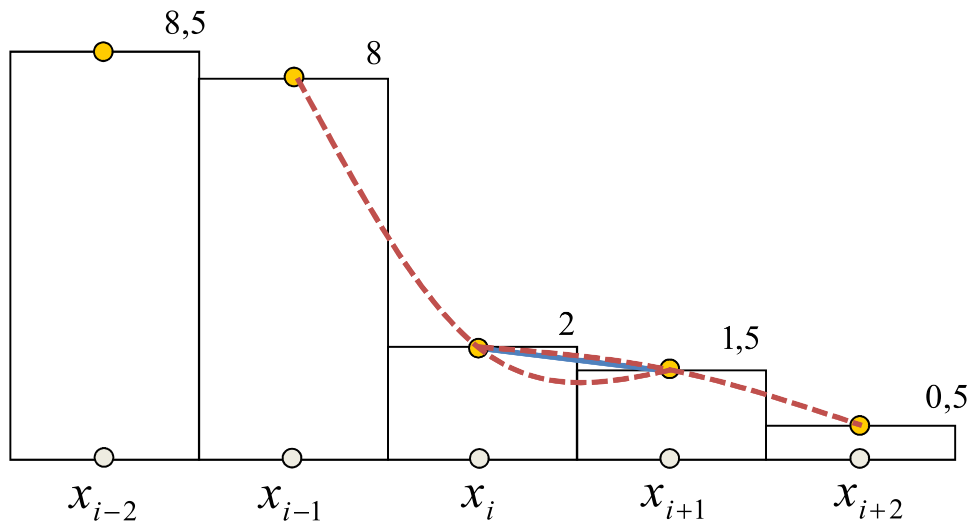
- Candidates: left or right neighbor x_{i-1} and x_{i+1}



- x_{i+1} yields smaller change compared to previous (0^{th} degree) polynomial

Example: ENO Reconstruction - 2nd Step

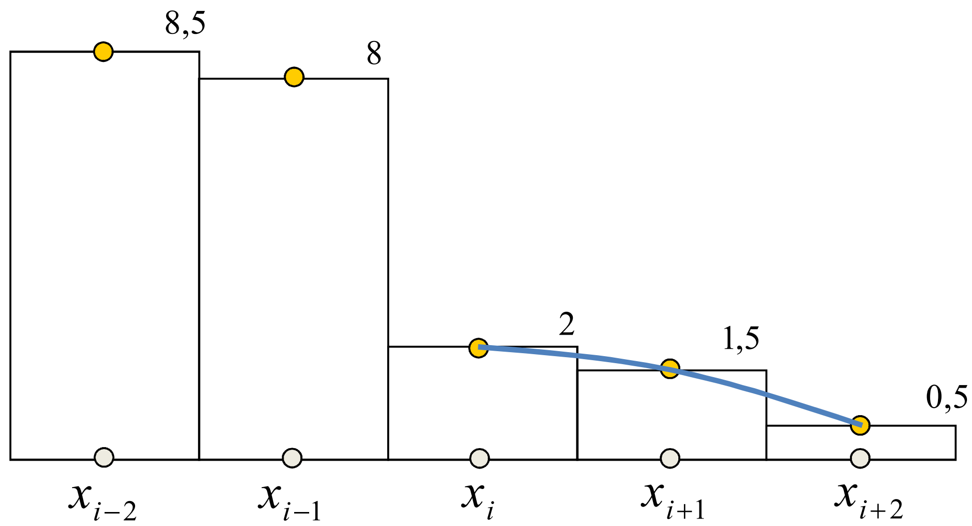
- Candidates: left or right neighbor x_{i-1} and x_{i+2}



- x_{i+2} yields smaller change compared to previous (1st degree) polynomial

Example: ENO Reconstruction - 2nd Step

- Candidates: left or right neighbor x_{i-1} and x_{i+2}



- x_{i+2} yields smaller change compared to previous (1^{st} degree) polynomial

Some Comments on the ENO Reconstruction

- Formal method: Newton polynomials (N_0=1, N_1=(x-x_0), …) , divided differences with smallest coefficients
- Stencil depends on the approximate solution
- Important is to keep the integral value

$$\int_{Q_i} p_n(x) = \overline{u}_i$$

Weighted Essentially Non-Oscillatory Reconstruction (WENO)

- Take the weighted average of the polynomials of all stencils:

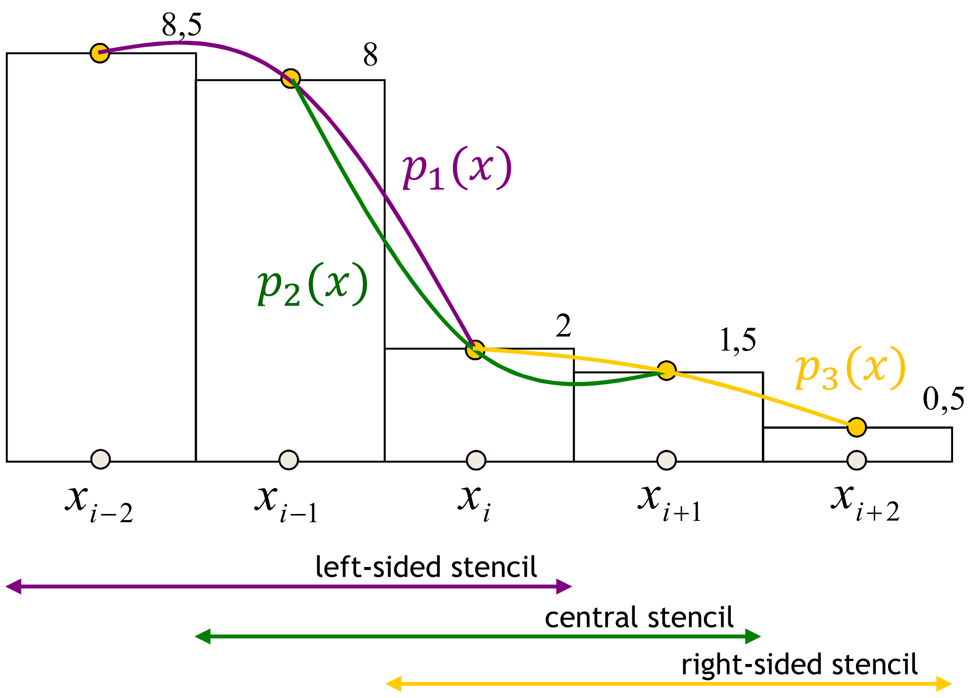
$$u^{WENO}(x) = \sum_{k=0}^{N} \omega_k p_k(x)\,,$$

N degree of polynomial
N+1 polynomials
N+1 points
w_k weights

- Adaptive weighting via an oscillation indicator

$$S_k = S(p_k) = \sum_{j=1}^{N} \Delta x^{2j-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{\partial^j}{\partial x^j} p_k(x)\right)^2 dx, \qquad \omega_k = \omega_k(\{S_j\})$$

Example: WENO Reconstruction



$$u^{WENO}(x) = \omega_1 p_1(x) + \omega_2 p_2(x) + \omega_3 p_3(x)$$

WENO: How To Get Weights From Oscillation Indicator $\omega(S)$?

- O5 stencil is linear combination of three O3 stencils:
  $\tilde{u}(x_{i+1/2}) = \sum_{k=1}^{3} \gamma_k \, p_k(x_{i+1/2})$ is O5!

- With this:

$$\tilde{\omega}_k = \frac{\gamma_k}{(S_k+\epsilon)^2}, \qquad \omega_k = \frac{\tilde{\omega}_k}{\tilde{\omega}_1+\tilde{\omega}_2+\tilde{\omega}_3}$$

➔ Smooth regions: O5! (similar S_k means $\omega_k = \gamma_k$)

➔ Unsmooth regions: Weighting strongest for least oscillating stencil

Reconstruction in Multi Dimensions

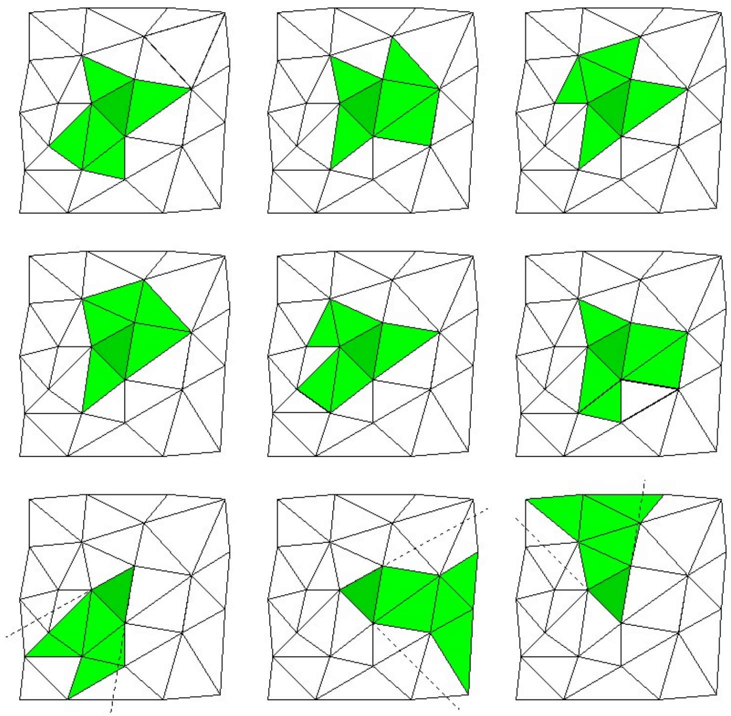
Structured Grids: Works well

- High computational effort and memory requirements
- Requirement: grid transformation is high-order accurate and smooth

Unstructured grids: Troublesome!

- Choice of neighboring grid cells
- In practical simulations the high-order accuracy is usually not obtained (grid regularity)

Choice of Stencils



Stencils for O(3) reconstruction, triangular grid

High Order in Time

- **Unsteady**
  - Explicit Runge-Kutta (RK) methods, IMEX-RK (implicit-explicit), fully-implicit RK
  - Space-time expansion: One step method with space time coupling.
- **Steady**
  - Implicit in time and low order accuracy in time (accuracy in time does not affect the steady state)
  - Implicit 1st order

Outline

1. Motivation
2. 2nd order Finite Volume: Reconstruction, Limiting
3. Higher-order Finite Volume: ENO, WENO
4. Discontinuous Galerkin

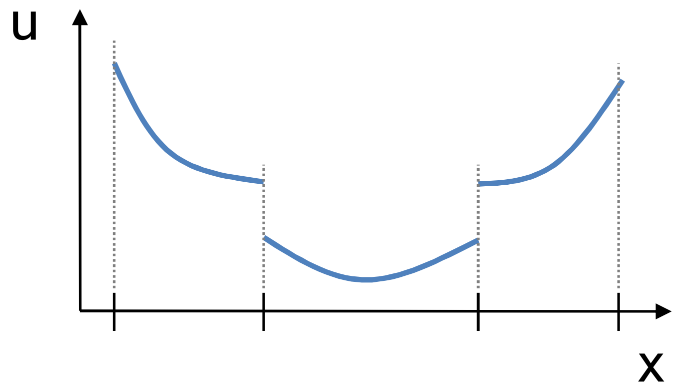
Discontinuous Galerkin (DG): Idea

- Finite Element (Galerkin Method) within cells
- Discontinuities between cells allowed
- Riemann fluxes over discontinuities as in FV
- Difference to (W)ENO / FD: Polynomial *within* cell

--> High order accurate

--> Can handle shocks / discontinuities

--> Compact stencil (local, efficient)



DGSEM: A Special DG Method

7 Steps for derivation:

1. Split into elements, transform to reference element
2. Multiplication with test function, integration over elements
3. Integration by parts
4. Choose basis & test functions
5. Numerical evaluation of the spatial integrals
6. Riemann solvers for fluxes over discontinuities
7. Time integration with explicit RK

DGSEM Derivation (1D)

Step 0: Starting point: Transport equation

$$u_t + \frac{\partial}{\partial x} f(u) = 0.$$

Step 1: Split domain into elements, map each to reference element $[-1,1]^d$

$$\xi = \xi(x); \quad \text{chain rule:} \quad \frac{\partial}{\partial x} = \frac{\partial}{\partial \xi}\left(\frac{\partial \xi}{\partial x}\right)_m = \frac{1}{J_m}\frac{\partial}{\partial \xi}$$

J_m is the Jacobian of element m. Result:

$$J_m\, u_t + \frac{\partial}{\partial \xi} f(u) = 0$$

DGSEM Derivation (1D)

Step 2: Multiplication by test function $\psi = \psi(\xi)$ and integration over the **grid cell** Q_m leads to

$$\int_{Q_m} J_m \, u_t \, \psi \, d\xi + \int_{Q_m} \frac{\partial}{\partial \xi} f(u) \psi \;\, d\xi = 0$$

Step 3: Integration by parts

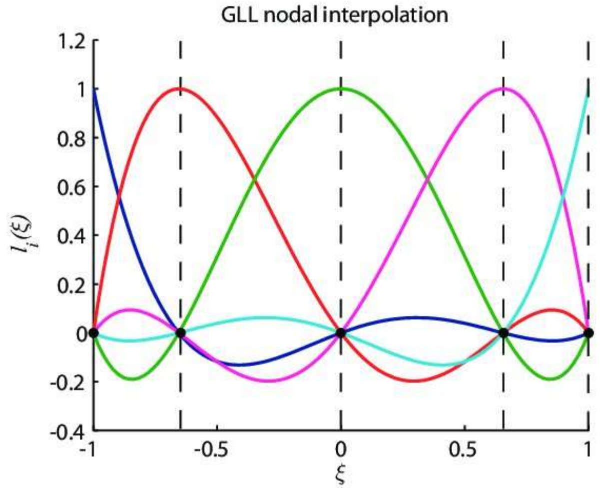
$$\int_{Q_m} J_m \, u_t \, \psi \, d\xi - \int_{Q_m} f(u) \frac{\partial}{\partial \xi} \psi \;\, d\xi + \psi f(u) \Big|_{\xi=1} - \psi f(u) \Big|_{\xi=-1} = 0$$

volume integral   surface terms

DGSEM Derivation (1D)

Step 4: Choose basis for numerical solution.
Polynomial ansatz:

$$u(\xi) \approx u_h(\xi) = \sum_{i=0}^{N} \hat{u}_i \phi_i(\xi)$$



We choose **Lagrange polynomials** $l_i(\xi_j) = \delta_{ij}$ as basis and test functions $l_i = \phi_i = \psi_i$.

Lagrange polynomials are defined on a point set $\{\xi_j\}$
- In DGSEM, **Legendre-Gauss** or **Legendre-Gauss-Lobatto** points

DGSEM Derivation (1D)

Insert:

$$\int_{Q_m} J_m \frac{\partial}{\partial t} u_h \, l_k \, d\xi - \int_{Q_m} f(u_h) \frac{\partial}{\partial \xi} l_k \, d\xi$$
$$+ l_k(1) \, f\big(u_h(1)\big) - l_k(-1) \, f(u_h(-1)) = 0$$

Step 5: Evaluate integrals with numerical quadrature:

$$\int_{Q_m} F \, d\xi \approx \sum_{j=0}^{N} F(\xi_j) w_j$$

In DGSEM, integration points = interpolation points
(**Legendre-Gauss** or **Legendre-Gauss-Lobatto)**

DGSEM Derivation (1D)

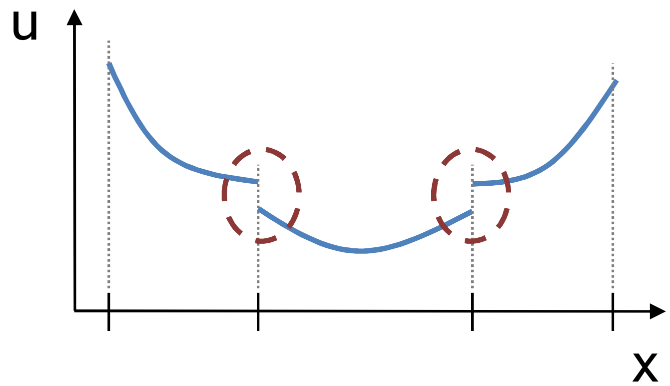
Step 6:

Surface terms

$$l_k(1)\, f\big(u_h(1)\big) - l_k(-1)\, f(u_h(-1))$$

Problem: Solution u_h is discontinuous at element interface $\xi = \pm 1$.
→ Flux $f(u_h)$ is not uniquely defined

Solution: Replace by numerical flux approximation $f^*(u_l, u_r)$ as in FV.



DGSEM Derivation (1D)

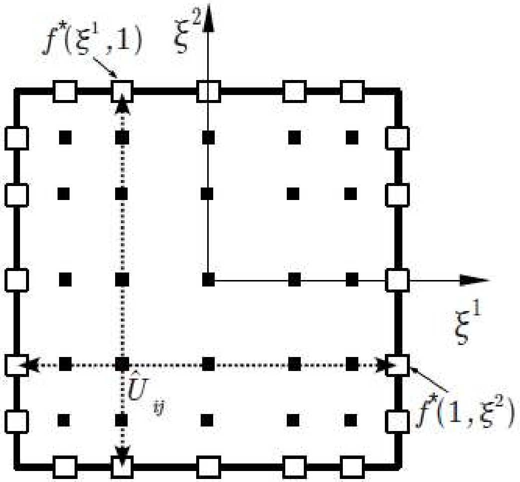
Put it all together: (notation $F^{\pm} = F(\xi = \pm 1)$)

$$J_m \sum_{j=0}^{N} \sum_{i=0}^{N} \frac{\partial}{\partial t} \hat{u}_i l_i(\xi_j) l_k(\xi_j) w_j - \sum_{j=0}^{N} f\left(\sum_{i=0}^{N} \hat{u}_i l_i(\xi_j)\right) l_k'(\xi_j) w_j$$
$$+ l_k^+ f^*(u_m^+, u_{m+1}^-) - l_k^- f^*(u_{m-1}^+, u_m^-) = 0$$

Insert $l_i(\xi_j) = \delta_{ij}$ and re-arrange:

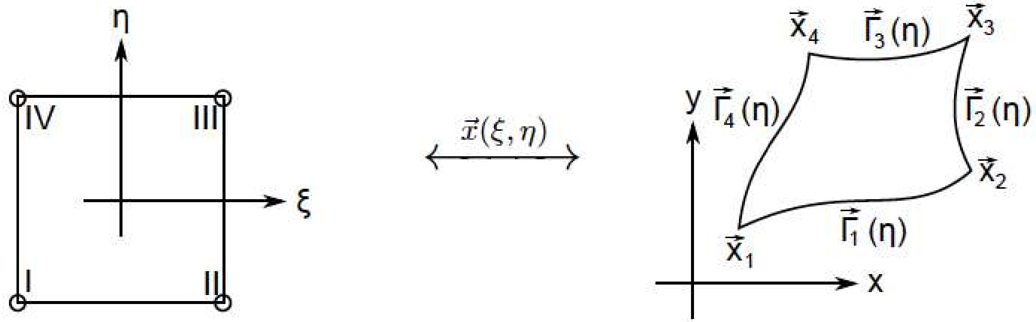
$$\frac{\partial}{\partial t} \hat{u}_k = \frac{1}{J_m w_k} \left[\sum_{j=0}^{N} f(\hat{u}_j) l_k'(\xi_j) w_j - l_k^+ f^*(u_m^+, u_{m+1}^-) + l_k^- f^*(u_{m-1}^+, u_m^-) \right]$$

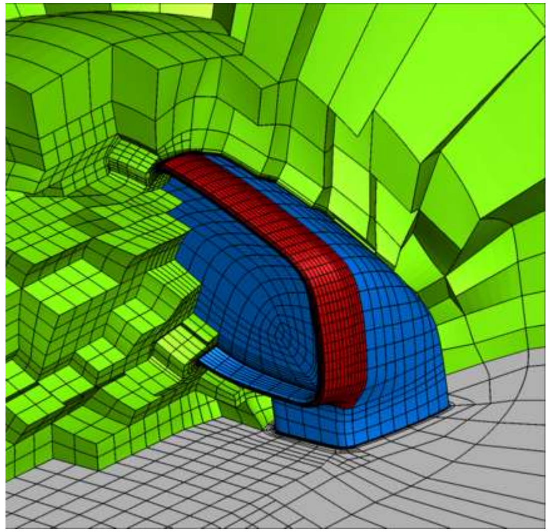
DGSEM 2D/3D



- In 2D/3D, we simply use the 1D operator along the grid lines

- Unstructured curved meshes are possible!





Conclusion

- High-order increases efficiency in most cases
- 2nd order FV via reconstruction and limiting (standard in industry)
- High-order ENO/WENO for FV:
  - mostly block-structured meshes
  - limited flexibility
- Several other high-order methods, e.g. DG(SEM):
  - increased flexibility (unstructured grids)
  - increased efficiency (compact stencil)