# A high-order discontinuous Galerkin method for hyperbolic conservation laws on cut cell meshes

Florian Streitbürger[1]

Joint work: Sandra May[1], Christian Engwer[2],

[1]Department of Mathematics, TU Dortmund University

[2]Applied Mathematics Münster, WWU Münster

Hirschegg 2021

- 13.09.2021 -

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) grant 439956613 (Hypercut).

1. General setting
   - Small cell problem
   - Cut cell model problem in 1D

2. Domain of dependence (DoD) stabilization
   - DoD stabilization for linear advection equation
   - Extension to non-linear equations

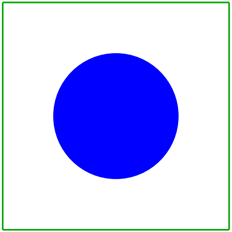
3. Numerical results

# Overview

# Outline

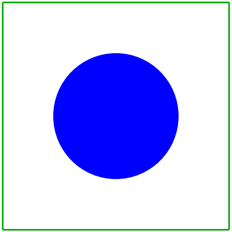
# Cut cells

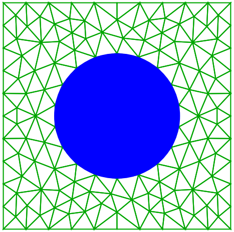
Task: Construction of a body-fitted mesh



# Cut cells

Task: Construction of a body-fitted mesh



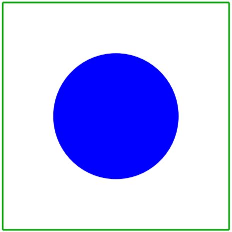


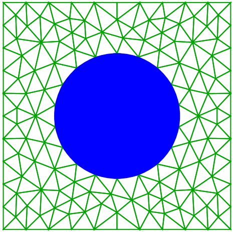
Unstructured triangle grid:

- Time-consuming for complicated geometries

# Cut cells

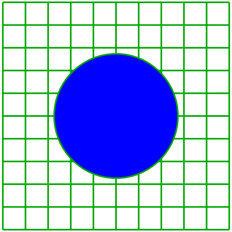
Task: Construction of a body-fitted mesh





Unstructured triangle grid:

- Time-consuming for complicated geometries

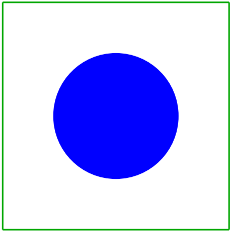


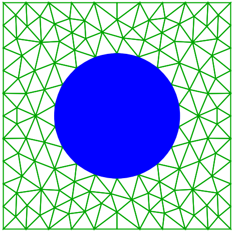
Cut cell grid:

+ Grid generation is easy and cheap
+ Most cells are Cartesian cells

# Cut cells

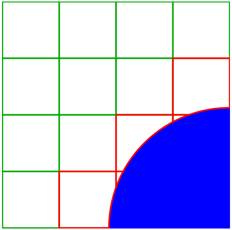
Task: Construction of a body-fitted mesh





Unstructured triangle grid:

- Time-consuming for complicated geometries



Cut cell grid:

+ Grid generation is easy and cheap
+ Most cells are Cartesian cells
- Cut cells irregular & arbitrarily small

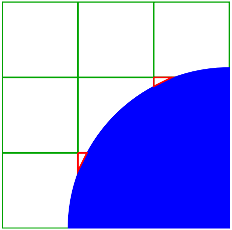
# The small cell problem

Background:

- Typically, explicit timestepping schemes are used for hyp. cons. laws
- CFL condition requires roughly $\Delta t = \mathcal{O}(h)$

Small cell problem:

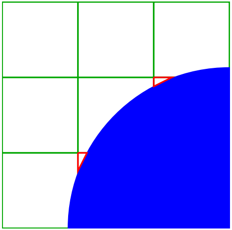
- Goal: Choose $\Delta t$ based on size of Cartesian cells $\Rightarrow$ stability issues on cut cells



# The small cell problem

Background:

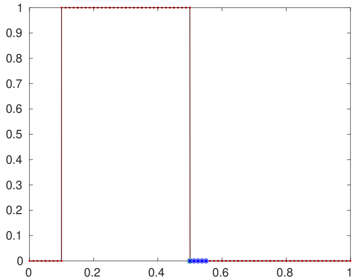
- Typically, explicit timestepping schemes are used for hyp. cons. laws
- CFL condition requires roughly $\Delta t = \mathcal{O}(h)$

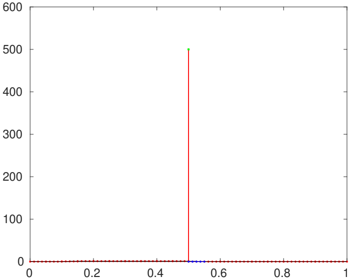


Small cell problem:

- Goal: Choose $\Delta t$ based on size of Cartesian cells $\Rightarrow$ stability issues on cut cells

Numerical example:



# State of the art

**Cell merging / Cell agglomeration** (Krivodonova, Kronbichler, Kummer, Müller, etc.)
$\Rightarrow$ Moves complexity back into grid generation

# State of the art

**Cell merging / Cell agglomeration** (Krivodonova, Kronbichler, Kummer, Müller, etc.)
$\Rightarrow$ Moves complexity back into grid generation

## FV schemes:

- **H-box method** (Berger, Helzel, LeVeque)
- **Flux redistribution** (Colella et al.)
- **Dimensionally split approach** (Klein et al.)
- **Mixed explicit implicit method** (Berger & May)
- **State redistribution** (Berger & Giuliani)
- **Active flux method** (Helzel & Kerkmann)
- & more...

## DG schemes:

- **State redistribution** (Berger & Giuliani)
- **Ghost penalty** (Fu & Kreiss)
- **DoD stabilization**

# Outline

# Hyperbolic conservation laws in 1D

Time-dependent systems of hyperbolic conservation laws in 1D:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0} \quad \text{in } \Omega \times (0,T) \quad \mathbf{u}_0 = \mathbf{u}(\cdot, 0).$$

- $\mathbf{u} : \Omega \times (0,T) \to \mathbb{R}^m$, $m \in \mathbb{N}$ vector of conserved variables
- $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ flux function

# Hyperbolic conservation laws in 1D

Time-dependent systems of hyperbolic conservation laws in 1D:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0} \quad \text{in } \Omega \times (0,T) \quad \mathbf{u}_0 = \mathbf{u}(\cdot, 0).$$

- $\mathbf{u} : \Omega \times (0,T) \to \mathbb{R}^m$, $m \in \mathbb{N}$ vector of conserved variables
- $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ flux function

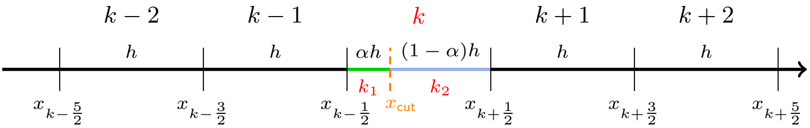
Examples:

- **Burgers equation:** $u_t + \left(\frac{1}{2}u^2\right)_x = 0$
- **Euler equations:** $\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}$ with

$$\mathbf{u} = \begin{pmatrix} \varrho \\ \varrho v \\ E \end{pmatrix} \quad \text{and} \quad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \varrho v \\ \varrho v^2 + p \\ (E+p)v \end{pmatrix}.$$

# Model problem for the small cell problem in 1d

Model problem in 1D with two cut cells $k_1$ and $k_2$:




with:

$$\mathcal{I}_{\text{equi}} = \{1 \leq j \leq N | j \neq k\},\ \mathcal{I}_{\text{all}} = \mathcal{I}_{\text{equi}} \cup \{k_1, k_2\},\ \mathcal{I}_{\mathcal{N}} = \{k-1, k_1, k_2\}.$$

# Model problem for the small cell problem in 1d

DG approach in space with finite-dimensional function space:

$$\mathcal{V}_h^p = \left\{ \mathbf{v}^h \in (L^2(\Omega))^m \;\middle|\; \mathbf{v}^h|_{I_j} \in P^p(I_j) \text{ for all } j \in \mathcal{I}_{\mathsf{all}} \right\},$$

# Model problem for the small cell problem in 1d

DG approach in space with finite-dimensional function space:

$$\mathcal{V}_h^p = \left\{ \mathbf{v}^h \in (L^2(\Omega))^m \,\middle|\, \mathbf{v}^h|_{I_j} \in P^p(I_j) \text{ for all } j \in \mathcal{I}_{\text{all}} \right\},$$

$\Rightarrow$ Semi-discrete problem:

Find $\mathbf{u}^h \in V_h^p$ s.t.: $\left(d_t \mathbf{u}^h(t), \mathbf{w}^h\right)_{L^2} + a_h\left(\mathbf{u}^h(t), \mathbf{w}^h\right) = 0 \quad \forall\, \mathbf{w}^h \in \mathcal{V}_h^p,$

with

$$a_h(\mathbf{u}^h, \mathbf{w}^h) = -\sum_{j \in \mathcal{I}_{\text{all}}} \int_j \mathbf{f}(\mathbf{u}^h) \cdot \partial_x \mathbf{w}^h \, \mathrm{d}x$$

$$+ \sum_{j=0}^{N} \mathcal{H}(\mathbf{u}_j, \mathbf{u}_{j+1})(x_{j+\frac{1}{2}}) \cdot [\![\mathbf{w}^h]\!]_{j+\frac{1}{2}} + \mathcal{H}(\mathbf{u}_{k_1}, \mathbf{u}_{k_2})(x_{\text{cut}}) \cdot [\![\mathbf{w}^h]\!]_{\text{cut}}.$$

# Model problem for the small cell problem in 1d

DG approach in space with finite-dimensional function space:

$$\mathcal{V}_h^p = \left\{ \mathbf{v}^h \in (L^2(\Omega))^m \,\middle|\, \mathbf{v}^h|_{I_j} \in P^p(I_j) \text{ for all } j \in \mathcal{I}_{\text{all}} \right\},$$

⇒ Semi-discrete problem:

Find $\mathbf{u}^h \in V_h^p$ s.t.: $\left(d_t \mathbf{u}^h(t), \mathbf{w}^h\right)_{L^2} + a_h\left(\mathbf{u}^h(t), \mathbf{w}^h\right) = 0 \quad \forall\, \mathbf{w}^h \in \mathcal{V}_h^p,$

with

$$\begin{aligned} a_h(\mathbf{u}^h, \mathbf{w}^h) &= -\sum_{j \in \mathcal{I}_{\text{all}}} \int_j \mathbf{f}(\mathbf{u}^h) \cdot \partial_x \mathbf{w}^h \, \mathrm{d}x \\ &+ \sum_{j=0}^{N} \mathcal{H}(\mathbf{u}_j, \mathbf{u}_{j+1})(x_{j+\frac{1}{2}}) \cdot [\![\mathbf{w}^h]\!]_{j+\frac{1}{2}} + \mathcal{H}(\mathbf{u}_{k_1}, \mathbf{u}_{k_2})(x_{\text{cut}}) \cdot [\![\mathbf{w}^h]\!]_{\text{cut}}. \end{aligned}$$

⇒ Later: Use explicit RK scheme in time of suitable order

# Goal

Goal: Develop a penalty stabilization for general hyperbolic conservation laws on cut cell meshes:

Find $\mathbf{u}^h \in V_h^p$ s.t.:

$$\left(d_t \mathbf{u}^h(t), \mathbf{w}^h\right)_{L^2} + a_h\left(\mathbf{u}^h(t), \mathbf{w}^h\right) + J_h(\mathbf{u}^h(t), \mathbf{w}^h) = 0 \quad \forall\, \mathbf{w}^h \in \mathcal{V}_h^p$$

# Goal

Goal: Develop a penalty stabilization for general hyperbolic conservation laws on cut cell meshes:

Find $\mathbf{u}^h \in V_h^p$ s.t.:

$$\left(d_t \mathbf{u}^h(t), \mathbf{w}^h\right)_{L^2} + a_h\left(\mathbf{u}^h(t), \mathbf{w}^h\right) + J_h(\mathbf{u}^h(t), \mathbf{w}^h) = 0 \quad \forall\, \mathbf{w}^h \in \mathcal{V}_h^p$$

with the following properties:

- Stable scheme that solves the small cell problem
- $L^2$-stability for the semi-discrete scheme
- Expected order of convergence: Order $p+1$ for $\mathcal{V}_h^p$
- Monotone scheme for piecewise constant polynomials

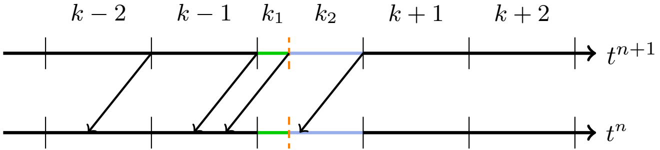
# Overview

# Outline

1. General setting
   - Small cell problem
   - Cut cell model problem in 1D
2. Domain of dependence (DoD) stabilization
   - DoD stabilization for linear advection equation
   - Extension to non-linear equations
3. Numerical results

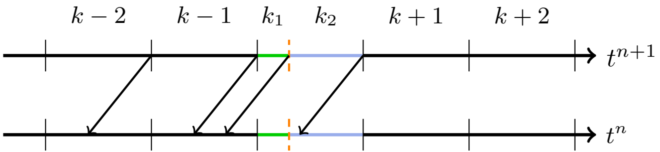
# Update formula without stabilization

Consider the linear advection equation with $\beta > 0$



# Update formula without stabilization

Consider the linear advection equation with $\beta > 0$




Update formulas for $\mathcal{V}_h^0$ + expl. Euler:

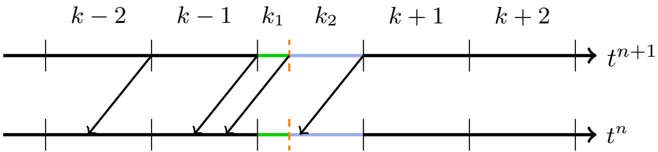
$$u_{k-1}^{n+1} = u_{k-1}^n - \frac{\beta \Delta t}{h}\{u_{k-1}^n - u_{k-2}^n\}$$

$$u_{k_1}^{n+1} = u_{k_1}^n - \frac{\beta \Delta t}{\alpha h}\{u_{k_1}^n - u_{k-1}^n\}$$

$$u_{k_2}^{n+1} = u_{k_2}^n - \frac{\beta \Delta t}{(1-\alpha) h}\{u_{k_2}^n - u_{k_1}^n\}$$

# Update formula without stabilization

Consider the linear advection equation with $\beta > 0$




Update formulas for $\mathcal{V}_h^0$ + expl. Euler:

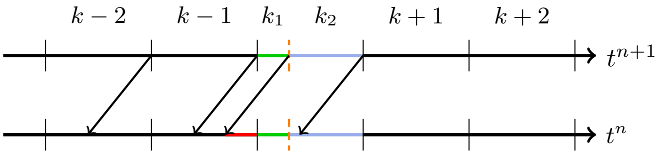
$$u_{k-1}^{n+1} = u_{k-1}^n - \frac{\beta \Delta t}{h}\{u_{k-1}^n - u_{k-2}^n\}$$

$$u_{k_1}^{n+1} = u_{k_1}^n - \frac{\beta \Delta t}{\alpha h}\underbrace{\{u_{k_1}^n - u_{k-1}^n\}}_{\sim \mathcal{O}(\frac{(1+\alpha)h}{2})}$$ ⚠ unstable update

$$u_{k_2}^{n+1} = u_{k_2}^n - \frac{\beta \Delta t}{(1-\alpha)h}\{u_{k_2}^n - u_{k_1}^n\}$$

# Update formula without stabilization

Consider the linear advection equation with $\beta > 0$



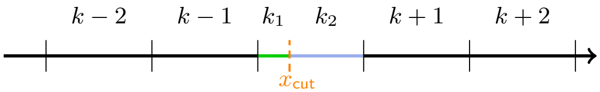

Update formulas for $\mathcal{V}_h^0$ + expl. Euler:

$$u_{k-1}^{n+1} = u_{k-1}^n - \frac{\beta \Delta t}{h}\{u_{k-1}^n - u_{k-2}^n\}$$

$$u_{k_1}^{n+1} = u_{k_1}^n - \frac{\beta \Delta t}{\color{red}{\alpha} h}\underbrace{\{u_{k_1}^n - u_{k-1}^n\}}_{\sim\mathcal{O}(\frac{(1+\alpha)h}{2})} \triangle! \quad \color{red}{\text{unstable update}}$$

$$u_{k_2}^{n+1} = u_{k_2}^n - \frac{\beta \Delta t}{(1-\alpha)h}\{u_{k_2}^n - u_{k_1}^n\} \triangle! \quad \color{red}{\text{lack of information}}$$

# Idea of DoD stabilization

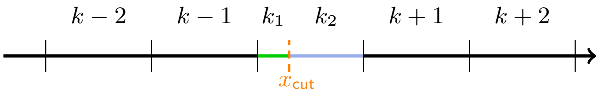



Our approach

$$J_h^0(u^h, w^h) = \beta \eta_{k_1} \left[ u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}}) \right] \left[\!\left[ w^h \right]\!\right]_{\text{cut}}$$

with $\eta_{k_1} \sim 1 - \alpha$.

# Idea of DoD stabilization



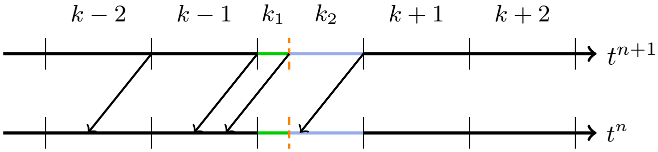

Our approach

$$J_h^0(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w^h]\!]_{\text{cut}}$$

with $\eta_{k_1} \sim 1-\alpha$.

- Adds additional flux on edge $x_{\text{cut}}$
- Evaluates inflow neighbor $u_{k-1}$ outside of its support
- Shifts mass from cell $k-1$ directly into cell $k_2$
- Restores the proper domain of dependence on $k_2$

# Stabilized update formulas



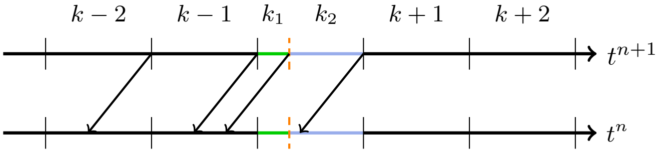

Unstabilized update formulas

$$u_{k-1}^{n+1} = u_{k-1}^{n} - \frac{\beta \Delta t}{h}\{u_{k-1}^{n} - u_{k-2}^{n}\}$$

$$u_{k_1}^{n+1} = u_{k_1}^{n} - \frac{\beta \Delta t}{\alpha h}\{u_{k_1}^{n} - u_{k-1}^{n}\} \triangle$$ unstable update

$$u_{k_2}^{n+1} = u_{k_2}^{n} - \frac{\beta \Delta t}{(1-\alpha) h}\{u_{k_2}^{n} - u_{k_1}^{n}\} \triangle$$ lack of information

# Stabilized update formulas




Unstabilized update formulas

$$u_{k-1}^{n+1} = u_{k-1}^{n} - \frac{\beta \Delta t}{h}\{u_{k-1}^{n} - u_{k-2}^{n}\}$$

$$u_{k_1}^{n+1} = u_{k_1}^{n} - \frac{\beta \Delta t}{\alpha h}\{u_{k_1}^{n} - u_{k-1}^{n}\} ⚠ \quad \text{unstable update}$$

$$u_{k_2}^{n+1} = u_{k_2}^{n} - \frac{\beta \Delta t}{(1-\alpha) h}\{u_{k_2}^{n} - u_{k_1}^{n}\} ⚠ \quad \text{lack of information}$$

Stabilized update formulas; $1 - \eta_{k_1} \sim \alpha$

$$u_{k-1}^{n+1} = u_{k-1}^{n} - \frac{\beta \Delta t}{h}\{u_{k-1}^{n} - u_{k-2}^{n}\},$$

$$u_{k_1}^{n+1} = u_{k_1}^{n} - \frac{\beta \Delta t}{\alpha h}(1-\eta_{k_1})\{u_{k_1}^{n} - u_{k-1}^{n}\},$$

$$u_{k_2}^{n+1} = u_{k_2}^{n} - \frac{\beta \Delta t}{(1-\alpha) h}\{u_{k_2}^{n} - (1-\eta_{k_1}) u_{k_1}^{n} - \eta_{k_1} u_{k-1}^{n}\}$$

# $L^2$-stability

$$J_h^0(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\mathsf{cut}}) - u_{k_1}(x_{\mathsf{cut}})\right] \left[\!\left[ w^h \right]\!\right]_{\mathsf{cut}}$$

- Problem: When using $\mathcal{V}_h^p,\ p > 0$ adding $J_h^0$ destroys the $L^2$-stability
- Next goal: Restore $L^2$-stability

# $L^2$-stability

$$J_h^0(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w^h]\!]_{\text{cut}}$$

- Problem: When using $\mathcal{V}_h^p,\ p > 0$ adding $J_h^0$ destroys the $L^2$-stability
- Next goal: Restore $L^2$-stability
- Theoretical and numerical investigations yield to the following term:

$$J_h^1(u^h, w^h) = \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\left[\partial_x w_{k-1}(x) - \partial_x w_{k_1}(x)\right]\mathrm{d}x$$

Full stabilization given by:

$$J_h(u^h, w^h) = J_h^0(u^h, w^h) + J_h^1(u^h, w^h)$$

# Theoretical properties

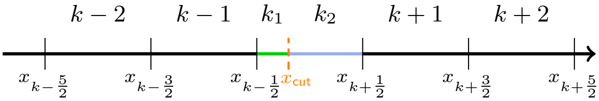
**Theorem**

*DoD stabilized scheme is:*

- *Conservative on a patch (cut cell + neighbor)*
- *$L^2$ stable in semi-discrete form*
- *Monotone using a first-order scheme*

# Outline

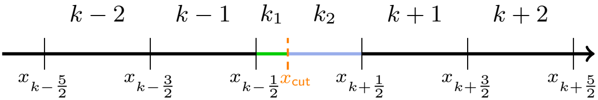
# Extension of $J_h^0$




$$J_h^0 = \beta \eta_{k_1} \left[ u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}}) \right] [\![ w^h ]\!]_{\text{cut}}$$

Q: How to extend that to non-linear equations?

# Extension of $J_h^0$



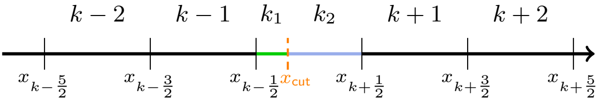

$$J_h^0 = \beta \eta_{k_1} \left[ u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}}) \right] \left[\!\left[ w^h \right]\!\right]_{\text{cut}}$$

Q: How to extend that to non-linear equations?

Our interpretation:

$$\Rightarrow J_h^0 = \eta_{k_1} \left[ \mathcal{H}(u_{k-1}, u_{k_2})(x_{\text{cut}}) - \mathcal{H}(u_{k_1}, u_{k_2})(x_{\text{cut}}) \right] \left[\!\left[ w^h \right]\!\right]_{\text{cut}}$$

# Extension of $J_h^0$




$$J_h^0 = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] \llbracket w^h \rrbracket_{\text{cut}}$$

Q: How to extend that to non-linear equations?

Our interpretation:

$$\Rightarrow J_h^0 = \eta_{k_1}\left[\mathcal{H}(u_{k-1}, u_{k_2})(x_{\text{cut}}) - \mathcal{H}(u_{k_1}, u_{k_2})(x_{\text{cut}})\right] \llbracket w^h \rrbracket_{\text{cut}}$$

Considering non-linear systems $\Rightarrow$ Two possible flow directions

$$\begin{aligned} J_h^0(\mathbf{u}^h, \mathbf{w}^h) = \eta_{k_1}&\left[\mathcal{H}(\mathbf{u}_{k-1}, \mathbf{u}_{k_2})(x_{k-\frac{1}{2}}) - \mathcal{H}(\mathbf{u}_{k-1}, \mathbf{u}_{k_1})(x_{k-\frac{1}{2}})\right] \cdot \llbracket \mathbf{w}^h \rrbracket_{k-\frac{1}{2}} \\ &+ \eta_{k_1}\left[\mathcal{H}(\mathbf{u}_{k-1}, \mathbf{u}_{k_2})(x_{\text{cut}}) - \mathcal{H}(\mathbf{u}_{k_1}, \mathbf{u}_{k_2})(x_{\text{cut}})\right] \cdot \llbracket \mathbf{w}^h \rrbracket_{\text{cut}} \end{aligned}$$

# Extension of $J_h^1$

- Again: Adding $J_h^0$ destroys the $L^2$-stability

$\Rightarrow$ Derive missing terms by $L^2$-stability proof

$$
\begin{aligned}
J_h^1(\mathbf{u}^h, \mathbf{w}^h) = \eta_{k_1} \sum_{j \in \mathcal{I}_\mathcal{N}} \mathbf{K}(j) \int_{k_1} \left( \mathcal{H}(\mathbf{u}_{k-1}, \mathbf{u}_{k_2}) - \mathbf{f}(\mathbf{u}_j) \right) \cdot \partial_x \mathbf{w}_j \mathrm{d}x \\
+ \eta_{k_1} \sum_{j \in \mathcal{I}_\mathcal{N}} \mathbf{K}(j) \int_{k_1} \left( \mathcal{H}_a(\mathbf{u}_{k-1}, \mathbf{u}_{k_2}) \mathbf{u}_j \right) \cdot \partial_x \mathbf{w}_{k-1} \mathrm{d}x \\
+ \eta_{k_1} \sum_{j \in \mathcal{I}_\mathcal{N}} \mathbf{K}(j) \int_{k_1} \left( \mathcal{H}_b(\mathbf{u}_{k-1}, \mathbf{u}_{k_2}) \mathbf{u}_j \right) \cdot \partial_x \mathbf{w}_{k_2} \mathrm{d}x.
\end{aligned}
$$

- $J_h^1$ controls the mass distribution within the small cut cell $k_1$ and its neighbors $k-1$ and $k_2$
- Note: For lin adv equation $J_h^1(\mathbf{u}^h, \mathbf{w}^h)$ reduces to known term

# Properties for non-linear equations

**Theorem**

*Properties of DoD stabilization transfer to non-linear equations when monotone fluxes are used:*

- *Conservative on a patch (cut cell + neighbor)*
- $L^2$ *stable in semi-discrete form for scalar hyperbolic equations*
- *First order scheme is monotone for scalar hyperbolic equations*

# Overview

# Test setup

Between 0.1 and 0.9 we split each cell in a pair of cut cells


Two test cases:

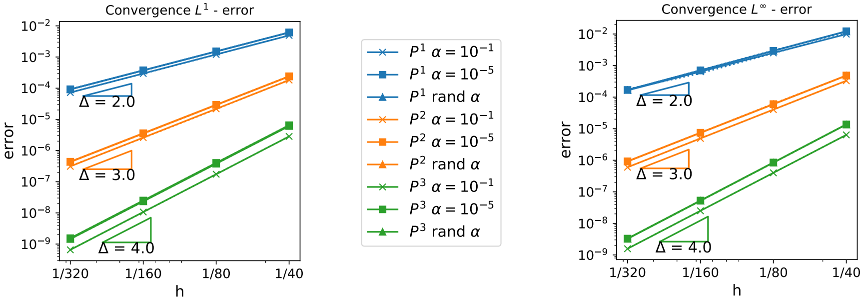
- Case 1 ('$\alpha = 10^{-\square}$'): Same cut cell fraction for all cut cell pairs
- Case 2 ('rand $\alpha$'): Choose $\alpha_k = 10^{-2} X_k$ with $X_k$ being uniformly distributed random number in $(0, 1)$

For convergence tests of non-linear equations we will use the concept of manufactured solutions.

# Burgers equation: convergence test

Burgers equation $u_t + \left(\frac{1}{2}u^2\right)_x = g$ with the manufactured solution

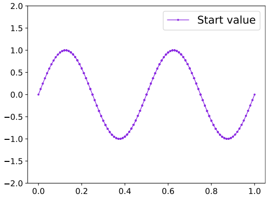
$$u(x,t) = \sin(4\pi(x-t))$$



# Burgers equation: stability test

Initial data:

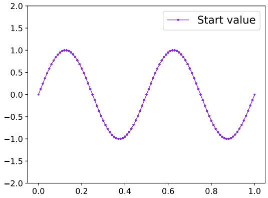
$$u_0(x) = \sin(4\pi(x + 0.5))$$



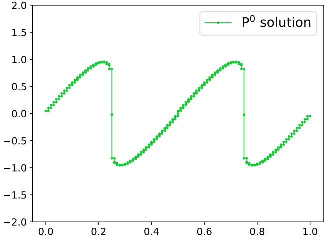
# Burgers equation: stability test

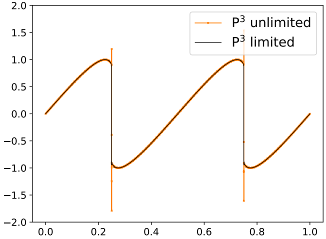
Initial data:

$$u_0(x) = \sin(4\pi(x + 0.5))$$




Solution $(T = 0.1)$:

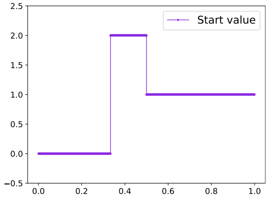


# Burgers equation stability test

Initial data:

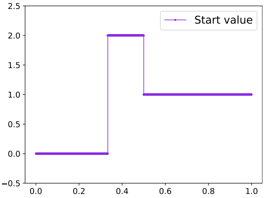
$$u_0(x) = \begin{cases} 0 & \text{if } x < \frac{1}{3} \\ 2 & \text{if } \frac{1}{3} \leq x < \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq x \end{cases}$$



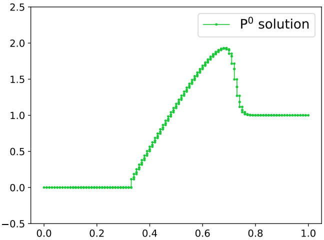
# Burgers equation stability test

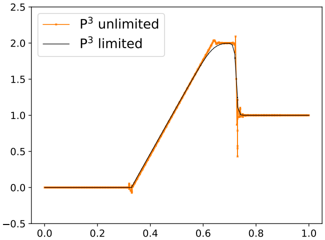
Initial data:

$$u_0(x) = \begin{cases} 0 & \text{if } x < \frac{1}{3} \\ 2 & \text{if } \frac{1}{3} \leq x < \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \leq x \end{cases}$$




Solution ($T = 0.15$):

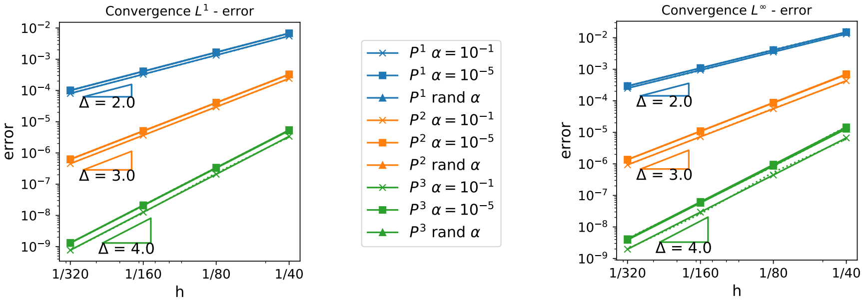


# Euler equations: convergence test

Manufactured solution (in primitive variables):

$$\begin{pmatrix} \varrho \\ v \\ p \end{pmatrix} = \begin{pmatrix} 2 + \sin(2\pi(x-t)) \\ \sin(2\pi(x-t)) \\ 2 + \cos(2\pi(x-t)) \end{pmatrix}$$



# Euler equations: Sod shock tube test

Sod shock tube test:

$$(\varrho, \varrho v, E) = \begin{cases} (1, 0, 2.5) & \text{if } x < 0, \\ (0.125, 0, 0.25) & \text{otherwise.} \end{cases}$$

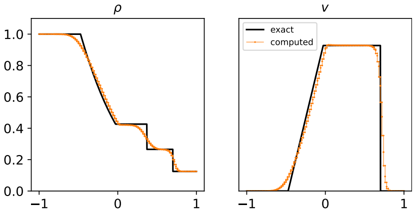
# Euler equations: Sod shock tube test

Sod shock tube test:

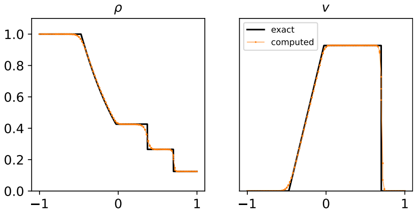
$$(\varrho, \varrho v, E) = \begin{cases} (1, 0, 2.5) & \text{if } x < 0, \\ (0.125, 0, 0.25) & \text{otherwise.} \end{cases}$$

Solution $(T = 0.4)$ for test case 2 ('rand $\alpha$'):

$V_h^0$:




$V_h^1$ + limiter:



# Resumé

Summary:

- Stable scheme for hyperbolic conservation laws on cut cell meshes
- Monotone for scalar conservation laws + $\mathcal{V}_h^0$
- Semi-discrete scheme is $L^2$ stable for $\mathcal{V}_h^p$
- Numerical results: robust behavior + correct order of convergence


S. May, F. Streitbürger
DoD Stabilization for non-linear hyperbolic conservation laws on cut cell meshes in one dimension.
*Under review.* Preprint available: [arXiv:2107.03689]

C. Engwer, S. May, C. Nüßing, F. Streitbürger
A stabilized discontinuous Galerkin cut cell method for discretizing the linear transport equation.
SIAM J. Sci. Comput., 42(6):A3677–A3703, 2020.


Future plans:

- Extend the formulation to non-linear equations in higher dimensions

# Thank you for your attention!

Deutsche Forschungsgemeinschaft

This work was financially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 439956613 (Hypercut).

# References


S. May, F. Streitbürger
DoD Stabilization for non-linear hyperbolic conservation laws on cut cell meshes in one dimension.
*Under review.* Preprint available: [arXiv:2107.03689]

F. Streitbürger, C. Engwer, S. May, A. Nüßing
Monotonicity considerations for stabilized DG cut cell schemes for the unsteady advection equation.
In F.J. Vermolen and C. Vuik, editors, Numerical Mathematics and Advanced Applications ENUMATH 2019, pages 929–937. Springer International Publishing, 2021.

C. Engwer, S. May, C. Nüßing, F. Streitbürger
A stabilized discontinuous Galerkin cut cell method for discretizing the linear transport equation.
SIAM J. Sci. Comput., 42(6):A3677–A3703, 2020.

P. Fu, G. Kreiss
High order cut discontinuous Galerkin methods for hyperbolic conservation laws in one space dimension.
SIAM J. Sci. Comput., 43(4):A2404–A2424, 2021.

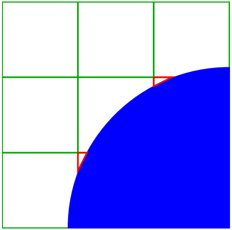
A. Giuliani
A two-dimensional stabilized discontinuous Galerkin method on curviliniear embedded boundary grids.
*Under review.* Preprint available: [arXiv:2102.01857].

# BACKUP

# The small cell problem

Task: Construction of a body-fitted mesh

One possible way: unstructured triangle grid
$\Rightarrow$ Time-consuming for complicated geometries }
Alternative: Cut cell meshes



- + Grid generation is easy and cheap
- + Most cells are Cartesian cells
- - Cut cells irregular & arbitrarily small
- $\Rightarrow$ Small cell problem

Background:

- Typically, explicit timestepping schemes are used for hyp. cons. laws
- CFL condition requires roughly $\Delta t = \mathcal{O}(h)$

The small cell problem:

- Goal: Choose $\Delta t$ based on size of Cartesian cells $\Rightarrow$ stability issues on cut cells

# Overview: Equations

- Linear advection equation: $f(u) = \beta u$ with $\beta \in \mathbb{R}$ constant

$$\Rightarrow u_t + \beta u_x = 0$$

- Burgers equation: $f(u) = \frac{1}{2}u^2$

$$\Rightarrow u_t + (\frac{1}{2}u^2)_x = 0$$

- Linear system: $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$ with constant matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$

$$\Rightarrow \mathbf{u}_t + \mathbf{A}\mathbf{u}_x = 0$$

# Overview: Equations

- Compressible Euler equations satisfy

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}$$

with

$$\mathbf{u} = \begin{pmatrix} \varrho \\ \varrho v \\ E \end{pmatrix} \quad \text{and} \quad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \varrho v \\ \varrho v^2 + p \\ (E+p)v \end{pmatrix}.$$

- Conserved variables:
  $\varrho$ : density, $v$ : velocity, $p$ : pressure, $E$ : energy
- Equation of state:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\varrho v^2$$

- We set $\gamma = 1.4$ in our numerical tests.

# Reminder: DoD stabilization

Quick reminder: DoD stabilization

- Add penalization $J_h$ on semi-discrete formulation in space:

$$\left(d_t u^h(t), w^h\right)_{L^2} + a_h\left(u^h(t), w^h\right) + J_h(u^h(t), w^h) = 0 \quad \forall\, w^h \in V_h^p$$

$\Rightarrow$ Able to use explicit time stepping schemes on cut cell meshes with $\Delta t$ chosen according to cartesian cells

# Reminder: DoD stabilization

Quick reminder: DoD stabilization

- Add penalization $J_h$ on semi-discrete formulation in space:

$$\left(d_t u^h(t), w^h\right)_{L^2} + a_h\left(u^h(t), w^h\right) + J_h(u^h(t), w^h) = 0 \quad \forall\, w^h \in V_h^p$$

$\Rightarrow$ Able to use explicit time stepping schemes on cut cell meshes with $\Delta t$ chosen according to cartesian cells

- Split $J_h(u^h(t), w^h)$ in two terms:

$$J_h(u^h(t), w^h) = J_h^0(u^h(t), w^h) + J_h^1(u^h(t), w^h)$$

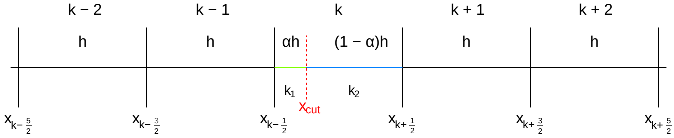
- $J_h^0(u^h(t), w^h)$: Redistributes mass between the small cut cell and its neighbors
- $J_h^1(u^h(t), w^h)$: Redistributes mass within the small cut cell

# DoD stabilization for lin adv equation in 1D

Linear advection equation:

$$u_t + \beta u_x = 0$$

For case of positive $\beta > 0 + \mathcal{V}_h^1 +$ model problem:



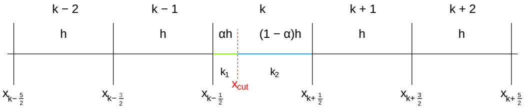

$$J_h^0 = \beta \eta_{k_1} \left[ u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}}) \right] [\![ w_h ]\!]_{\text{cut}}$$

$$J_h^1 = -\beta \eta_{k_1} \int_{k_1} \left[ u_{k-1}(x) - u_{k_1}(x) \right] \partial_x w_{k_1}(x) \, \mathrm{d}x$$

with

$$\eta_{k_1} \sim 1 - \alpha.$$

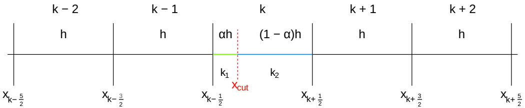
# Piecewise constant polynomials & explicit Euler




Scalar conservation law: $u_t + f(u)_x = 0$

$$\begin{aligned} u_{k-1}^{n+1} &= u_{k-1}^{n} - \frac{\Delta t}{h}\{\mathcal{H}(u_{k-1}^n, u_{k_1}^n) - \mathcal{H}(u_{k-2}^n, u_{k-1}^n)\}, \\ u_{k_1}^{n+1} &= u_{k_1}^{n} - \frac{\Delta t}{\alpha h}\{\mathcal{H}(u_{k_1}^n, u_{k_2}^n) - \mathcal{H}(u_{k-1}^n, u_{k_1}^n)\}, \\ u_{k_2}^{n+1} &= u_{k_2}^{n} - \frac{\Delta t}{(1-\alpha) h}\{\mathcal{H}(u_{k_2}^n, u_{k+1}^n) - \mathcal{H}(u_{k_1}^n, u_{k_2}^n)\} \end{aligned}$$

# Piecewise constant polynomials & explicit Euler



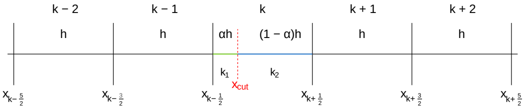

Scalar conservation law: $u_t + f(u)_x = 0$

$$
\begin{aligned}
u_{k-1}^{n+1} &= u_{k-1}^{n} - \frac{\Delta t}{h}\{\mathcal{H}(u_{k-1}^{n}, u_{k_1}^{n}) - \mathcal{H}(u_{k-2}^{n}, u_{k-1}^{n})\},\\
u_{k_1}^{n+1} &= u_{k_1}^{n} - \frac{\Delta t}{\alpha h}\{\mathcal{H}(u_{k_1}^{n}, u_{k_2}^{n}) - \mathcal{H}(u_{k-1}^{n}, u_{k_1}^{n})\},\\
u_{k_2}^{n+1} &= u_{k_2}^{n} - \frac{\Delta t}{(1-\alpha)h}\{\mathcal{H}(u_{k_2}^{n}, u_{k+1}^{n}) - \mathcal{H}(u_{k_1}^{n}, u_{k_2}^{n})\}
\end{aligned}
$$

Stabilized update formula: $\eta_{k_1} \sim 1-\alpha \quad \Rightarrow \quad 1-\eta_{k_1} \sim \alpha$

$$
\begin{aligned}
u_{k-1}^{n+1} &= u_{k-1}^{n} - \frac{\Delta t}{h}\{(1-\eta_{k_1})\mathcal{H}(u_{k-1}^{n}, u_{k_1}^{n}) + \eta_{k_1}\mathcal{H}(u_{k-1}^{n}, u_{k_2}^{n}) - \mathcal{H}(u_{k-2}^{n}, u_{k-1}^{n})\},\\
u_{k_1}^{n+1} &= u_{k_1}^{n} - \frac{\Delta t}{\alpha h}(1-\eta_{k_1})\{\mathcal{H}(u_{k_1}^{n}, u_{k_2}^{n}) - \mathcal{H}(u_{k-1}^{n}, u_{k_1}^{n})\},\\
u_{k_2}^{n+1} &= u_{k_2}^{n} - \frac{\Delta t}{(1-\alpha)h}\{\mathcal{H}(u_{k_2}^{n}, u_{k+1}^{n}) - (1-\eta_{k_1})\mathcal{H}(u_{k_1}^{n}, u_{k_2}^{n}) - \eta_{k_1}\mathcal{H}(u_{k-1}^{n}, u_{k_2}^{n})\}
\end{aligned}
$$

# Difference for lin adv equation

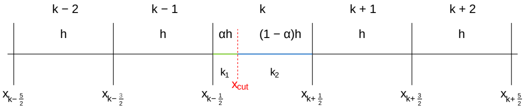



For the special case of linear advection ($\beta > 0$):

Extension of formulation for $\mathcal{V}_h^p$

$$J_h(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w_h]\!]_{\text{cut}}$$
$$+ \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\left[\partial_x w_{k-1}(x) - \partial_x w_{k_1}(x)\right]\mathrm{d}x\,.$$

# Difference for lin adv equation




For the special case of linear advection ($\beta > 0$):

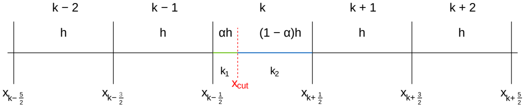
Extension of formulation for $\mathcal{V}_h^p$

$$J_h(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w_h]\!]_{\text{cut}} + \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\left[\partial_x w_{k-1}(x) - \partial_x w_{k_1}(x)\right]\mathrm{d}x\,.$$

Compare to formulation for $\mathcal{V}_h^1$

$$J_h(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w_h]\!]_{\text{cut}} - \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\partial_x w_{k_1}(x)\,\mathrm{d}x\,.$$

# Difference for lin adv equation




For the special case of linear advection ($\beta > 0$):

Extension of formulation for $\mathcal{V}_h^p$

$$J_h(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w_h]\!]_{\text{cut}} + \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\left[\partial_x w_{k-1}(x) - \partial_x w_{k_1}(x)\right] \mathrm{d}x\,.$$

Compare to formulation for $\mathcal{V}_h^1$

$$J_h(u^h, w^h) = \beta\eta_{k_1}\left[u_{k-1}(x_{\text{cut}}) - u_{k_1}(x_{\text{cut}})\right] [\![w_h]\!]_{\text{cut}} - \beta\eta_{k_1}\int_{k_1}\left[u_{k-1}(x) - u_{k_1}(x)\right]\partial_x w_{k_1}(x)\, \mathrm{d}x\,.$$

# Extension of $J_{1,h}$: Choice of $\mathbf{K}(j)$

We set

$$\mathbf{K}(k-1) = \mathbf{L}_{k_1}, \quad \mathbf{K}(k_1) = -\mathbf{I}^m, \quad \text{and} \quad \mathbf{K}(k_2) = \mathbf{R}_{k_1}.$$

Consider linear equation: $\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = \mathbf{0}$

- Hyperbolicity $\Rightarrow$ $\mathbf{A}$ is diagonalizable with $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$

# Extension of $J_{1,h}$: Choice of $\mathbf{K}(j)$

We set

$$\mathbf{K}(k-1) = \mathbf{L}_{k_1}, \quad \mathbf{K}(k_1) = -\mathbf{I}^m, \quad \text{and} \quad \mathbf{K}(k_2) = \mathbf{R}_{k_1}.$$

Consider linear equation: $\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = \mathbf{0}$

- Hyperbolicity $\Rightarrow$ $\mathbf{A}$ is diagonalizable with $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$

Define

$$\mathbf{I}^+_{ii} = \begin{cases} 1 & \text{if } \mathbf{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \mathbf{\Lambda}_{ii} = 0, \\ 0 & \text{if } \mathbf{\Lambda}_{ii} < 0, \end{cases} \quad \text{and} \quad \mathbf{I}^-_{ii} = \begin{cases} 0 & \text{if } \mathbf{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \mathbf{\Lambda}_{ii} = 0, \\ 1 & \text{if } \mathbf{\Lambda}_{ii} < 0. \end{cases}$$

# Extension of $J_{1,h}$: Choice of $\mathbf{K}(j)$

We set

$$\mathbf{K}(k-1) = \mathbf{L}_{k_1}, \quad \mathbf{K}(k_1) = -\mathbf{I}^m, \quad \text{and} \quad \mathbf{K}(k_2) = \mathbf{R}_{k_1}.$$

Consider linear equation: $\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = \mathbf{0}$

- Hyperbolicity $\Rightarrow$ $\mathbf{A}$ is diagonalizable with $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$

Define

$$\mathbf{I}^+_{ii} = \begin{cases} 1 & \text{if } \mathbf{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \mathbf{\Lambda}_{ii} = 0, \\ 0 & \text{if } \mathbf{\Lambda}_{ii} < 0, \end{cases} \quad \text{and} \quad \mathbf{I}^-_{ii} = \begin{cases} 0 & \text{if } \mathbf{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \mathbf{\Lambda}_{ii} = 0, \\ 1 & \text{if } \mathbf{\Lambda}_{ii} < 0. \end{cases}$$

Then:

$$\mathbf{L}_{k_1} = \mathbf{Q}\mathbf{I}^+\mathbf{Q}^{-1} \quad \text{and} \quad \mathbf{R}_{k_1} = \mathbf{Q}\mathbf{I}^-\mathbf{Q}^{-1}.$$

# Extension of $J_{1,h}$: Choice of $\mathbf{K}(j)$

We set

$$\mathbf{K}(k-1) = \mathbf{L}_{k_1}, \quad \mathbf{K}(k_1) = -\mathbf{I}^m, \quad \text{and} \quad \mathbf{K}(k_2) = \mathbf{R}_{k_1}.$$

Consider linear equation: $\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = \mathbf{0}$

- Hyperbolicity $\Rightarrow$ $\mathbf{A}$ is diagonalizable with $\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}$

Define

$$\mathbf{I}^+_{ii} = \begin{cases} 1 & \text{if } \boldsymbol{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \boldsymbol{\Lambda}_{ii} = 0, \\ 0 & \text{if } \boldsymbol{\Lambda}_{ii} < 0, \end{cases} \quad \text{and} \quad \mathbf{I}^-_{ii} = \begin{cases} 0 & \text{if } \boldsymbol{\Lambda}_{ii} > 0, \\ \frac{1}{2} & \text{if } \boldsymbol{\Lambda}_{ii} = 0, \\ 1 & \text{if } \boldsymbol{\Lambda}_{ii} < 0. \end{cases}$$

Then:

$$\mathbf{L}_{k_1} = \mathbf{Q}\mathbf{I}^+\mathbf{Q}^{-1} \quad \text{and} \quad \mathbf{R}_{k_1} = \mathbf{Q}\mathbf{I}^-\mathbf{Q}^{-1}.$$

Non-linear equations: $\mathbf{u}_t + \mathbf{f}_{\mathbf{u}}(\mathbf{u})\mathbf{u}_x = \mathbf{0}$

$\Rightarrow$ Linearize the (non-linear) Jacobian matrix $\mathbf{f}_{\mathbf{u}}(\mathbf{u})$ by suitable average of $\mathbf{u}_{k-1}$ and $\mathbf{u}_{k_2}$

# Monotonicity

**Theorem**

*Consider the model problem MP for a scalar conservation law*

- *using $\mathcal{V}_h^0$ with the DoD stabilization*
- *explicit Euler time stepping*
- *periodic boundary conditions.*

*If the numerical flux is a monotone flux and satisfies*

$$|\mathcal{H}_a(u,v)| + |\mathcal{H}_b(w,u)| \leq \frac{\nu h}{\Delta t} \quad \forall u, v, w.$$

*Then, for the time step be given by $\Delta t = \frac{\nu h}{\lambda_{\max}}$ for $0 < \alpha < \nu < 1 - \alpha$, the scheme is monotone.*

# $L^2$ stability for the semi-discrete scheme

**Theorem**

*Let $u^h(t)$ be the solution to the semi-discrete problem for the scalar equation $u_t + f(u)_x = 0$. Let the numerical flux be a monotone flux. Then, the soluiton satisfies for all $t \in (0, T)$*

$$\left\| u^h(t) \right\|_{L^2(\Omega)} \leq \left\| u^h(0) \right\|_{L^2(\Omega)}.$$

Remark:

- *un*stabilized semi-discrete scheme is also $L^2$ stable
- BUT: unstabilized scheme + explicit time stepping scheme $\Rightarrow$ small cell problem

**Challenge:**

Design a stabilization term $J_h$ that is $L^2$ stable and solves the small cell problem

# Manufactured solutions

Concept of manufactured solutions:

- For convergence tests we need smooth solutions

Problem: Non-trivial for non-linear hyperbolic equations, especially for the compressible Euler equations

$\Rightarrow$ Manufactured solutions: We select a smooth function $\mathbf{u}(x,t)$ that should be the solution of our system.

- $\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x$ will probably not be zero but will result in a non-zero source term $\mathbf{g}$

$\Rightarrow$ For convergence tests of non-linear equations we then solve the system:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{g} \quad \text{in } \Omega \times (0,T).$$

# Limiter

Use TVDM generalized slope limiter by Cockburn & Shu [1] with modifications on cut cells:

1. Compute ($\tilde{m}()$ is minmod-function):

$$u_j^{\lim}(x^+_{j-\frac{1}{2}}) = \bar{u}_j - \tilde{m}(\bar{u}_j - u_j(x^+_{j-\frac{1}{2}}), \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j)$$
$$u_j^{\lim}(x^-_{j+\frac{1}{2}}) = \bar{u}_j + \tilde{m}(u_j(x^-_{j+\frac{1}{2}}) - \bar{u}_j, \bar{u}_j - \bar{u}_{j-1}, \bar{u}_{j+1} - \bar{u}_j)$$

2. If

$$u_j^{\lim}(x^+_{j-\frac{1}{2}}) == u_j(x^+_{j-\frac{1}{2}}) \quad \&\& \quad u_j^{\lim}(x^-_{j+\frac{1}{2}}) == u_j(x^-_{j+\frac{1}{2}})$$

do not limit. Otherwise, reduce $u_j$ to $P^1$. Limit linear polynomial.

3. Postprocess and additionally enforce

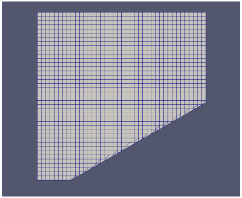
$$\min\left(\bar{u}^n_{k-1}, \bar{u}^n_{k_1}, \bar{u}^n_{k_2}\right) \leq u_{k-1}(x_{\text{cut}}) \leq \max\left(\bar{u}^n_{k-1}, \bar{u}^n_{k_1}, \bar{u}^n_{k_2}\right),$$
$$\min\left(\bar{u}^n_{k-1}, \bar{u}^n_{k_1}, \bar{u}^n_{k_2}\right) \leq u_{k_2}(x_{k-\frac{1}{2}}) \leq \max\left(\bar{u}^n_{k-1}, \bar{u}^n_{k_1}, \bar{u}^n_{k_2}\right).$$

[1] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous GalerkinFinite Element Method for Conservation Laws II: General framework.

# Numerical results in 2d

Setup:

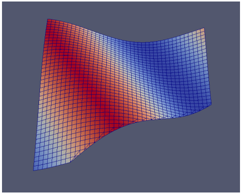
- $N \times N$ grid
- Cut out lower right corner $\Rightarrow$ Create a ramp with angle $\gamma$



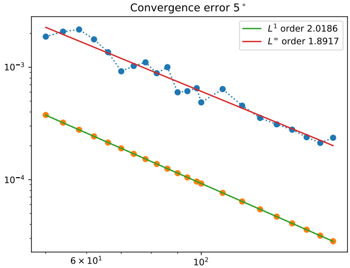
# Numerical results in 2d

Setup:

- $N \times N$ grid
- Cut out lower right corner $\Rightarrow$ Create a ramp with angle $\gamma$
- Initial value: Sine-function parallel to ramp



Ramp $\gamma = 5°$




Ramp $\gamma = 35°$