# DG lecture notes

Florian Kummer & Björn Müller

March 6, 2017

## Contents

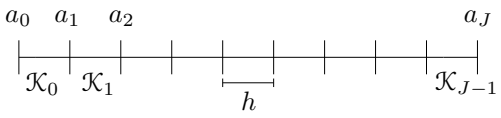
# 1 Approximation of functions by broken polynomials

## 1.1 Definition of a grid in one dimension

- Grid as a set of cells $\mathcal{K}_0, \dots, \mathcal{K}_{J-1}$:

$$\mathfrak{K}_h := \{\mathcal{K}_0, \mathcal{K}_1, \dots, \mathcal{K}_{J-1}\}, \tag{1}$$

- 1D-example:




$$\mathfrak{K}_h := \{\mathcal{K}_0, \mathcal{K}_1, \dots, \mathcal{K}_{J-1}\}, \tag{2}$$

  $h$ is characteristic length-scale of the grid.

- Index convention: we start at index 0, since this is also used in the C#-language and therefore in the BoSSS code.
- Cells are open sets: $\mathcal{K}_j = (a_j, a_{j+1}) = \{x; a_j < x < a_{j+1}\}$
- Closure: $\overline{\mathcal{K}_j} = [a_j, a_{j+1}] = \{x; a_j \leq x \leq a_{j+1}\}$
- The domain is also an open set: $\Omega = (x_0, x_J)$ (as usual in the theory of PDE's)
- Common properties of grids (also 2D, 3D):
  - Cells do not overlap: $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$ resp. $\int_{\overline{\mathcal{K}_1} \cap \overline{\mathcal{K}_2}} 1 \, dV = 0$
  - Cells cover the entire domain: $\overline{\Omega} = \bigcup_{j=0}^{J-1} \overline{\mathcal{K}_j}$

## 1.2 Polynomial basis

We want to represent functions using a *basis*

$$\underline{\Phi} = (\Phi_0, \dots, \Phi_{N_p-1}), \tag{3}$$

then we can represent $u(\underline{x})$ as a linear combination:

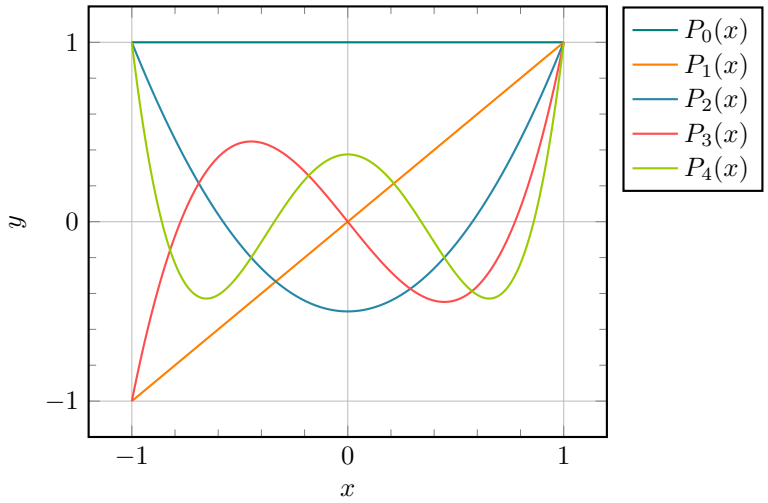
$$u(\underline{x}) = \sum_{n=0}^{N_p-1} \Phi_n(\underline{x}) \tilde{u}_n = \underline{\Phi} \cdot \underline{\tilde{u}} \tag{4}$$

**Examples for a polynomial basis in 1D:**

- Monomials up to degree $p$, i.e. $\Phi_n = x^n$, $n \leq p$ with $N_p = p + 1$ (in 1D)
- *Legendre-polynomials*, also called *modal polynomials* (solution of Legendre ODE)

$$P_0(x) = 1 \tag{5}$$
$$P_1(x) = x \tag{6}$$
$$\vdots \tag{7}$$

$$P_{n+1}(x) = \frac{1}{(n+1)}\left((2n+1)\cdot x\cdot P_n(x) - n\cdot P_{n-1}(x)\right) \qquad \text{for } n \geq 1 \tag{8}$$



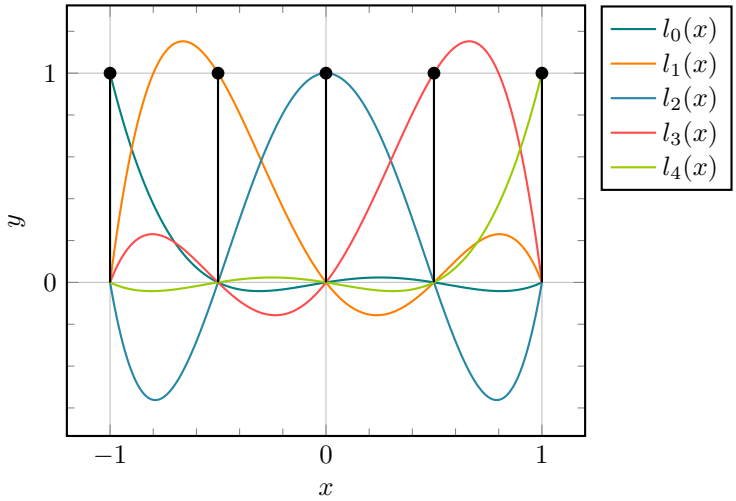

Distinct property: All functions orthogonal

$$\langle P_i(x)|P_j(x)\rangle_{[-1,1]} = 0 \qquad \forall i,j, i \neq j \tag{9}$$

- *Lagrange-polynomials*, also *nodal polynomials*, nodes $x_0, \ldots, x_{L-1}$:

$$\ell_n(x) = \prod_{\substack{0\leq l<L \\ l\neq n}} \frac{x - x_l}{x_n - x_l} \tag{10}$$

with nodal property: $\ell_n(x_l) = \delta_{nl}$, i.e. on each node, all polynomials but one vanish on each node. Example using five equidistant nodes:




**Remarks:** The choice of polynomials:

- has no impact on abstract numerical properties like convergence or stability (to be defined)
- has an impact on round-off curves; in exact architectures all would behave the same way
- has an impact on implementation efficiency: especially in 2D and 3D, some representations can be implemented more efficiently than others.

## 1.3 Approximation of some function $f(x)$ by a nodal interpolation

- Consider single cell $\mathcal{K} = (-1, 1)$ with *nodes* $x_0, \ldots, x_{L-1}$ and $-1 \leq x_l \leq 1$:


- We want to interpolate $f(x)$, using the nodal interpolation Ansatz, by a polynomial

$$g(x) = \sum_{n=0}^{N_p-1} \Phi_n(x) \cdot \tilde{g}_n = \underline{\Phi} \cdot \underline{\tilde{g}} \tag{11}$$

so that in all nodes $x_l$ the value of $g$ is equal to the value of $f$, i.e. for all $l$, $0 \leq l < L$: $g(x_l) = f(x_l)$, $\Rightarrow$ System of equations: for all $l$, $0 \leq l < L$:

$$\sum_{n=0}^{N_p-1} \Phi_n(x_l)\tilde{g}_n = f(x_l), \tag{12}$$

or, in matrix notation:

$$\underbrace{\begin{bmatrix} \Phi_0(x_0) & \ldots & \Phi_{N_p-1}(x_0) \\ \vdots & \ddots & \vdots \\ \Phi_0(x_L) & \ldots & \Phi_{N_p-1}(x_L) \end{bmatrix}}_{=P^{-1}_{\text{Nodal}}} \cdot \begin{bmatrix} \tilde{g}_0 \\ \vdots \\ \tilde{g}_{N_p-1} \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_{N_p-1}) \end{bmatrix} \tag{13}$$

This system has a unique solution for $N_p = L$ (if all nodes are pair-wise different, i.e. $x_l \neq x_i$ for $l \neq i$):

$$\tilde{g} = P_{\text{Nodal}} \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_{L-1}) \end{bmatrix} \tag{14}$$

For nodal polynomials, with the property $\Phi_n(x_k) = \delta_{nk}$, obviously: $P_{\text{Nodal}} = \underline{\underline{I}}$ .

## 1.4 A bit of theory: functional analysis crash-course

**Definition:** A norm $x \to u \mapsto |||u|||_* \in \mathbb{R}$ maps an object (e.g. a vector, matrix or a function) to a real number, i.e. has to fulfill the following properties:

- positivity: $|||u|||_* \geq 0$
- definiteness: $|||u|||_* = 0 \Rightarrow u = 0$
- homogeneity: $|||\alpha u|||_* = |\alpha| \; |||u|||_*$ for some $\alpha \in \mathbb{R}$
- fulfills the triangle inequality: $|||u + v|||_* \leq |||u|||_* + |||v|||_*$

**Examples for norms:**

- 2-norm of a vector: in 2D $\underline{x} = (x, y)$, $|\underline{x}| := \sqrt{x^2 + y^2}$; in $D$ dimensions $\underline{x} = (x_0, \ldots, x_{D-1})$, $|\underline{x}| := \left(\sum_{d=0}^{D-1} x_d^2\right)^{\frac{1}{2}}$
- 2-norm (or $L^2$-norm) of a function $u(x)$:

$$\|u\|_2 := \|u\|_{L^2(\mathbb{R})} := \left(\int_{\mathbb{R}} u^2(x) \mathrm{d}x\right)^{\frac{1}{2}} \tag{15}$$

- $q$-norm (or $L^q$-norm) of a function $u(x)$:

$$\|u\|_q := \|u\|_{L^q(\mathbb{R})} := \left(\int_{\mathbb{R}} u^q(x)\mathrm{d}x\right)^{\frac{1}{q}} \tag{16}$$

- Maximum-norm (or infinity-norm or $\infty$-norm) of a vector: in $D$ dimensions $\underline{x} = (x_0, \ldots, x_{D-1})$, $|\underline{x}|_\infty = \max\{|x_0|, \ldots, |x_{D-1}|\}$. (Remark: one can show that $|\underline{x}|_q$ converges to the infinity-norm for $q \to \infty$.)
- Maximum-norm of a function $u(x)$: in analog fashion.

**Definition: Linear, normed space.** A set $X$ is called

- a *linear space*, if: one has an addition operator $+$ and scalar multiplication which fulfill:

$$\forall \underline{u}, \underline{v} \in X,\ \forall \alpha, \beta \in \mathbb{R} : \alpha\underline{u} + \beta\underline{v} \in X$$

- a *normed space*, if for all $\underline{u}$ in $X$ the norm $\|\underline{u}\|_X$ is defined.
- $\underline{\Phi} = (\Phi_0, \ldots, \Phi_{N-1})$ is a basis of $X$, if every $\underline{u} \in X$ can be represented as $\underline{u} = \underline{\Phi} \cdot \tilde{u}$, where $\tilde{u}$ is a vector of real numbers. In this case, the dimension of $X$ is $N$.

**Examples for linear normed spaces.**

- Linear spaces: e.g. any sub-plane of $\mathbb{R}^D$ passing through the origin.
- Counter-example: affine sub-planes which do not pass through the origin, or curves.
- the space

$$L^2(\Omega) := \{f(\underline{x})|\ \|f\|_2 < \infty\}. \tag{17}$$

**Definition: Scalar products.** A mapping of two objects to a real number, $x \to \underline{f}, \underline{g} \mapsto \langle \underline{f}, \underline{g}\rangle \in \mathbb{R}$ is called a *scalar product* if, and only if it fulfills the following properties:

- linearity: $\langle \alpha\underline{f} + \beta\underline{h}|\underline{g}\rangle = \alpha\langle \underline{f}|\underline{g}\rangle + \beta\langle \underline{h}|\underline{g}\rangle$
- symmetry: $\langle \underline{f}|\underline{g}\rangle = \langle \underline{g}|\underline{f}\rangle$
- positive definiteness: $\langle \underline{f}|\underline{f}\rangle \geq 0$, $\langle \underline{f}|\underline{f}\rangle = 0$ if and only if $\underline{f} = 0$.
- (Remark: a scalar product induces a norm: $||\underline{f}|| := \sqrt{\langle \underline{f}|\underline{f}\rangle}$.)

**Definition: Hilbert space.** A *Hilbert space* is a *complete* space (each Cauchy sequence with respect to the norm on the space converges) whose norm is induced by a scalar product.

**Remarks and examples**

- space $\mathbb{R}^D$, $\underline{x} = (x_0, \ldots, x_{D-1})$, $\underline{y} = (y_0, \ldots, y_{D-1})$ scalar product $\langle \underline{x}|\underline{y}\rangle := \underline{x} \cdot \underline{y} := \sum_{d=0}^{D-1} x_d y_d$, this induces the 2-norm $|\underline{x}|_2 = \sqrt{\langle \underline{x}, \underline{x}\rangle}$.
- Polynomials $\Phi_n(x)$, $0 \leq n < N_p$ span a linear space:

$$\mathbb{P}_p(\{\mathcal{K}\}) = \left\{ g(x)|\ g(x) = \sum_n \Phi_n(x)\tilde{g}_n \right\} = \underline{\Phi} \cdot \mathbb{R}^{N_p} \tag{18}$$

- Dimension of $\mathbb{P}_p(\{\mathcal{K}\})$ is $N_p$: every member can be described uniquely by $N_p$ numbers, the *coordinates* $\tilde{g}_n$. Often, the coordinates are also called *coefficients* or *modes*.

- $\mathbb{P}_N((-1,1))$ is a subspace of $L^2((-1,1)) = \{f(x) | \int_{-1}^{1} f^2(x)\, dV < \infty\}$
- Scalar product on function spaces

$$\langle f|g\rangle := \int_{(x\in\Omega)} f(\underline{x}) g(\underline{x})\, dV \tag{19}$$

- Using this scalar product, $L^2(\Omega)$ becomes a Hilbert space of *infinite dimension*.
- Now, we have orthogonality of functions: $f(x)$ is orthogonal to $h(x)$, if and only if $\langle f|g\rangle = 0$

## 1.5 Approximation of some function $f(x)$ by error minimization

### General approach

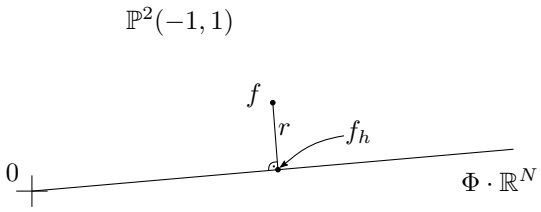
- Don't care about values at specific points, minimize the overall error/residual. We want to find the best approximation $f_h \in \mathbb{P}_p(\{\Omega\})$ to $f \in L^2(\Omega)$, in the following sense:

$$\int_\Omega \underbrace{(f_h(x) - f(x))}_{=:r(x)}{}^2\, dV = ||f_h - f||_2^2 \to \min \tag{20}$$

- Minimization is equivalent to requiring

$$\langle r(x)|\Phi_j\rangle = \langle f_h - f|\Phi_j\rangle \stackrel{!}{=} 0 \qquad \forall \Phi_j \tag{21}$$

- Verbally: Find $f_h(x)$ such that the residual $r(x)$ is orthogonal to our Ansatz space w.r.t. the scalar product $\langle\cdot|\cdot\rangle$




- Our Ansatz $f_h(x) = \sum_{n=0}^{N_p-1} \Phi_n(x) \cdot \tilde{f}_n$ leads to the linear system

$$\underbrace{\begin{bmatrix} \langle\Phi_0|\Phi_0\rangle & \dots & \langle\Phi_0|\Phi_{N_p-1}\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi_{N_p-1}|\Phi_0\rangle & \dots & \langle\Phi_{N_p-1}|\Phi_{N_p-1}\rangle \end{bmatrix}}_{=:\langle\underline{\Phi}^T|\underline{\Phi}\rangle} \cdot \begin{pmatrix} \tilde{f}_0 \\ \vdots \\ \tilde{f}_{N_p-1} \end{pmatrix} = \underbrace{\begin{pmatrix} \langle f|\Phi_0\rangle \\ \vdots \\ \langle f|\Phi_{N_p-1}\rangle \end{pmatrix}}_{=:\langle f|\underline{\Phi}^T\rangle}, \tag{22}$$

  i.e. we obtain the coordinate vector $\underline{\tilde{f}}$ of $f_h$ as

$$\underline{\tilde{f}} = \left(\langle\underline{\Phi}^T|\underline{\Phi}\rangle\right)^{-1} \langle f|\underline{\Phi}^T\rangle \tag{23}$$

- This defines the projection operator

$$L^2(\Omega) \ni f \mapsto \pi_p(f) = f_h \in \mathbb{P}_p(\mathfrak{K}_h) \tag{24}$$

  which is given uniquely by the property

$$\langle f_h - f|g_h\rangle = 0 \qquad \forall g_h \in \mathbb{P}_p(\mathfrak{K}_h) \tag{25}$$

  and can explicitly be computed for a given basis $\underline{\Phi}$

$$f \mapsto \pi_p(f) := \underline{\Phi} \underbrace{\left(\left(\langle\underline{\Phi}^T|\underline{\Phi}\rangle\right)^{-1} \langle f|\underline{\Phi}^T\rangle\right)}_{=\underline{\tilde{f}}} = f_h \tag{26}$$

**Analogy to Euclidian vector algebra**

- Consider the following problem: Find the point $\underline{x}^* \in \mathbb{R}^3$ on a given plane $E$ with minimum distance to a given point $\underline{x} \in \mathbb{R}^3$
- In other words: We are looking for the $L^2$-projection of $\underline{x}$ onto $E$!
- Let $E$ be spanned by two linear independent *basis* vectors $\underline{v}_0, \underline{v}_1 \in \mathbb{R}^3$. We can then reuse equation (21) to obtain the system of equations

$$\langle \underline{x} - \underline{x}^* | \underline{v}_0 \rangle = 0 \tag{27}$$
$$\langle \underline{x} - \underline{x}^* | \underline{v}_1 \rangle = 0 \tag{28}$$

  which can be rewritten as

$$\underline{x}^* \cdot \underline{v}_0 = \underline{x} \cdot \underline{v}_0 \tag{29}$$
$$\underline{x}^* \cdot \underline{v}_1 = \underline{x} \cdot \underline{v}_1 \tag{30}$$

- We now have to introduce our Ansatz

$$\underline{x}^* = a\underline{v}_0 + b\underline{v}_1 \tag{31}$$

  with yet unknown scalars $a$ and $b$
- This directly leads us to the system

$$\begin{bmatrix} \underline{v}_0 \cdot \underline{v}_0 & \underline{v}_1 \cdot \underline{v}_0 \\ \underline{v}_0 \cdot \underline{v}_1 & \underline{v}_1 \cdot \underline{v}_1 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \underline{x} \cdot \underline{x}_0 \\ \underline{x} \cdot \underline{x}_1 \end{pmatrix} \tag{32}$$

  that can be solved for $a$ and $b$. This is the direct equivalent of system (22)!

**Idempotence of the projection operator**

- Equation (25) resp. (26) define the $L^2$-*orthogonal projection* of $f$ onto the approximation space $\mathbb{P}_p(\Omega)$
- General definition of a projector operator: A mapping $\pi$ is called a projector if, and only if it is linear and idempotent ($\pi^2 = \pi$). (Applying the operator twice is the same as applying it once.)

*Proof of the idempotency:* We want to show that $\pi_p(\pi_p(f)) = \pi_p(f)$. For $f_h := \pi_p(f) = \phi \cdot \tilde{f}$, we have the relation

$$\langle \phi^T, \phi \rangle \tilde{f} = \langle \phi^T, f \rangle \tag{33}$$

For the projection of the projection, i.e. for $\pi_p(\pi_p(f)) = \pi_p(f_h) = \phi \cdot \tilde{\tilde{f}}$, we have the relation

$$\langle \phi^T, \phi \rangle \tilde{\tilde{f}} = \langle \phi^T, f_h \rangle \tag{34}$$

by inserting (33) into (34), one gets

$$\langle \phi^T, \phi \rangle \tilde{\tilde{f}} = \langle \phi^T, \phi \tilde{f} \rangle = \langle \phi^T, \phi \rangle \tilde{f} \tag{35}$$

Since $\langle \phi^T, \phi \rangle$ is invertible, we reason that $\tilde{\tilde{f}} = \tilde{f}$, which proofs that the $L^2$-projection is idempotent.

**Approximation error of the projection**

- Application of the Bramble-Hilbert lemma: Let $f$ be $p$ times continuously differentiable. Then

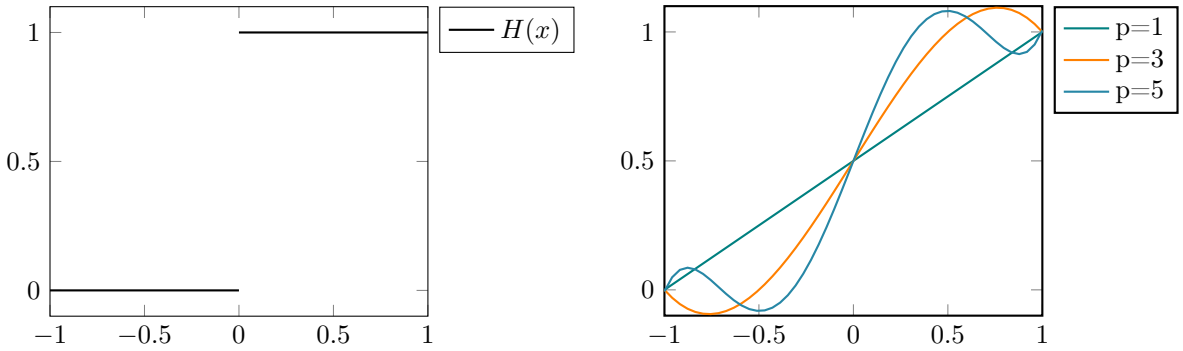
$$\| f(x) - \pi_p(f(x)) \|_{L^2(\Omega)} \leq C \cdot h^{p+1} \tag{36}$$

  with a constant $C$ that depends on $f$ but not on $h$.
- Verbally: If $f$ is sufficiently smooth, the approximation error is of order $O(h^{p+1})$, which is one of the major motivations for using higher order methods

- The differentiability assumption is essential. If $f$ is non-smooth, the so-called *Gibbs phenomenon* occurs, which is one of the major drawbacks of higher order methods
- Example: Approximation of the Heaviside function

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases} \tag{37}$$

by polynomials of various degrees:




Note that the $L^2$ error *does* decrease when increasing $p$, even though extremely slowly

- The magnitude of the modes $\tilde{f}_n$ decays rapidly, but only if $f$ is sufficiently smooth (cf. exercise 2). This can also be used as a smoothness detector for problems involving shocks or other discontinuities.

## 1.6 Extension to multiple dimensions

- From now on: Consider a grid $\mathfrak{K}_h$ consisting of $D$-dimensional grid cells $\mathcal{K}_i$, $i = 0, ..., J-1$
- Common cell types: triangles, quadrilaterals, tetrahedrons, hexahedrons, prisms
- Different choices for the characteristic mesh size $h$ exist. Common choices are the inner/outer cell diameter, or the length of longest/shortest edge of the cell
- The definitions and approximation results naturally generalize to this case, e.g. the Ansatz

$$f_h(\underline{x}) = \sum_{n=0}^{N_p-1} \Phi_n(\underline{x}) \tilde{g}_n \tag{38}$$

and the projection

$$\underline{f} = P_{\text{Modal}} \left\langle f | \underline{\Phi}^T \right\rangle \tag{39}$$

- Basis functions on these element can be defined in different ways. More on this topic in Section 5. For now, we assume that a suitable basis as given
- Example: Third order monomial basis for $D = 2$:

$$\underline{\Phi} = \left(1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3\right)$$

- Main difference to one-dimensional case: Size of the polynomial space $\mathbb{P}_P(\{\mathcal{K}\})$ for $\mathcal{K} \in \mathbb{R}^D$ is given by

| P | D = 1 | D = 2 | D = 3 |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 6 | 10 |
| 3 | 4 | 10 | 20 |
| 4 | 5 | 15 | 35 |

These numbers can be obtained by counting the entries of the first $k+1$ rows of Pascal's triangle ($D = 2$) or the first $k+1$ layers in Pascal's pyramid ($D = 3$)

# 2 DG for first order problems

## 2.1 Motivation

- In Section 1.5, we tried to find the *best* approximation $f_h \in \mathbb{P}_p(\{\mathcal{K}\})$ of a function $f \in L^2(\{\mathcal{K}\})$ by error minimization
- Main result: Require that the residual

$$r(\underline{x}) = f(\underline{x}) - f_h(\underline{x}) \tag{40}$$

is orthogonal to approximation space, i.e. that

$$\langle r(\underline{x}) | \Phi_j \rangle_{\mathcal{K}} = 0 \qquad \forall \Phi_j \tag{41}$$

- Idea: Follow the same idea for some conservation law

$$\frac{\partial c}{\partial t} + \nabla \cdot \underline{f}(c) = 0 \tag{42}$$

with suitable initial and boundary conditions for the residual

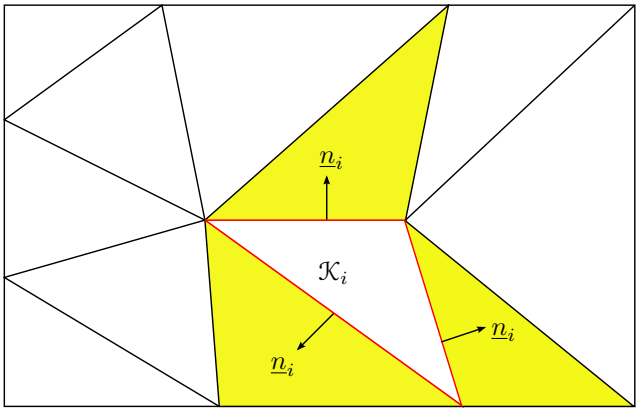
$$r(c_h) = \frac{\partial c_h}{\partial t} + \nabla \cdot \underline{f}(c_h) \tag{43}$$

for some approximation $c_h$ of $c$. That is, find the best approximation $c_h \in \mathbb{P}_p(\{\mathcal{K}\})$ such that

$$\langle r(c_h) | \Phi_j \rangle_{\mathcal{K}} = 0 \qquad \forall \Phi_j \tag{44}$$

in the given cell

## 2.2 Requirement: Computational grid




- Extending the definitions in section 1.1 to multiple, we define a computational grid $\mathfrak{K}_h := \{\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_{J-1}\}$ as a set of $D$-dimensional mesh cells $\mathcal{K}_i$ (usually polytopes) that cover the computational domain $\Omega$
- The *edges* $\mathfrak{E}_h = \{\mathcal{E}_e\}_{e=0,\ldots,E-1}$ of this grid are given by the boundaries of these mesh cells (lines in the picture above), i.e. $\partial\mathcal{K}_i$. That is, an edge $\mathcal{E}_e$ is either a zero-dimensional (1D grid), a one-dimensional (2D grid) or a two-dimensional (3D grid) entity.
- On the other hand, the *vertices* of cell $\mathcal{K}_i$ (intersections of lines in the picture above) are always zero-dimensional entities (that is, points)
- We call a cell $\mathcal{K}_j$ a *neighbor* of $\mathcal{K}_i$ if both cells share at least one *edge*, i.e. if

$$\exists \mathcal{E}_e \in \mathfrak{E}_h : \quad \overline{\mathcal{K}}_i \cap \overline{\mathcal{K}}_j = \mathcal{E}_e. \tag{45}$$

In the picture above, the yellow cells are the neighbors of $\mathcal{K}_i$ since they are connected to $\mathcal{K}_i$ via the red edges. Moreover, the maximum number of neighbors of a cell solely depends on the number of edges of that cell, *not* on the number of vertices shared with other cells.

- An edge $\mathcal{E}_e$ is called a *boundary edge* if $\mathcal{E}_e \subset \partial\Omega$, i.e. if it coincides with a part of the boundary of the computational domain. All other edges are called *inner edges* that connect exactly two cells. That is, we distinguish the set of inner edges
$$\mathfrak{E}_h^i = \{\mathcal{E} \in \mathfrak{E}_h : \mathcal{E} \cap \partial\Omega = \emptyset\} \tag{46}$$
and the set of boundary edges
$$\mathfrak{E}_h^b = \{\mathcal{E} \in \mathfrak{E}_h : \mathcal{E} \cap \partial\Omega = \mathcal{E}\}, \tag{47}$$
where obviously $\mathfrak{E}_h^i \cup \mathfrak{E}_h^b = \mathfrak{E}_h$ and $\mathfrak{E}_h^i \cap \mathfrak{E}_h^b = \emptyset$ hold

- For each cell $\mathcal{K}_i$, we define the associated normal vectors $\underline{n}_i$ on $\partial\mathcal{K}_i$ such that they are pointing outwards. As a result, the direction of a normal vector is not unique on some edge $\mathcal{E}_e$, but depends on the particular cell under consideration.

- Note: The above definitions are sensible for deriving the *local* form of a DG scheme, which we will outline in the following subsections. This local form is extremely helpful for understanding the general idea of DG methods. However, efficient implementations of DG schemes should follow a different approach. Some of the associated issue will thus be discussed in Section 5

- Each grid cell $\mathcal{K}_i$ is associated with its own, *cell-local* basis $\underline{\Phi}_i = (\Phi_{i,n})_{n=0,\dots,N_p-1}$. The locality of the basis stems from the fact each basis function is only non-zero in its associated cell, i.e.
$$\mathrm{supp}(\Phi_{i,j}) = \bar{\mathcal{K}}_i, \tag{48}$$
where $\mathrm{supp}(\Phi_{i,j})$ denotes the *support* of $\Phi_{i,j}$

- The locality of the basis functions often allows us to switch easily between global and local expressions, for example
$$c_h = \sum_{i=0}^{J-1} c_{h,i} \tag{49}$$
and
$$\langle g|\Phi_{i,j}\rangle_\Omega = \langle g|\Phi_{i,j}\rangle_{\mathcal{K}_i} \tag{50}$$
for a generic function $g$

## 2.3 Semi-discrete weak formulation

### Derivation

- We start from the conservation law
$$\frac{\partial c}{\partial t} + \nabla \cdot \underline{f}(c) = 0 \tag{42, repeated}$$

- As in equation (21), we require that the residual
$$r(c_h) \coloneqq \frac{\partial c_h}{\partial t} + \nabla \cdot \underline{f}(c_h) \tag{51}$$
is orthogonal to some *test function* $\Phi_{i,j}$. This leads to
$$\langle r(c_h)|\Phi_{i,j}\rangle_{\mathcal{K}_i} = \left\langle \frac{\partial c_h}{\partial t} + \nabla \cdot \underline{f}(c_h) \middle| \Phi_{i,j} \right\rangle_{\mathcal{K}_i} \tag{52}$$
$$= \left\langle \frac{\partial c_h}{\partial t} \middle| \Phi_{i,j} \right\rangle_{\mathcal{K}_i} + \left\langle \nabla \cdot \underline{f}(c_h) \middle| \Phi_{i,j} \right\rangle_{\mathcal{K}_i} \tag{53}$$
$$\stackrel{!}{=} 0 \quad \forall \Phi_{i,j} \tag{54}$$
in each cell $\mathcal{K}_i$

- Here, an index $j \in [0, \dots, N_p - 1]$ corresponds to the $j$-th *test function* $\Phi_{i,j}$ in cell $\mathcal{K}_i$

- Reminder: Inserting the definition of the scalar product, this is equivalent to

$$0 \overset{!}{=} \left\langle \frac{\partial c_h}{\partial t} + \nabla \cdot \underline{f}(c_h) \middle| \Phi_{i,j} \right\rangle_{\mathcal{K}_i} \tag{55}$$

$$= \int_{\mathcal{K}_i} \left( \frac{\partial c_h}{\partial t} + \nabla \cdot \underline{f}(c_h) \right) \Phi_{i,j} \, dV \tag{56}$$

$$= \underbrace{\int_{\mathcal{K}_i} \frac{\partial c_h}{\partial t} \Phi_{i,j} \, dV}_{\text{Temporal term}} + \underbrace{\int_{\mathcal{K}_i} \left( \nabla \cdot \underline{f}(c_h) \right) \Phi_{i,j} \, dV}_{\text{Spatial term}} \qquad \forall \Phi_{i,j} \tag{57}$$

- For the spatial term in (57), we use integration by parts to obtain

$$\int_{\mathcal{K}_i} \left( \nabla \cdot \underline{f}(c_h) \right) \Phi_{i,j} \, dV = \underbrace{\int_{\partial \mathcal{K}_i} \left( \underline{f}(c_h) \cdot \underline{n}_i \right) \Phi_{i,j} \, dA}_{\text{Surface term}} - \underbrace{\int_{\mathcal{K}_i} \underline{f}(c_h) \cdot \nabla \Phi_{i,j} \, dV}_{\text{Volume term}} \tag{58}$$

- In sum, we have

$$\underbrace{\int_{\mathcal{K}_i} \frac{\partial c_h}{\partial t} \Phi_{i,j} \, dV}_{\text{Temporal term}} + \underbrace{\int_{\partial \mathcal{K}_i} \left( \underline{f}(c_h) \cdot \underline{n}_i \right) \Phi_{i,j} \, dA}_{\text{Surface term}} - \underbrace{\int_{\mathcal{K}_i} \underline{f}(c_h) \cdot \nabla \Phi_{i,j} \, dV}_{\text{Volume term}} = 0 \qquad \forall \Phi_{i,j} \tag{59}$$

**Galerkin Ansatz**

- We did not yet specify how we approximate $c_h$. Galerkin approach: Use identical trial/Ansatz and test functions by defining the local solution

$$c_{h,i}(\underline{x}, t) = \sum_{n=0}^{N_p - 1} \Phi_{i,n}(\underline{x}) \, \tilde{c}_{i,n}(t) = \underline{\Phi}_i(\underline{x}) \, \underline{\tilde{c}_i}(t) \tag{60}$$

  (which also defines the global solution according to (49)).

- The yet unknown coefficients $\tilde{c}_{i,n} = \tilde{c}_{i,n}(t)$ are called the degrees of freedom (DOF)
- Important observations: The coefficients only depend on time, while the basis only depends on $\underline{x}$. This allows us to rearrange the integrals in the following paragraphs
- For each cell, we have $N_p$ DOF (one for each coefficient, respectively trial function) and $N_p$ equations (one for each test function). The total number of DOF/equations in the system is thus $J \cdot N_p$, where $J$ is the total number of cells.
- In the three following paragraphs, we will discuss the results of inserting (60) into (59) for each term separately

**Temporal term**

- Inserting this into the temporal term is straightforward:

$$\int_{\mathcal{K}_i} \frac{\partial c}{\partial t} \Phi_{i,j} \, dV \approx \int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j} \, dV \tag{61}$$

$$= \int_{\mathcal{K}_i} \frac{\partial}{\partial t} \left( \sum_{n=0}^{N_p - 1} \Phi_{i,n} \, \tilde{c}_{i,n} \right) \Phi_{i,j} \, dV \tag{62}$$

$$= \int_{\mathcal{K}_i} \sum_{n=0}^{N_p - 1} \frac{\partial \tilde{c}_{i,n}}{\partial t} \Phi_{i,n} \Phi_{i,j} \, dV \tag{63}$$

$$= \sum_{n=0}^{N_p-1} \frac{\partial \tilde{c}_{i,n}}{\partial t} \underbrace{\int_{\mathcal{K}_i} \Phi_{i,n} \Phi_{i,j} \, dV}_{=:(\underline{\underline{M}}_i)_{j,n}} \tag{64}$$

- The *mass matrix* $\underline{\underline{M}}_i$ of cell $\mathcal{K}_i$ only depends on the choice of the *cell-local* basis functions:

$$\underline{\underline{M}}_i = \begin{bmatrix} \langle \Phi_{i,0} | \Phi_{i,0} \rangle_{\mathcal{K}_i} & \langle \Phi_{i,0} | \Phi_{i,1} \rangle_{\mathcal{K}_i} & \cdots \\ \langle \Phi_{i,1} | \Phi_{i,0} \rangle_{\mathcal{K}_i} & \langle \Phi_{i,1} | \Phi_{i,1} \rangle_{\mathcal{K}_i} & \cdots \\ \vdots & & \ddots \end{bmatrix} \tag{65}$$

  In particular, $\underline{\underline{M}}_i$ does not depend on neighboring cells, which implies that the global mass matrix is block-diagonal:

$$\underline{\underline{M}} = \begin{bmatrix} \underline{\underline{M}}_0 & 0 & 0 & \cdots \\ 0 & \underline{\underline{M}}_1 & 0 & \cdots \\ 0 & 0 & \underline{\underline{M}}_2 & \\ \vdots & \vdots & & \ddots \end{bmatrix} \tag{66}$$

- Example for $p = 1$ with $J = 3$ cells:

$$\underline{\underline{M}} = \begin{bmatrix} \langle \Phi_{0,0} | \Phi_{0,0} \rangle_{\mathcal{K}_0} & \langle \Phi_{0,0} | \Phi_{0,1} \rangle_{\mathcal{K}_0} & 0 & 0 & 0 & 0 \\ \langle \Phi_{0,1} | \Phi_{0,0} \rangle_{\mathcal{K}_0} & \langle \Phi_{0,1} | \Phi_{0,1} \rangle_{\mathcal{K}_0} & 0 & 0 & 0 & 0 \\ 0 & 0 & \langle \Phi_{1,0} | \Phi_{1,0} \rangle_{\mathcal{K}_1} & \langle \Phi_{1,0} | \Phi_{1,1} \rangle_{\mathcal{K}_1} & 0 & 0 \\ 0 & 0 & \langle \Phi_{1,1} | \Phi_{1,0} \rangle_{\mathcal{K}_1} & \langle \Phi_{1,1} | \Phi_{1,1} \rangle_{\mathcal{K}_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \langle \Phi_{2,0} | \Phi_{2,0} \rangle_{\mathcal{K}_2} & \langle \Phi_{2,0} | \Phi_{2,1} \rangle_{\mathcal{K}_2} \\ 0 & 0 & 0 & 0 & \langle \Phi_{2,1} | \Phi_{2,0} \rangle_{\mathcal{K}_2} & \langle \Phi_{2,1} | \Phi_{2,1} \rangle_{\mathcal{K}_2} \end{bmatrix} \tag{67}$$

- Obviously, the mass matrix is symmetric
- All in all, we can write

$$\int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j} \, dV = \sum_{n=0}^{N_p-1} \frac{\partial \tilde{c}_{i,n}}{\partial t} (\underline{\underline{M}}_i)_{j,n} = \underline{\underline{M}}_i \frac{\partial \underline{\tilde{c}}_i}{\partial t} \tag{68}$$

**Volume term**

- The volume term yields

$$\int_{\mathcal{K}_i} \underline{f}(c) \cdot \nabla \Phi_{i,j} \, dV \approx \int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla \Phi_{i,j} \, dV \tag{69}$$

  which cannot be simplified significantly in general

- Observation: Just as the temporal term, it only depends on *cell-local* values
- *Special case*: $\underline{f}$ is linear (or linearized), e.g. $\underline{f}(c) = \underline{u}c$ with a constant vector $\underline{u} \in \mathbb{R}^D$. Interpretation: Advection of a scalar concentration $c$ in a given velocity field. Then

$$\int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla \Phi_{i,j} \, dV = \int_{\mathcal{K}_i} \underline{u} c_h \cdot \nabla \Phi_{i,j} \, dV \tag{70}$$

$$= \int_{\mathcal{K}_i} \underline{u} \left( \sum_{n=0}^{N_p-1} \Phi_{i,n} \tilde{c}_{i,n} \right) \cdot \nabla \Phi_{i,j} \, dV \tag{71}$$

$$= \sum_{n=0}^{N_p-1} \tilde{c}_{i,n} \underbrace{\int_{\mathcal{K}_i} \underline{u} \Phi_{i,n} \cdot \nabla \Phi_{i,j} \, dV}_{=:(\underline{\underline{S}}_i)_{j,n}} \tag{72}$$

$$\tag{73}$$

- Like the mass matrix, the *stiffness matrix* $\underline{\underline{S}}_i$ is block-diagonal. However, each block of $\underline{\underline{S}}_i$ is singular. Using the example of a monomial basis, this directly follows from

$$(\underline{\underline{S}}_i)_{n,0} = \int_{\mathcal{K}_i} \underline{u}\Phi_{i,n} \cdot \nabla\Phi_{i,0}\, dV \tag{74}$$

$$= \int_{\mathcal{K}_i} \underline{u}\Phi_{i,n} \cdot \nabla 1\, dV \tag{75}$$

$$= 0 \qquad \forall\Phi_{i,n}, \tag{76}$$

  but this result also extends to the general case

**Surface term**

- The surface term is given by

$$\int_{\partial\mathcal{K}_i} \left(\underline{f}(c_h) \cdot \underline{n}_i\right) \Phi_{i,j}\, dA \tag{77}$$

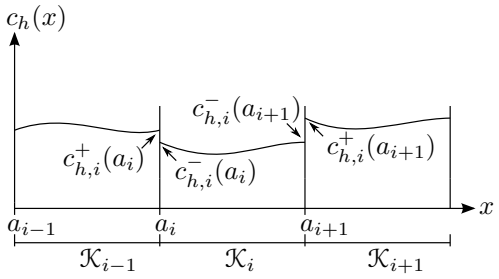
- Inserting Ansatz into (59) is problematic since $\partial\mathcal{K}_i$ is shared by other cells, but we do not enforce continuity! Thus, there is a difference between the inner value

$$c^-_{h,i} = c^-_{h,i}(\underline{x}, t) \coloneqq \lim_{\epsilon\to 0^+} c_h(\underline{x} - \epsilon\underline{n}_i, t) \tag{78}$$

  and the outer value

$$c^+_{h,i} = c^+_{h,i}(\underline{x}, t) \coloneqq \lim_{\epsilon\to 0^+} c_h(\underline{x} + \epsilon\underline{n}_i, t) \tag{79}$$

  on all edges that do not coincide with a domain boundary




- Solution: Introduce a *numerical flux* function

$$\hat{f}(c^-_{h,i}, c^+_{h,i}, \underline{n}_i) \approx \underline{f}(c_h) \cdot \underline{n}_i \tag{80}$$

  that defines a unique value on $\partial\mathcal{K}_i$

- At boundary edges, $c^+_{h,i}(\underline{x}, t)$ is given by some boundary condition (the section 2.5)

- Most simple (but unstable!) example: Central flux

$$\hat{f}(c^-_{h,i}, c^+_{h,i}, \underline{n}_i) = \underline{f}\left(\frac{c^-_{h,i} + c^+_{h,i}}{2}\right) \cdot \underline{n}_i \tag{81}$$

- The numerical flux couples the DOF of neighboring cells. It should satisfy certain mathematical and physical properties, which will be discussed in detail in Section 3

- In sum, we have

$$\int_{\partial\mathcal{K}_i} \left(\underline{f}(c_h) \cdot \underline{n}_i\right) \Phi_{i,j}\, dA \approx \int_{\partial\mathcal{K}_i} \hat{f}(c^-_{h,i}, c^+_{h,i}, \underline{n}_i)\Phi_{i,j}\, dA \tag{82}$$

**Putting everything together**

- The semi-discrete weak formulation the scalar conservation law (42) is given by

$$\int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j} \, dV + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \Phi_{i,j} \, dA - \int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla \Phi_{i,j} \, dV = 0 \qquad \forall \Phi_{i,j} \tag{83}$$

- This system of equations has been discretized in space but not in time, hence it is called *semi-discrete*
- Commonly, we will abbreviate this as

$$\underline{\underline{M}}_i \frac{\partial \underline{\tilde{c}}_i}{\partial t} + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \Phi_{i,j} \, dA - \int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla \Phi_{i,j} \, dV = 0 \qquad \forall \Phi_{i,j} \tag{84}$$

  using the mass matrix $\underline{\underline{M}}_i$. This can be written even shorter as

$$\underline{\underline{M}}_i \frac{\partial \underline{\tilde{c}}_i}{\partial t} + \underline{f}_{h,i} = 0 \qquad \forall \mathcal{K}_i \tag{85}$$

  by defining the *discrete flux* or *discrete operator*

$$\underline{f}_{h,i} = \underline{f}_{h,i}(c_h) \coloneqq \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \Phi_{i,j} \, dA - \int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla \Phi_{i,j} \, dV \tag{86}$$

## 2.4 Strong form and flux continuity

- From (83), we can obtain a *strong form* of the semi-discrete formulation by integrating the volume by parts once again, but this time using only inner values in the corresponding surface term:

$$\int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j} \, dV + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \Phi_{i,j} \, dA - \left( \int_{\partial \mathcal{K}_i} (\underline{f}(c_{h,i}^-) \cdot \underline{n}_i) \Phi_{i,j} \, dA - \int_{\mathcal{K}_i} (\nabla \cdot \underline{f}(c_{h,i})) \Phi_{i,j} \, dV \right) = 0 \tag{87}$$

- Rearranging yields

$$\int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j} \, dV + \int_{\mathcal{K}_i} (\nabla \cdot \underline{f}(c_{h,i})) \Phi_{i,j} \, dV + \int_{\partial \mathcal{K}_i} \left( \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) - \underline{f}(c_{h,i}^-) \cdot \underline{n}_i \right) \Phi_{i,j} \, dA = 0 \tag{88}$$

- The strong form is hardly used in practice, but allows for an instructive interpretation of the role of the surface term: The first two terms are zero if $c_{h,i}$ is a *local* solution of the PDE, while the last term *weakly* enforces *flux continuity* across cell boundaries (see *consistency* in section 3.2).

## 2.5 Incorporation of boundary conditions

- For first order problems, we are in general only allowed to enforce Dirichlet boundary conditions. The concrete PDE under consideration dictates the actual set of required boundary conditions
- For example, consider the scalar conservation law discussed above. On inflow boundaries, we *have* to specify a boundary condition, while we *must not* define a boundary value at outflow boundaries
- The situation is even more complex for systems of PDEs. When using the Euler equations, for example, the admissible number of boundary values on a particular boundary additionally depends on the flow conditions (subsonic vs. supersonic). This issue is however out of the scope of this lecture and we will not discuss it any further here
- In any case, definition (79) has to be extended in order to be valid on boundary edges. Thus, we define

$$c_{h,i}^+(\underline{x}, t) = \begin{cases} \lim\limits_{\epsilon \to 0^+} c_h(\underline{x} + \epsilon \underline{n}_i, t) & \text{if } \underline{x} \in \mathfrak{E}_h^i \\ c_B(\underline{x}, t) & \text{if } \underline{x} \in \mathfrak{E}_h^b \end{cases} \tag{89}$$

  with some boundary value $c_B(\underline{x}, t)$ that has to be specified

- If $c_B(\underline{x}, t)$ is a Dirichlet condition to be enforced on a particular edge, we thus simply insert it into the numerical flux function. Otherwise, i.e. if no boundary condition should be given, we set $c_B(\underline{x}, t) = c_{h,i}(\underline{x}, t)$.
- We will see in Section 3.3, that this distinction follows naturally from the concept of *Riemann solvers*

# 3 Numerical fluxes

- In section 2.3, we have introduced the numerical flux function

$$\hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \approx \underline{f}(c_{h,i}) \cdot \underline{n}_i \qquad \text{(80, repeated)}$$

  that defines a unique flux across some inner edge $\mathcal{E}_e \in \mathfrak{E}_h^i$
- Obviously, $\hat{f}$ has to possess some mathematical as well as physical properties in order to be useful in practice
- In principle, the following requirements only need to be satisfied on a set of *admissible states*. Considering linear advection of some concentration, for example, the admissible set is $[0, 1]$ since only such concentrations are physical. However, we will ignore this subtlety in the following for the sake of simplicity.
- This chapter is largely based on (Di Pietro and Ern, 2012, Chapter 3.2)

## 3.1 Requirements

- *Lipschitz continuity*:

$$\exists C_a \in \mathbb{R} : \left| \hat{f}(a_1, b, \underline{n}) - \hat{f}(a_2, b, \underline{n}) \right| \leq C_a \left| a_1 - a_2 \right| \qquad \forall a_1, a_2 \in \mathbb{R} \tag{90}$$

  and

$$\exists C_b \in \mathbb{R} : \left| \hat{f}(a, b_1, \underline{n}) - \hat{f}(a, b_2, \underline{n}) \right| \leq C_b \left| b_1 - b_2 \right| \qquad \forall b_1, b_2 \in \mathbb{R} \tag{91}$$

- *Consistency*:

$$\hat{f}(a, a, \underline{n}) = \underline{f}(a) \cdot \underline{n} \qquad \forall a \in \mathbb{R} \tag{92}$$

- *Conservativity*:

$$\hat{f}(a, b, \underline{n}) = -\hat{f}(b, a, -\underline{n}) \qquad \forall a, b \in \mathbb{R} \tag{93}$$

- *Monotonicity*:

$$\frac{\partial \hat{f}(a, b, \underline{n})}{\partial a} \geq 0 \quad \wedge \quad \frac{\partial \hat{f}(a, b, \underline{n})}{\partial b} \leq 0 \qquad \forall a, b, \underline{n} \tag{94}$$

## 3.2 Interpretation

### Lipschitz continuity

- Lipschitz continuity is a technical assumption which is required to be able to prove the existence and uniqueness of a solution of the semi-discrete system (83)
- All relevant choices for $\hat{f}$ satisfy this property, so we will not discuss it here any further

**Consistency**

- Consider the strong form of the discrete system:

$$\int_{\mathcal{K}_i} \frac{\partial c_{h,i}}{\partial t} \Phi_{i,j}\, dV + \int_{\mathcal{K}_i} (\nabla \cdot \underline{f}(c_{h,i})) \Phi_{i,j}\, dV + \int_{\partial \mathcal{K}_i} \left( \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) - \underline{f}(c_{h,i}^-) \cdot \underline{n}_i \right) \Phi_{i,j}\, dA = 0. \quad (88, \text{repeated})$$

  If the *consistency* condition (92) is fulfilled, it directly follows that the surface term vanishes if $c_h$ is continuous across cell boundaries ($c_{h,i}^- = c_{h,i}^+$). In other words, the term is only *active* if the fluxes across edges are *not* continuous in order to enforce flux continuity

- As a consequence, we have

$$\langle r(c_h) | \Phi_{i,j} \rangle_{\mathcal{K}_i} = 0 \quad \forall \Phi_{i,j} \tag{95}$$

  if $\hat{f}$ is consistent and $c$ is a continuous exact solution of (42)

**Conservativity**

- The scalar conservation law (42) is in *conservative form*. Integrating over the problem domain $\Omega$ leads to

$$\int_{\Omega} \frac{\partial c}{\partial t}\, dV + \int_{\Omega} \nabla \cdot \underline{f}(c)\, dV = 0 \tag{96}$$

  and, assuming $c$ is smooth, further to

$$\frac{\partial}{\partial t} \left( \int_{\Omega} c\, dV \right) + \int_{\partial \Omega} \underline{f}(c) \cdot \underline{n}_{\partial \Omega}\, dA = 0 \tag{97}$$

  after applying the Gauss theorem. We see that the total amount of $c$ only changes due to fluxes across the domain boundary, hence the term *conservative form*

- Taking w.l.o.g. $\Phi_{i,0} = 1$ in system (83) gives

$$\int_{\mathcal{K}_{h,i}} \frac{\partial c_h}{\partial t}\, 1\, dV + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i)\, 1\, dA - \underbrace{\int_{\mathcal{K}_i} \underline{f}(c_{h,i}) \cdot \nabla 1\, dV}_{=0} = 0 \qquad \forall \mathcal{K}_i, \tag{98}$$

  which can be rewritten as

$$\frac{\partial}{\partial t} \int_{\mathcal{K}_i} c_{h,i}\, dV + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i)\, dA = 0 \qquad \forall \mathcal{K}_i, \tag{99}$$

  which is the discrete *local* equivalent of (97). However, this does *not* imply *global* conservativity

- Now consider a single edge $\mathcal{E} = \partial \mathcal{K}_i \cap \partial \mathcal{K}_k$. Then *conservativity* of the flux leads to

$$\int_{\mathcal{E}} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i)\, dA \overset{(93)}{=} - \int_{\mathcal{E}} \hat{f}(c_{h,i}^+, c_{h,i}^-, -\underline{n}_i)\, dA \tag{100}$$

$$= - \int_{\mathcal{E}} \hat{f}(\tilde{c}_{h,k}^-, \tilde{c}_{h,k}^+, \underline{n}_k)\, dA. \tag{101}$$

- Consequently, summing (99) over *all* elements gives

$$\sum_{\mathcal{K}_i} \left( \frac{\partial}{\partial t} \int_{\mathcal{K}_i} c_{h,i}\, dV + \int_{\partial \mathcal{K}_i} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i)\, dA \right) \tag{102}$$

$$= \sum_{\mathcal{K}_i} \left( \frac{\partial}{\partial t} \int_{\mathcal{K}_i} c_{h,i}\, dV + \sum_{\mathcal{E} \in \partial \mathcal{K}_i} \int_{\mathcal{E}} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i)\, dA \right) \tag{103}$$

$$\overset{(49)}{=} \frac{\partial}{\partial t} \int_\Omega c_h \, dV + \sum_{\mathcal{K}_i} \left( \sum_{\mathcal{E} \in \partial \mathcal{K}_i} \int_\mathcal{E} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \, dA \right) \tag{104}$$

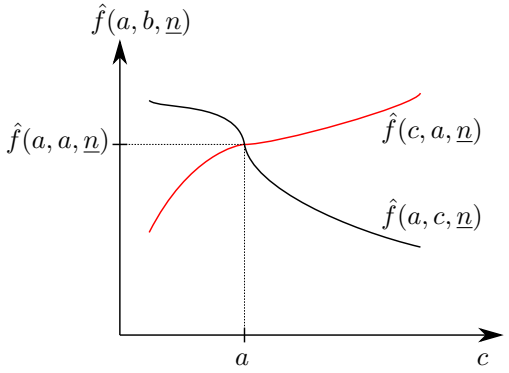
$$\overset{(101)}{=} \frac{\partial}{\partial t} \int_\Omega c_h \, dV + \sum_{\mathcal{E} \in \mathfrak{E}_h^b} \int_\mathcal{E} \hat{f}(c_{h,i}^-, c_{h,i}^+, \underline{n}_i) \, dA \tag{105}$$

$$= \frac{\partial}{\partial t} \int_\Omega c_h \, dV + \int_{\partial\Omega} \hat{f}(c_h^-, c_B, \underline{n}_{\partial\Omega}) \, dA, \tag{106}$$

which is the discrete *global* equivalent of (97)

**Monotonicity**

- Interpretation: *Monotonicity* in combination with *consistency* can be visualized as follows




- Monotonicity is required to prove *stability* of a scheme. Note that there are several notions of stability in numerical methods. In this section, we define stability in the continuous setting via the *energy estimate*

$$\|c(x,t)\|_\Omega^2 \leq \|c(x,0)\|_\Omega^2 \quad \forall t \geq 0, \tag{107}$$

where we assume zero Dirichlet boundary conditions wherever applicable. That is, we call a system stable if the *energy* $\|c(x,t)\|_\Omega^2$ of the system can only decrease in the absence of inflow

- The discrete equivalent of (107) is the energy estimate

$$\|c_h(x,t)\|_\Omega^2 \leq \|\pi_p(c(x,0))\|_\Omega^2 \quad \forall t \geq 0, \tag{108}$$

where we once again assume zero Dirichlet boundary conditions wherever applicable

- Important result: It can be shown that (83) satisfies (108) if $\hat{f}$ is Lipschitz-continuous and fulfills the monotonicity property (94). For the proof, e.g. see (Di Pietro and Ern, 2012, p. 103)

- Remark: The quantity

$$\Theta = \frac{1}{2} \|\pi_p(c(x,0))\|_\Omega^2 - \frac{1}{2} \|c_h(x,t)\|_\Omega^2 \tag{109}$$

is a measure for the numerical dissipation of a scheme. It must be positive for all times, but should be as low as possible too.

## 3.3 Exemplary classes of numerical fluxes

**Notation in the remainder of this chapter**

- We drop the cell-index $i$ since we consider a single edge, as a result

$$c_h^- = c_{h,i}^- \tag{110}$$

$$c_h^+ = c_{h,i}^+ \tag{111}$$

$$\underline{n} = \underline{n}_i \tag{112}$$

- We use the following abbreviations:

$$\{c_h\} \coloneqq \frac{c_h^- + c_h^+}{2} \tag{113}$$

$$[\![c_h]\!] \coloneqq c_h^- - c_h^+ \tag{114}$$

$$f_{\underline{n}}(c_h) = \underline{f}(c_h) \cdot \underline{n} \tag{115}$$

$$\left\{\underline{f}(c_h)\right\} \coloneqq \frac{1}{2}\left(\underline{f}(c_h^-) + \underline{f}(c_h^+)\right) \cdot \underline{n} = \frac{1}{2}\left(f_{\underline{n}}(c_h^-) + f_{\underline{n}}(c_h^+)\right) \tag{116}$$

  Note that $\left\{\underline{f}(c_h)\right\}$ is a scalar quantity

**Central flux**

- Definition:

$$\hat{f}_{\mathrm{CF}}(c_h^-, c_h^+, \underline{n}) \coloneqq \underline{f}\left(\{c_h\}\right) \cdot \underline{n} = f_{\underline{n}}\left(\{c_h\}\right) \tag{117}$$

- For linear advection, one can show that the energy norm is conserved exactly, i.e.

$$\|c_h(x,t)\|_\Omega = \|\pi_p(c(x,0))\|_\Omega \quad \forall t \geq 0, \tag{118}$$

  which renders the flux optimal in some sense ($\Theta = 0$) even though it is not monotone in general:

$$\frac{\partial}{\partial c_h^-}\hat{f}_{\mathrm{CF}}(c_h^-, c_h^+, \underline{n}) = \frac{\partial}{\partial c_h^-} f_{\underline{n}}\left(\{c_h\}\right) \tag{119}$$

$$= \frac{\partial}{\partial c_h^-} f_{\underline{n}}\left(\frac{c_h^- + c_h^+}{2}\right) \tag{120}$$

$$= f'_{\underline{n}}\left(\frac{c_h^- + c_h^+}{2}\right)\frac{\partial}{\partial c_h^-}\left(\frac{c_h^- + c_h^+}{2}\right) \tag{121}$$

$$= f'_{\underline{n}}\left(\{c_h\}\right)\frac{1}{2} \tag{122}$$

$$\Rightarrow \operatorname{sgn}\left(\frac{\partial}{\partial c_h^-}\hat{f}_{\mathrm{CF}}(c_h^-, c_h^+, \underline{n})\right) = \operatorname{sgn}\left(f'_{\underline{n}}\left(\{c_h\}\right)\right), \tag{123}$$

  where $\operatorname{sgn}(\cdot)$ is the sign function

- However, this flux is typically unstable for first order conservation laws (cf. exercise 6). 'Conclusion': Some numerical dissipation is necessary to ensure stability
- However, it can still be useful in situations where fluxes need to be defined for multiple terms. The most notable example is the incompressible Navier-Stokes equation, where the pressure is most often treated using a central flux

**Upwind flux**

- Definition:

$$\hat{f}(c_h^-, c_h^+, \underline{n}) = \begin{cases} \underline{f}(c_h^-) \cdot \underline{n} & \text{if } U > 0 \\ \underline{f}(c_h^+) \cdot \underline{n} & \text{if } U < 0 \end{cases} \tag{124}$$

  for some flow velocity $U$ normal to the edge.

- In case of linear advection ($\underline{f} = \underline{u}(\underline{x})\, c$) or convection, a sensible choice is given by $U = \underline{u} \cdot \underline{n}$
- Extremely stable due to very high numerical dissipation (that is, $\Theta$ is extremely large)

**Lax-Friedrichs flux**

- Definition:

$$\hat{f}_{\mathrm{LF}}(c_h^-, c_h^+, \underline{n}) = \left\{\underline{f}(c_h)\right\} + \frac{C}{2}[\![c_h]\!] \tag{125}$$

  where $C \in \mathbb{R}^+$ is chosen sufficiently large to ensure stability

- Classical interpretation: Artificial viscosity $C$. In context of DG, $C$ is often called *penalty factor* because the flux is amplified if the jump of the solution is large (cf. equation (88))
- Classical Lax-Friedrichs:
$$C_{\mathrm{LF}} \coloneqq \sup_c \left| f'_{\underline{n}}(c) \right| \tag{126}$$
- Ratio:
$$\frac{\partial}{\partial c_h^-} \hat{f}_{\mathrm{LF}}(c_h^-, c_h^+, \underline{n}) = \frac{\partial}{\partial c_h^-} \left\{ \underline{f}(c_h) \right\} + \frac{\partial}{\partial c_h^-} \frac{C_{\mathrm{LF}}}{2} [\![ c_h ]\!] \tag{127}$$
$$= \frac{1}{2} f'_{\underline{n}}(c_h^-) + \frac{C_{\mathrm{LF}}}{2} \tag{128}$$
$$= \frac{1}{2} f'_{\underline{n}}(c_h^-) + \frac{1}{2} \sup_c \left| f'_{\underline{n}}(c) \right| \tag{129}$$
$$\geq 0 \tag{130}$$
- Local Lax-Friedrichs flux: $C$ is chosen locally (in each edge or each quadrature node), i.e.
$$C_{\mathrm{LLF}} \coloneqq \sup_{c \in [c_h^-, c_h^+]} \left| f'_{\underline{n}}(c) \right| \tag{131}$$
- Example: Linear advection of a concentration field $c$ in a velocity field $\underline{u}$. Then $\underline{f} = \underline{u}c$ and $\underline{f}' = \underline{u}$, and thus
$$\sup_c \left| f'_{\underline{n}}(c) \right| = \sup_c |\underline{u} \cdot \underline{n}| = |\underline{u} \cdot \underline{n}| =: C_{\mathrm{LLF}} \tag{132}$$
gives a stable formulation. Inserting this into (125) gives
$$\hat{f}_{\mathrm{LF}}(c_h^-, c_h^+, \underline{n}) = \left\{ \underline{f}(c_h) \right\} + \frac{C_{\mathrm{LLF}}}{2} [\![ c_h ]\!] \tag{133}$$
$$= (\underline{u} \cdot \underline{n}) \{ c_h \} + \frac{1}{2} |\underline{u} \cdot \underline{n}| [\![ c_h ]\!] \tag{134}$$
$$= \frac{1}{2} (\underline{u} \cdot \underline{n}) \begin{cases} (c_h^- + c_h^+) + (c_h^- - c_h^+) & \text{if } \underline{u} \cdot \underline{n} > 0 \\ (c_h^- + c_h^+) - (c_h^- - c_h^+) & \text{if } \underline{u} \cdot \underline{n} < 0 \end{cases} \tag{135}$$
$$= (\underline{u} \cdot \underline{n}) \begin{cases} c_h^- & \text{if } \underline{u} \cdot \underline{n} > 0 \\ c_h^+ & \text{if } \underline{u} \cdot \underline{n} < 0 \end{cases}, \tag{136}$$
which is precisely the definition of the upwind flux for this case.
- The original Lax-Friedrichs flux is extremely dissipative (that is, $\Theta$ is very large), but very easy to compute. The local Lax-Friedrichs flux is much less dissipative in general, but still not optimal. More severely, both variants do not respect the physics behind the PDE, because the direction of the exchange of information is not taken into account. We will discuss this in more detail in Section 3.4 when introducing the concept of *characteristics*

**Godunov flux**

- Definition:
$$\hat{f}(c_h^-, c_h^+, \underline{n}) = \begin{cases} \min\limits_{s \in [c_h^-, c_h^+]} \underline{f}(s) \cdot \underline{n} & \text{if } c_h^- \leq c_h^+ \\ \max\limits_{s \in [c_h^+, c_h^-]} \underline{f}(s) \cdot \underline{n} & \text{otherwise} \end{cases} \tag{137}$$
- In a certain sense, the *best* approximation that *always* ensures stability (monotonicity directly follows from the definition via minimum/maximum)
- Requires the solution of a nonlinear optimization problem at each node on each edge of the grid, which is typically much too costly. However, we will see in Section 3.4 that it helps us getting insights into the construction of fluxes that are suitable for individual PDEs

**Discussion**

- All fluxes we discussed so far have their advantages and disadvantages
- The *central flux* has the best dissipation properties, but is typically unstable
- The *upwind flux* is extremely stable, but extremely dissipative
- The *Lax-Friedrichs* flux is less dissipative than the upwind flux and stable. However, it does not respect the physical propagation direction of information
- The *Godunov flux* is stable, hardly dissipative and physical. However, it is prohibitively costly to compute and thus rarely used.

## 3.4 Riemann solvers

- This section is largely based on (Toro, 2009, Chapter 2)

**The scalar Riemann problem**

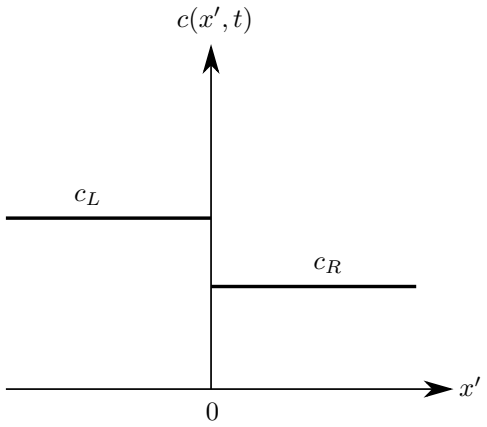
- A one-dimensional initial value problem of the type

$$\frac{\partial c}{\partial t} + \frac{\partial f(c)}{\partial x'} = 0 \tag{138a}$$

with piecewise constant initial data

$$c(x', 0) = \begin{cases} c_L & x' < 0 \\ c_R & x' > 0 \end{cases} \tag{138b}$$

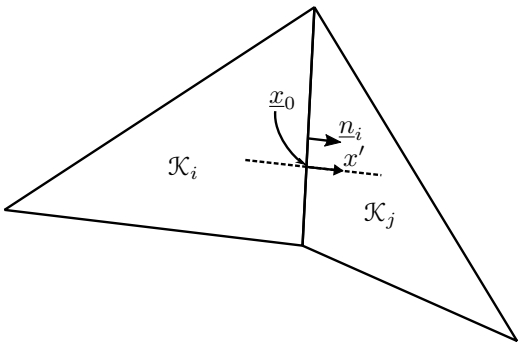
with constants $c_L$ and $c_R$ is called a *Riemann problem*




- A *Riemann solver* $\mathrm{RS}(c_L, c_R)$ computes the exact solution of the Riemann problem at $x' = 0$
- An *approximate Riemann solver* computes a suitable approximation $\mathrm{RS}_h(c_L, c_R) \approx \mathrm{RS}(c_L, c_R)$
- Idea: Define edge-normal coordinate $x'$ such that

$$\underline{x} = \underline{x}_0 + x' \, \underline{n} \tag{139}$$

for each point $\underline{x}_0$ on some edge of cell $\mathcal{K}_i$:




Then solve the Riemann problem with $c_L = c_h^-(\underline{x_0}, t)$ and $c_R = c_h^+(\underline{x_0}, t)$

- Important result: The Godunov flux is equivalent to

$$\hat{f}(c_h^-, c_h^+, \underline{n}) = \underline{f}(\mathrm{RS}\,(c_L, c_R)) \cdot \underline{n}, \tag{140}$$

which motivates the setting

$$\hat{f}(c_h^-, c_h^+, \underline{n}) = \underline{f}(\mathrm{RS}_h\,(c_L, c_R)) \cdot \underline{n} \tag{141}$$

to obtain numerical flux functions that are physically correct, reasonably accurate and cheap to evaluate

**Characteristics**

- Assuming $f$ in (138) is sufficiently smooth, we can write the Riemann problem in quasi-linear form:

$$\frac{\partial c}{\partial t} + a\frac{\partial c}{\partial x'} = 0 \tag{142a}$$

with

$$c(x', 0) = \begin{cases} c_L & x' < 0 \\ c_R & x' > 0 \end{cases} \tag{142b}$$

and

$$a = a(c) = \frac{\partial f(c)}{\partial c} \tag{142c}$$

- If $a$ is constant, (142a) is a scalar *linear PDE* . Otherwise, (142a) is non-linear
- We call a curve $x'_c(t)$ in the $x'$-$t$-plane along which $c$ is constant, i.e. where

$$\frac{dc(x'_c(t), t)}{dt} = \frac{\partial c}{\partial t} + \frac{dx'_c(t)}{dt}\frac{\partial c}{\partial x'} = 0 \tag{143}$$

for some initial condition $x'_c(0) = x'_0$, a *characteristic (curve)*. Scalar conservation laws admit exactly one curve of this type for each starting point $x'_0$

- Comparing (143) to (142a), we see that this condition is fulfilled if $x'_c(t)$ satisfies the ODE

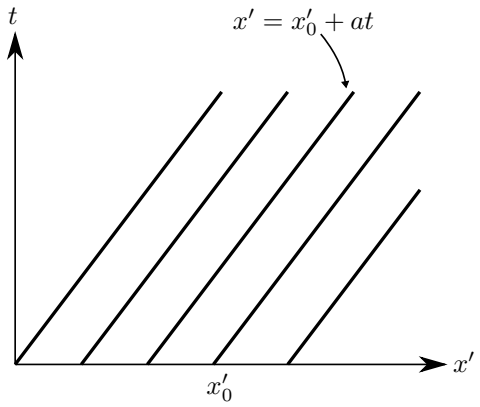
$$\frac{dx'_c(t)}{dt} = a(c). \tag{144}$$

**The linear case**

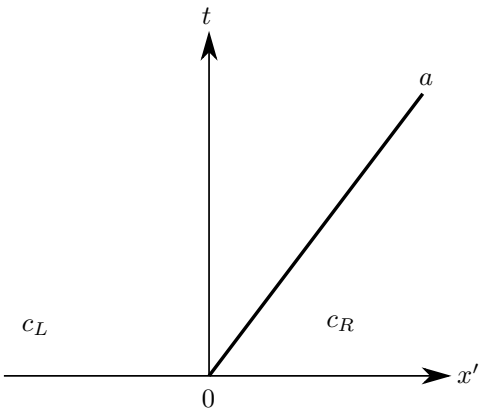
- If $a$ is constant, (144) can easily be solved using the initial condition $x'_c(0) = x'_0$:

$$x'_c(t) = x'_0 + at \tag{145}$$

- That is, a characteristic emanating from $x'_0$ is a *straight line* in the $x'$-$t$-plane along which information propagates with the *characteristic speed* $a$. Consider a family of characteristics emanating from different points on $x'$. As $a$ is constant, all these characteristics are parallel and cannot interact

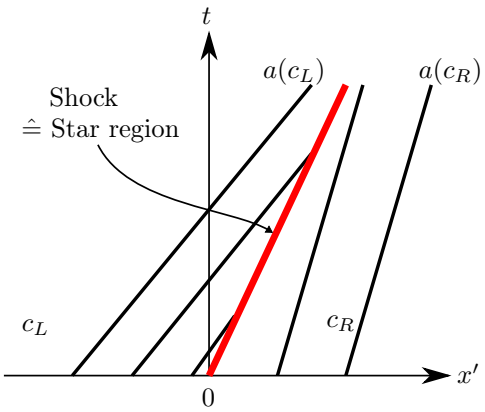



- The solution of the Riemann problem for this case consists of two constant regions that are separated by a single discontinuity propagating with the characteristic speed $a$:

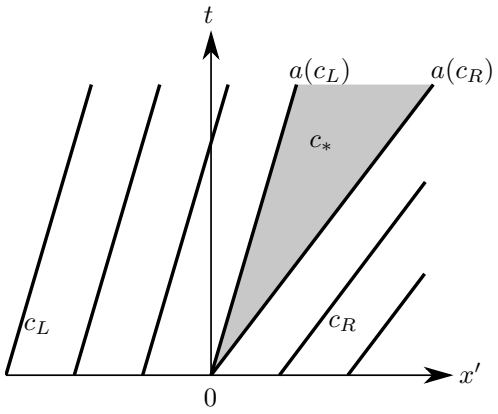



**The non-linear case**

- In the context of the Riemann problem, two characteristics with speeds $a(c_L)$ and $a(c_R)$ emanate from $x' = 0$. These characteristics are *not* parallel in general, which is why they will form a single *wave* across which the solution changes. The region outside of this wave remains undisturbed, while the solution $c_*$ in the *star region* between the characteristics needs to be computed by considering their interaction. If characteristics *intersect*, a *shock wave* is formed:




Here, the star region is denoted by the thick red line. If characteristics *expand*, a *rarefaction wave* is formed:




Here, the star region (light gray) has finite size

- In the general case, we have to distinguish three cases: If $a(c_L) > a(c_R)$, both characteristics collapse into a *shock wave*. If $a(c_L) < a(c_R)$, both characteristics form a *rarefaction wave*. If $a(c_L) = a(c_R)$, there is a *contact wave*
- In case of a *shock wave*, the solution is given by the two constant states $c_L$ and $c_R$ with $a(c_L) > a(c_R)$ that are separated by a shock moving with the *shock speed* $S$ that can be determined from the *Rankine-Hugoniot* condition

$$S = \frac{f(c_R) - f(c_L)}{c_R - c_L} \tag{146}$$

- In case of a *rarefaction wave*, the solution is given by the two constant states $c_L$ and $c_R$ with $a(c_L) < a(c_R)$ that are separated by a region with a smooth transition of the solution from $c_L$ to $c_R$. One can show that the solution inside the star region is self-similar w.r.t. to the variable $x'/t$, which allows sampling the solution at $x' = 0'$
- In case of a *contact wave*, the characteristics are parallel, which is exactly what we have already seen in the linear case. The waves thus do not interact and the so-called *contact discontinuity* is trivially propagated in the $x'$-$t$-plane

**Systems of conservation laws**

- In the discussions above, we have limited ourselves to scalar conservation laws. The general multi-dimensional case is given by:

$$\frac{\partial \underline{C}}{\partial t} + \frac{\partial \underline{F}(\underline{C})}{\partial x'} = 0 \tag{147a}$$

with piecewise constant initial data

$$\underline{C}(x', 0) = \begin{cases} \underline{C}_L & x' < 0 \\ \underline{C}_R & x' > 0 \end{cases}, \tag{147b}$$

where $\underline{C} = (c_1, \ldots, c_M)^T$ is the vector of dependent variables, $\underline{F} = (f_1, \ldots, f_M)^T$ is the vector of fluxes and $\underline{C}_L \in \mathbb{R}^M$ and $\underline{C}_R \in \mathbb{R}^M$ are constant vectors

- We assume that $\underline{F}$ is smooth so that we can use the chain rule to write (147) in *quasi-linear* form:

$$\frac{\partial \underline{C}}{\partial t} + \underline{\underline{A}}(\underline{C}) \frac{\partial \underline{C}}{\partial x'} = 0 \tag{148a}$$

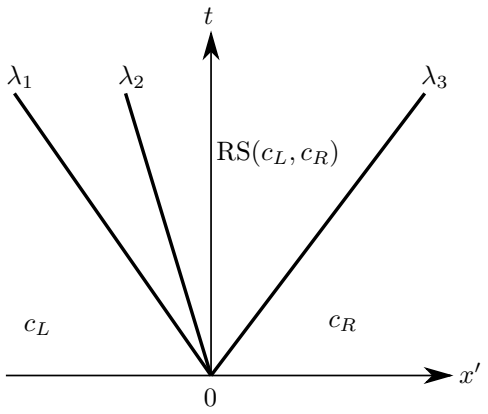
$$\underline{C}(x', 0) = \begin{cases} \underline{C}_L & x' < 0 \\ \underline{C}_R & x' > 0 \end{cases}, \tag{148b}$$

with the *flux Jacobian*

$$\underline{\underline{A}} = \underline{\underline{A}}(\underline{C}) = \frac{\partial \underline{F}(\underline{U})}{\partial \underline{C}}. \tag{148c}$$

Equation (148a) is *non-linear* is $\underline{C}$, but *linear* in $\nabla \underline{C}$, hence the term *quasi-linear* even though (147a) is a system of non-linear PDE

- If $\underline{\underline{A}}$ is constant, (148a) is linear. Otherwise, (148a) is non-linear
- In the following, we will assume that the eigenvalues $\{\lambda_m\}_{m=0,\dots,M-1}$ of $\underline{\underline{A}}$ are *real* and the corresponding eigenvectors are linearly independent, i.e. that (148a) is a system of *hyperbolic* conservation laws. Note that the eigenvalues depend on $\underline{C}$ if (148a) is non-linear
- The eigenvalues $\lambda_m$ represent the *characteristic speeds* of the $M$ characteristics. If $\underline{\underline{A}}$ is linear, all $\lambda_m$ are constant and the solution to the Riemann problem has the following pattern:




  In the non-linear case, each characteristic speed $\lambda_m$ defines a single wave between the characteristics associated $\lambda_m(\underline{C}_L)$ and $\lambda_m(\underline{C}_R)$, which we can analyze using the methods discussed in the previous section (i.e., is it a shock, rarefaction or contact wave?)
- Obtaining $\underline{C}_*$ for this general case is non-trivial

# 4 Temporal discretization

- The scalar conservation law (42) has to be discretized in time *and* space in order to obtain a fully discrete system. In section 2, the *spatial* discretization has been carried out while the temporal setting is still continuous. The resulting system of ODE (83) needs to be integrated in time, which leads to the so-called *method of lines*: *Discretize space first, then time*
- A first alternative is sometimes (especially in the German-speaking world) called *Rothe's method*: *Discretize time first, then space*. This approach is equivalent to the method of lines many cases, but can be advantageous for moving problem domains
- A second alternative is the *space-time* approach: *Discretize time and space equivalenty*. In essence, the temporal dimension is treated as yet another spatial dimension, i.e. the DG basis is extended to a space-time basis and numerical fluxes are introduced at space-time boundaries. The resulting method is very attractive from a theoretical point of view because it provides maximum flexibility in terms of adaptivity and mesh deformation. However, corresponding schemes are prohibitively expensive to solve in most application scenarios.
- The method of lines is predominantly used in the community, which is why we will limit ourselves to this case within this chapter.
- Once again, we will drop the cell-index $i$ within this section
- Parts of this section are based on LeVeque (2002)

## 4.1 Setting for investigating of time-stepping methods

- We recast (85) in the standard form for ODE solvers:

$$\frac{\partial \underline{\tilde{c}}}{\partial t} = -\underline{\underline{M}}^{-1}\underline{f}_h(t, c_h(t)) \tag{149}$$

- Many results regarding ODE solvers are based on systems of linear systems of ODE without source terms, i.e.
$$\frac{\partial \underline{\tilde{c}}}{\partial t} = \underline{\underline{A}}\,\underline{\tilde{c}} \tag{150}$$
for a constant matrix $\underline{\underline{A}}$. In case of our DG discretization, we can write
$$-\underline{\underline{M}}^{-1}\underline{f}_h(t, c_h(t)) = \underline{\underline{A}}\,\underline{\tilde{c}} + \underline{b} \tag{151}$$
if (42) is either linear or if $\underline{f}_h(t, c_h(t))$ is linearized around $c_h(t_0)$. Here, the vector $\underline{b}$ contains contributions from boundary conditions. In the following, we will assume periodic boundary conditions, i.e. $\underline{b} = \underline{0}$

- Commonly, time discretization is introduced by discussing Finite Difference approximations of the temporal derivative in some interval $[t_n, t_{n+1}]$ of length $\Delta t = t_{n+1} - t_n$. For example, consider the well-known forward difference
$$\left.\frac{\partial \underline{\tilde{c}}}{\partial t}\right|_{t=t_n} \approx \frac{\underline{\tilde{c}}(t_n + \Delta t) - \underline{\tilde{c}}(t_n)}{\Delta t} \tag{152}$$
leading to
$$\frac{\underline{\tilde{c}}(t_{n+1}) - \underline{\tilde{c}}(t_n)}{\Delta t} = -\underline{\underline{M}}^{-1}\underline{f}_h(t_n, c_h(t_n)) \tag{153}$$

- Another approach is to integrate (149) in time to obtain
$$\underline{\tilde{c}}(t_{n+1}) - \underline{\tilde{c}}(t_n) = -\underline{\underline{M}}^{-1}\int_{t_n}^{t_{n+1}} \underline{f}_h(t, c_h(t))\, dt \tag{154}$$
Different time integrators can thus also be distinguished by the evaluation of the time integral on the right-hand side

## 4.2 Explicit methods

- Due to the local nature of the approach, DG methods lend themselves to explicit approaches where only information from old time steps is required to advance the solution. More precisely, explicit methods only require operator evaluations at time levels that are either already known or can be interpolated

- The simplest example is the *Explicit Euler* method, which follows from the forward difference (152) and is thus given by
$$\underline{\tilde{c}}(t_{n+1}) = \underline{\tilde{c}}(t_n) - \Delta t\,\underline{\underline{M}}^{-1}\underline{f}_h(t_n, c_h(t_n)) \tag{155}$$
(cf. equation (153)). It can be shown to be first order accurate in time.

- In terms of (154), the Explicit Euler method can be interpreted as the *rectangle rule* for numerical integration using the *left* function value to determine the height of the rectangle over the interval of length $\Delta t$

- Many explicit time-stepping schemes with more favorable properties than the Explicit Euler method exist. *Adams-Bashforth* methods, for example, are linear *multi-step* methods where the last $S$ time-steps are linearly combined to create a scheme that converges with $S$-th order. Example: The second order two-step Adams-Bashforth scheme is given by
$$\underline{\tilde{c}}(t_{n+1}) = \underline{\tilde{c}}(t_n) + \frac{3}{2}\Delta t\,\underline{f}_h(t_n, c_h(t_n)) - \frac{1}{2}\Delta t\,\underline{f}_h(t_{n-1}, c_h(t_{n-1})) \tag{156}$$

- The most prominent class of methods is, however, the class of *Runge-Kutta* methods which can be written as
$$\underline{\tilde{c}}(t_{n+1}) = \underline{\tilde{c}}(t_n) - \Delta t\sum_{s=0}^{S-1}(\underline{\alpha})_s\underline{k}_s \tag{157}$$
where
$$\underline{k}_s = \underline{f}_h\left(t_n + (\underline{\beta})_s\Delta t,\ \underline{\tilde{c}}(t_n) + \Delta t\sum_{\tilde{s}=0}^{S-1}(\underline{\underline{\Gamma}})_{s,\tilde{s}}\underline{k}_{\tilde{s}}\right) \tag{158}$$
Here, $\underline{\alpha} \in \mathbb{R}^S$, $\underline{\beta} \in \mathbb{R}^S$ and $\underline{\underline{\Gamma}} \in \mathbb{R}^{S,S}$ are the coefficients for some specific Runge-Kutta variant. These coefficients are often given in the form of a *Butcher tableau*:

$$\begin{array}{c|c} \underline{\beta} & \underline{\underline{\Gamma}} \\ \hline & \underline{\alpha}^T \end{array}$$

- In terms of the integral form (154) of the ODE, the entries can be interpreted as follows: A Runge-Kutta method is a combination of a numerical integration rule for the integral on the right-hand side, supplemented by an interpolation rule that allows for an approximation of the integrand at the required nodes. The entries in $\underline{\alpha}$ and $\underline{\beta}$ denote the quadrature *weights* and quadrature *nodes* of the numerical integration rule, while the entries in $\underline{\underline{\Gamma}}$ define the interpolation scheme
- In an *explict Runge-Kutta* method, the vector $\underline{k}_s$ may only depend on coefficients $\underline{k}_{\tilde{s}}$ where $\tilde{s} < s$. As a result, $\underline{\underline{\Gamma}}$ must be a *strictly* lower triangular matrix for explicit Runge-Kutta schemes! Some well-known examples are:

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

(a) Explicit Euler

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) Heun's method

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

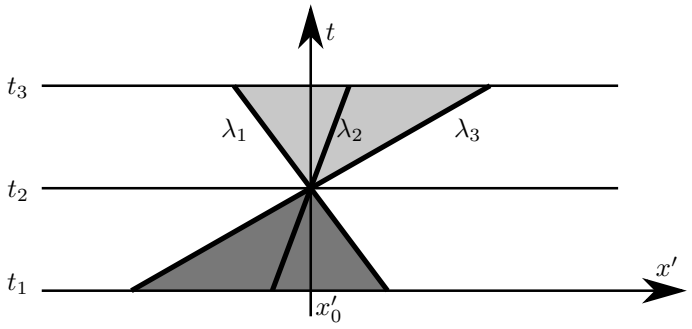
(c) Classical Runge-Kutta

- An $S$ stage Runge-Kutta scheme has a maximum order of $S$. On the other hand, additional stages can be added to an $S$-th order Runge-Kutta scheme to (i) decrease the truncation error for non-linear problems (ii) enlarge/shape the stability region (see below) (iii) *decrease* memory consumption (by decreasing the number of stages that need to be *stored*). It can be shown that no explicit $S$-stage Runge-Kutta scheme of order $S > 4$ exists. That is, a fifth order scheme requires at least six stages, which is why Runge-Kutta schemes of fifth order or higher are rarely used.

## 4.3 Stability of explict time integration

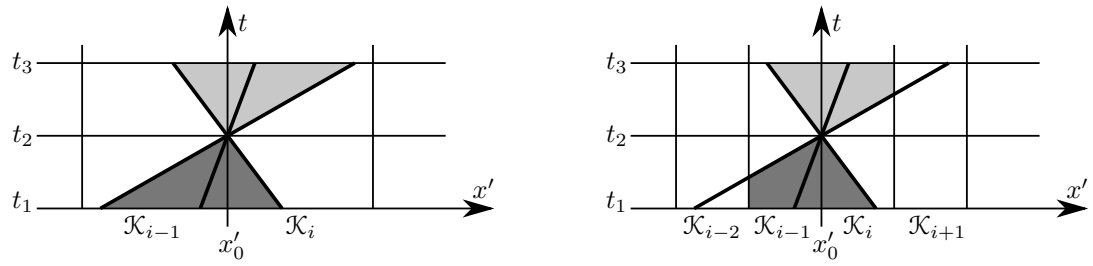
### The CFL condition

- The theory behind the original CFL criterion is only valid for the Explicit Euler method on context of Finite Difference discretizations of linear systems of PDE. Still, the idea behind the CFL condition has an instructive geometric interpretation. Additionally, we will see below how this criterion can be generalized
- In section 3.4, we have discussed the characteristics emanating from a single point $x'_0$. We define the *domain of influence* of this point as the region in the $x'$-$t$-plane that is influenced by the flow conditions at $x'_0$. The *domain of dependence* of a PDE in this point, on the other hand, is obtained by tracing the characteristics *back* in time in order to find all points $x'$ that have influenced the present flow state in $x'_0$.
- Using the example of a system of three linear first order PDE with characteristic speeds $\lambda_1$, $\lambda_2$ and $\lambda_3$ and three time levels $t_1 < t_2 < t_3$, this can be sketched as follows:




The light gray region is influenced by the flow state in $x'_0$, while the flow state in $x'_0$ depends on the flow states in dark gray region only. Note that the characteristics may be curved when considering non-linear problems

- The *numerical domain of influence* and the *numerical domain of dependence* are the corresponding discrete counterparts that depend on the discretization scheme.
- *CFL condition*: "*A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as* $\Delta t$ *and* $\Delta x$ *go to zero.*" (LeVeque, 2002, p. 69). In our case, $\Delta x = h$. Note that the CFL condition is a *necessary condition* for stability, not a sufficient one
- Ratio: If the numerical domain of dependence is *smaller* than the true domain of dependence, then there exist initial conditions that change the solution of the PDE, but that do *not* change the numerical solution. "Clearly the method cannot converge to the proper solution for all choices of initial data under these circumstances" (LeVeque, 2002, p. 70)
- In the context of above example, consider two grids with mesh sizes $h$ (left) and $h/2$ (right), respectively:




  Here, the *numerical* domains of influence (light gray) and dependence (dark gray) are sketched for a one-dimensional discretization. In the picture on the right ($h/2$), we clearly see that the CFL condition is violated because changes in the initial condition in $\mathcal{K}_{i-2}$ do not affect the solution at $x'_0$. The scheme will thus not converge to the true solution when letting $\Delta t$ and $h$ tend to zero *simultaneously*. In fact, we will then typically violate the energy estimate (108) (which was derived for exact time integration) and the solution diverges. To recover a stable scheme, we have to adapt the time-step size when decreasing the mesh size
- Mathematically, we can write the CFL condition as

$$\Delta t \, \lambda_m \leq h \qquad \forall \lambda_m \tag{159}$$

  which leads to the definition of the *Courant number*

$$\nu = \frac{\Delta t}{h} \bar{\lambda} \tag{160}$$

  with

$$\bar{\lambda} = \max_m |\lambda_m|. \tag{161}$$

  To fulfill the CFL condition, we have to ensure $\nu \leq 1$.
- Physically, this can be interpreted as follows: We have to ensure that information propagating with the characteristic speeds $\lambda_m$ does not *skip* a cell of size $h$, which we achieve by limiting the time of propagation such that $\lambda_m \Delta t < h$
- In practice, we choose a constant $0 < \nu_h \leq 1$ and compute the time-step according to

$$\Delta t \leq \nu_h \frac{h}{\bar{\lambda}} \tag{162}$$

  to account for non-linear effects and the influence of the approximation order $P$.

**The notion of absolute stability**

- As we have seen, the (original) CFL condition gives an intuitive bound for the time-step for the most simple configurations. The concept of *absolute stability* leads to a generalization of this result. It is defined w.r.t. the model equation

$$\frac{\partial c}{\partial t} = \lambda c \tag{163}$$

  for a constant $\lambda \in \mathbb{C}$ and

$$c(0) = 1. \tag{164}$$

- The solution of (163) is given by the wave equation

$$c(t) = \exp(\lambda t). \tag{165}$$

For $\Re(\lambda) < 0$, the solution is given by damped waves, while $\Re(\lambda) > 0$ leads to amplified waves. Purely imaginary values of $\lambda$ correspond to undamped oscillations

- Any time integrator for (163) can be written as

$$c(t_{n+1}) = \varphi(\Delta t\, \lambda) c(t_n) \tag{166}$$

and, due to linearity, further as

$$c(t_{T-1}) = (\varphi(\Delta t\, \lambda))^{T-1} c(t_0) \tag{167}$$

for time levels $\{t_n\}_{n=0,\dots,T-1}$

- The function $\varphi$ is called the *stability function* of the ODE solver. The region $\mathcal{Z} = \{z \in \mathbb{C} : |\varphi(z)| < 1\}$ is called the *region of absolute stability* (or *stability region*, for short) of the ODE solver. Ratio: For $T \to \infty$, $c(t_{T-1})$ can only stay bounded if $|\varphi(\Delta t\, \lambda)| < 1$. In particular, we have

$$\lim_{T\to\infty} c(t_{T-1}) = 0 \tag{168}$$

in this case

- *A-stability*: An ODE solver is called *A-stable* if the stability region $\{z \in \mathbb{C} : \Re(z) < 0\} \subset \mathcal{Z}$. That is, an ODE solver is A-stable if

$$\lim_{T\to\infty} c(t_{T-1}) = 0 \qquad \forall \lambda : \Re(\lambda) < 0 \tag{169}$$

- Unfortunately, no relevant A-stable explicit ODE solvers exist. Setting $z = \Delta t\, \lambda$ with $\Re(\lambda) < 0$, this implies that no explicit ODE solver is stable for large values of $\Delta t$

- Example: Applying the Explicit Euler scheme to (163) yields

$$\frac{c(t_{n+1}) - c(t_n)}{\Delta t} = \lambda c(t_n) \tag{170}$$

$$\Rightarrow c(t_{n+1}) = c(t_n) + \Delta t\, \lambda\, c(t_n) \tag{171}$$

and thus $\varphi(z) = z + 1$ where $z = \Delta t\, \lambda$. It follows that the corresponding region of absolute stability

$$\mathcal{Z} = \{z \in \mathbb{C} : |1 + z| < 1\} \tag{172}$$

is the unit circle around $-1$ in the complex plane. Setting $z = \Delta t\, \lambda$ with $\Re(\lambda) < 0$, we observe that $|1 + z| = |1 + \Delta t\, \lambda| > 1$ for sufficiently large $\Delta t$, which contradicts A-stability

- Remark: The stability region of Runge-Kutta schemes is *enlarged* as the number of stages $S$ is increased (cf. exercise 6), which is a major reason for the popularity of these schemes. While Adams-Bashforth schemes require fewer operator evaluations for a given stable value of $\Delta t$ (only *one* evaluation, irrespective of $S$!), they suffer from the fact their stability region *shrinks* when in $S$ is increased

**In the context of DG**

- Now consider the case of a linear DG discretization of the linear transport equation with periodic boundary conditions that we can write as

$$\underline{\underline{A}}\, \underline{\tilde{c}} = -\underline{\underline{M}}^{-1} \underline{f}_h(t, c_h(t)) \tag{173}$$

(cf. (151)). We thus have to integrate

$$\frac{\partial \underline{\tilde{c}}}{\partial t} = \underline{\underline{A}}\, \underline{\tilde{c}} \tag{150, repeated}$$

in time. Assuming $\underline{\underline{A}}$ is invertible, we can diagonalize this system by introducing the decomposition $\underline{\underline{A}} = \underline{\underline{K}}\, \underline{\underline{\Lambda}}\, \underline{\underline{K}}^{-1}$ where

$$\underline{\underline{\Lambda}} = \begin{pmatrix} \lambda_{h,1} & 0 & \dots & 0 \\ 0 & \lambda_{h,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{h,N_p \cdot J} \end{pmatrix} \tag{174}$$

is a diagonal matrix containing the eigenvalues of $\underline{\underline{A}}$. We call the set $\{\lambda_{h,l}\}_{l=0,\dots,N_p \cdot J}$ the discrete *spectrum* of our DG discretization. Note these are *not* the characteristic speeds of the system of PDE (even though they are related)

- Introducing the new variables $\underline{w} = \underline{\underline{K}}^{-1}$, (150) simplifies to a *decoupled* system of linear equations

$$\frac{\partial \underline{w}}{\partial t} = \underline{\underline{\Lambda}}\,\underline{w} \tag{175}$$

in the form of our model equation (163). As a result, our time integration is stable if $z = \Delta t\,\lambda_{h,l}$ is within the stability region of the ODE solver for all $\lambda_{h,l}$ of our discretization. In other words, we have to ensure to the spectrum of our discretization lies within the stability region. Using the example of the Explicit Euler method, we obtain the condition

$$|1 + \Delta t\,\lambda_{h,l}| < 1 \qquad \forall \lambda_{h,l}. \tag{176}$$

The step-size $\Delta t$ thus acts as a scaling of the spectrum

- Observation: The most critical eigenvalue within the spectrum is typically given by the eigenvalue with the most negative real part. For $P = 0$, a lower bound for real part of the spectrum can be estimated by

$$\Re(\lambda_{h,l}) \geq -1 - \frac{\bar{\lambda}}{h}. \tag{177}$$

We can combine this estimate with (176) in order to obtain the constraint

$$\left| 1 - 1 - \Delta t\,\frac{\bar{\lambda}}{h} \right| < 1 \tag{178}$$

which can be simplified to

$$\bar{\lambda}\,\frac{\Delta t}{h} < 1 \tag{179}$$

since $\bar{\lambda}$ is non-negative. This exactly matches definition of the Courant number $\nu$!

- Remaining question: How does $\Re(\lambda_{h,l})$ scale for different numerical fluxes and different values of $P$? In practice, the most common choice is

$$\Delta t \leq \nu_h \frac{h}{\bar{\lambda}} \tag{180a}$$

with

$$\nu_h \leq \frac{1}{2P+1} \tag{180b}$$

Cockburn and Shu (2001), even though this is only a rough estimate (cf. exercise 6).

- Remark: For second order PDE including physical diffusion, we obtain an additional diffusive CFL restriction

$$\Delta t \leq \nu_h \frac{h^2}{\bar{\lambda}_d} \tag{181a}$$

with

$$\nu_{h,d} \leq \frac{1}{(2P+1)^2}, \tag{181b}$$

where $\bar{\lambda}_d$ is a measure for the speed of diffusive processes Gassner et al. (2010). This restriction can get prohibitive for practical calculations, even for moderate values of $P$

## 4.4 Implicit methods

- If the CFL restriction is prohibitive (see above) or if strongly non-linear source terms and/or algebraic constraints (such as the incompressibility constraint in the incompressible Navier-Stokes equations) lead to *stiff* systems of ODE, *implicit* methods can significantly decrease computing times due to their generally larger stability region

- In general, the *implicit Euler* method leads to the non-linear system of algebraic equations

$$\underline{\underline{M}}\,\underline{\tilde{c}}(t_1) + \Delta t\,\underline{f}_h(t_1, c_h(t_1)) = \underline{\underline{M}}\,\underline{\tilde{c}}(t_0) \tag{182}$$

that has to be solved for $\underline{\tilde{c}}(t_1)$ iteratively

- In the linear case (cf. (151)), (182) reduces to

$$\left(\underline{\underline{M}} + \Delta t\, \underline{\underline{A}}\right) \underline{\tilde{c}}(t_1) = \underline{\underline{M}}\, \underline{\tilde{c}}(t_0) \tag{183}$$

- The implicit Euler method is A-stable, i.e. unconditionally stable w.r.t. the time-step size $\Delta t$, but is only first order accurate
- A simple extension of the implicit Euler scheme are BDF (Backward Differentiation Formulas) schemes. These schemes are multi-step schemes (just like Adams-Bashforth schemes) of order $S$ that make use of $S-1$ old time-steps. The second-order two-step BDF scheme, for example, is given by

$$c(t_{n+1}) - \frac{2}{3}\Delta t\, \underline{f}_h(t_{n+1}, c_h(t_{n+1})) = \frac{4}{3}c(t_n) - \frac{1}{3}c(t_{n-1}). \tag{184}$$

  As in the case of Adams-Bashforth schemes, their region of absolute stability *shrinks* when $S$ is increased
- Another extension of the implicit Euler schemes is given by *implicit Runge-Kutta* schemes, i.e. where $\underline{\underline{\Gamma}}$ in (158) is a full matrix. In general, such schemes are impractical since each time-step now consists of the solution of non-linear system involving *all* sub-steps. *Diagonally implicit Runge-Kutta* (DIRK), however, assume a (non-strictly) lower triangular $\underline{\underline{\Gamma}}$. In this case, an $S$-stage DIRK scheme consists of $S$ non-linear systems that can be solved *successively*. The most well known examples are:

| | |
|---|---|
| 1 | 1 |
| | 1 |

(a) Implicit Euler

| | | |
|---|---|---|
| 0 | | |
| 1 | $\frac{1}{2}$ | $\frac{1}{2}$ |
| | $\frac{1}{2}$ | $\frac{1}{2}$ |

(b) Crank-Nicolson

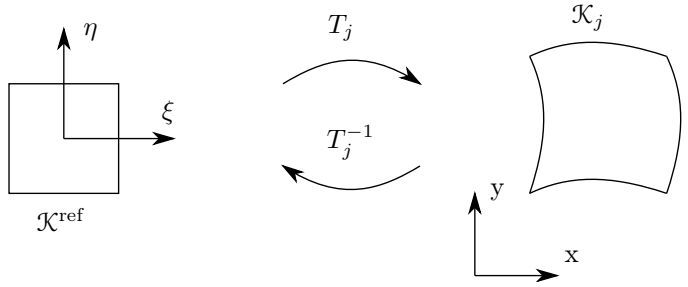
# 5 Implementation Issues

## 5.1 Evaluation of DG-representations

### Basis functions defined on reference elements

- Define reference elements. E.g. in the BoSSS code, we use
  - unit line $\mathcal{K}^{\text{ref}}_{\text{line}} = (-1, 1)$
  - unit square $\mathcal{K}^{\text{ref}}_{\text{sq}} = (-1, 1)^2$
  - unit triangle: $\mathcal{K}^{\text{ref}}_{\text{tri}} := \{\underline{\xi} | \underline{\xi} = \alpha_0 \cdot \underline{\xi}_0 + \alpha_1 \cdot \underline{\xi}_1 + \alpha_2 \cdot \underline{\xi}_2,\ \text{with } \alpha_i \geq 0, \sum_i \alpha_i = 1\}$ and $\underline{\xi}_0 = (0, \frac{4}{3})^T, \underline{\xi}_1 = (\frac{-2}{\sqrt{3}}, \frac{-2}{3})^T, \underline{\xi}_2 = (\frac{2}{\sqrt{3}}, \frac{-2}{3})^T$, i.e. $\mathcal{K}^{\text{ref}}_{\text{tri}}$ is the *convex hull* of $\underline{\xi}_i$.
  - unit cube of $\mathcal{K}^{\text{ref}}_{\text{cube}} = (-1, 1)^3$
  - unit tetrahedron: $\mathcal{K}^{\text{ref}}_{\text{tet}}$ is the convex hull of $(0, 0, \sqrt{2})^T, (0, \frac{4}{3}, \frac{-\sqrt{2}}{3})$, $(\frac{-2}{\sqrt{3}}, \frac{-2}{3}, \frac{-\sqrt{2}}{3}), (\frac{2}{\sqrt{3}}, \frac{-2}{3}, \frac{-\sqrt{2}}{3})$
- Notation:
  - reference element coordinates: $\underline{\xi} = (\xi_0, \xi_1) = (\xi, \eta)$ in 2D, $\underline{\eta} = (\xi_0, \xi_1, \xi_2) = (\eta, \xi, \vartheta)$
- Each cell $\mathcal{K}_j$ is the image of a reference element

$$\mathcal{K}_j = T_j(\mathcal{K}^{\text{ref}})$$



- coordinate transformation

$$\underline{\xi} \mapsto \underline{x} = T_j(\underline{\xi}) \qquad \underline{\xi} = T_j^{-1}(\underline{x}) \tag{185}$$

- we admit that a mesh has more than one reference element: e.g. 2D grid may consist of triangles and quadrilaterals; for the sake of simplicity, however, in what follows we only assume a unique reference element $\mathcal{K}^{ref}$.

- The transformation $T_j$ may be either *affine-linear* (in this case, $\mathcal{K}_j$ is called a linear cell) or *nonlinear* ($\mathcal{K}_j$ is nonlinear or *curved* cell)

- For each reference element, we define a list of polynomials, e.g.

$$\underline{\Phi}^{ref} := \left(\Phi_0^{\text{ref}}, \Phi_1^{\text{ref}}, \Phi_2^{\text{ref}}, \cdots\right) \tag{186}$$

which are orthonormal on the reference element, i.e.

$$\left(\Phi_n^{\text{ref}}, \Phi_m^{\text{ref}}\right)_{\underline{\xi}} = \int_{\mathcal{K}_{\text{ref}}} \Phi_n(\underline{\xi})\Phi_m(\underline{\xi})\, dV = \delta_{nm} \tag{187}$$

- The basis in physical coordinates, in cell $\mathcal{K}_j$,

$$\underline{\Phi}_{j,-} = \left(\Phi_{j,0}, \cdots, \Phi_{j,N_p-1}\right) \tag{188}$$

is obtained via a transformation of a *upper-triangular* matrix $B \in \mathbb{R}^{N_p \times N_p}$

$$\underline{\Phi}_{j,-} = \underline{\Phi}^{\text{ref}} B_{j,-} \tag{189}$$

$$\Phi_{j,m} = \begin{cases} \sum_{n=0}^{N_p-1} \Phi_n^{ref}(T^{-1}(\underline{x}))B_{j,mn} & \text{if } \underline{x} \in \mathcal{K}_j \\ 0 & \text{elsewhere} \end{cases}$$

So that $\Phi_{j,m}$ are orthonormal in the physical space, i.e.

$$(\Phi_{j,m}, \Phi_{l,n}) = \int_\Omega \Phi_{j,m}(\underline{x})\Phi_{l,n}(\underline{x})\, dV = \delta_{jl}\delta_{mn} \tag{190}$$

- The orthogonality in indices $l, j$ is trivial, since $\Phi_{j,m}$ is zero in cell $\mathcal{K}_l$, for $j \neq l$.

- To get orthonormality in the case $j = l$, we have to consider the integral transformation:

$$\int_{\underline{x}\in\mathcal{K}_j} f(\underline{x})\, dV = \int_{\underline{\xi}\in\mathcal{K}^{\text{ref}}} f\left(T_j(\underline{\xi})\right) \cdot \det\left((\partial_{\underline{\xi}} T_j)(\underline{\xi})\right)\, dV \tag{191}$$

- for linear cells, we can represent the transformation $T_j$ as

$$T_j(\underline{\xi}) = \underline{T}_j\underline{\xi} + b_j \tag{192}$$

thus the Jacobian $\partial_{\underline{\xi}} T$ is constant and therefore we can define

$$J_j := \det(\underline{T}_j) \tag{193}$$

and choosing

$$B_{j,nm} = \begin{cases} \frac{1}{\sqrt{J_j}} \text{ for } n = m \\ 0 \text{ otherwise} \end{cases}$$

yields orthogonality an cell$\mathcal{K}_j$ :

$$(\Phi_{j,n}\Phi_{j,m}) = \int_{\mathcal{K}_j} \Phi_{j,n}\Phi_{j,m}\, dV = \frac{1}{\sqrt{J_j}}\frac{1}{\sqrt{J_j}} J_j \int_{\mathcal{K}^{\text{ref}}} \Phi_n^*\Phi_m^*\, dV \tag{194}$$

Therefore, in a linear cell $\mathcal{K}_j$, it is not necessary to store the whole matrix $B_{j,-}$, but only number $J_j$ resp. $\frac{1}{\sqrt{J_j}}$.

We see that in linear cells, *orthogonality* is preserved under the transformation ($n \neq m$, then $\int_{\mathcal{K}^j} \Phi_{j,n}\Phi_{j,m}\, dV = 0$ because $\int_{\mathcal{K}^{\text{ref}}} \Phi_j^{\text{ref}}\Phi_m^{\text{ref}}\, dV$ is zero) , but *orthonormality* ($(\Phi_{j,n}\Phi_{j,m}) = 1$) is in general lost.

- For curved cells, a re-orthonormalization is required. We define a temporary basis

$$\Phi^*_{j,m}(\underline{x}) := \Phi^{\text{ref}}_n\left(T_j^{-1}(\underline{x})\right) \tag{195}$$

and compute the mass matrix $M^*$ of that basis

$$M^*_{n,m} = \int_{\mathcal{K}_j} \Phi^*_{j,n}\Phi^*_{j,m}\, dV = \int_{\mathcal{K}_{\text{ref}}} \Phi^*_n(\underline{\xi})\Phi^*_m(\underline{\xi}\det\left((\partial_\xi T)(\underline{\xi})\right)\, dV. \tag{196}$$

Then we compute the Choleski factorization of $M^*$, i.e. $Q^TQ = M$ and define $B_{j,-} = Q^{-1}$. Then, $\Phi_{j,-} = \Phi^{\text{ref}}B_{j,-}$ is orthonormal in $\mathcal{K}_j$.

(Proof: $(\Phi_n\Phi_m) = (\sum_l \Phi^*_l B_{j,ln}, \Phi^*_k B_{j,km}) = \sum_l\sum_k B_{j,ln}B_{j,km}\underbrace{(\Phi^*_l\Phi^*_k)}_{=M^*_{l,k}}$. This is equal to $(B^T_{j,-}M^*B_{j,-})_{n,m} = \delta_{nm}$ )

- Note that $B_{j,-}$ is *upper-triangular*, so $\Phi_{j,n}$ only depends on $\Phi^{\text{ref}}_m$ with $m \le n$.
- Storage problem for curved cells: memory requirements for the tensor $B_{j,n,m}$, which is an $N_p \times N_p$ - matrix in each cell. This easily requires more storage than the DOFs of the system that we want to solve: e.g. compressible Euler or Navier-Stokes in 3D requires $5N_p$ DOFs per cell. Alternative for curved cells: low-storage schemes, see Warburton (2013). There, one uses a basis ...

$$\Phi^{\text{alt}}_{j,n}(\underline{x}) := \frac{1}{\det\left((\partial T_j)(\underline{x})\right)}\Phi^{\text{ref}}_n(\underline{x}). \tag{197}$$

This, however yields complicated expressions for e.g. $\nabla_{\underline{x}}\Phi^{\text{alt}}_{j,n}$.

- Note that for curved cells, the basis functions $\Phi_{j,n}(\underline{x})$ are *not* polynomials in $x, y, z$ anymore. They are, however, polynomials in $\xi, \eta, \theta$. E.g., assume a 1D-setting and for some $j$ we assume $T_j(\xi) = (\xi+1)^2$, so $T_j^{-1}(x) = \sqrt{x} - 1$; here we have $\xi \in \mathcal{K}^{\text{ref}} = \mathcal{K}^{\text{ref}}_{\text{line}} = (-1, 1)$ and $\mathcal{K}_j = (0, 4)$ and for these domains $T_j$ resp. $T_j^{-1}$ is bijective. If e.g.

$$\Phi^{\text{ref}}_0 = \frac{1}{\sqrt{2}} \tag{198}$$

$$\Phi^{\text{ref}}_1 = \frac{\sqrt{6}}{2}\xi \tag{199}$$

$$\Phi^{\text{ref}}_2 = \frac{3\sqrt{10}}{4}(\xi^2 - \frac{1}{3})\dots \tag{200}$$

Then we have (note that $B_{j,-}$) is

$$\Phi_{j,0}(x) = B_{j,0,0}\ \Phi^{\text{ref}}_0(\sqrt{x}-1) = B_{j,0,0}\cdot\frac{1}{\sqrt{2}} \tag{201}$$

$$\Phi_{j,1}(x) = B_{j,0,1}\ \Phi^{\text{ref}}_0(\sqrt{x}-1) + B_{j,1,1}\ \Phi^{\text{ref}}_1(\sqrt{x}-1) = B_{j,0,1}\cdot\frac{1}{\sqrt{2}} + B_{j,1,1}\cdot\frac{\sqrt{6}}{2}(\sqrt{x}-1) \tag{202}$$

Obviously, there are no constants $B_{j,-,1}$ so that e.g. $\Phi_{j,1}(x)$ becomes a polynomial in $x$.

**Evaluation** We want to evaluate $u(x) = \sum_{jn}\Phi_{j,n}(\underline{x})\tilde{u}_{j,n}$

- For a single evaluation at one point $\underline{x}$, there is not much optimization potential form an algorithm point-of-view.
- If we evaluate in *all cells* and use *the same nodes in reference coordinates* in those cells, i.e. we have nodes

$$\underline{\xi}_0, \dots, \underline{\xi}_{K-1} \text{ and } \underline{x}_{j,k} := T_j(\underline{\xi}_k) \tag{203}$$

We can perform some algorithmic optimization in evaluating

$$u_{j,k} := u(x_{j,k}) \tag{204}$$

To evaluate $u_{j,k}$, we have to compute

$$\begin{aligned}
\forall : u_{j,k} &= \sum_{n=0}^{N_p-1} \Phi_{j,n}(\underline{x}_k)\tilde{u}_{j,n} && (205)\\
&= \sum_{n} \left( \sum_{l=0}^{N_p-1} \Phi_l^{\mathrm{ref}}(\underline{\xi}_k) B_{j,ln} \right) \tilde{u}_{j,n} && (206)\\
&= \sum_{l=0}^{N_p-1} \underbrace{\left( \sum_{n=0}^{N_p-1} B_{j,ln}\tilde{u}_{j,n} \right)}_{=:\tilde{v}_{j,l}} \underbrace{\Phi_l^{\mathbf{ref}}(\underline{\xi}_k)}_{\Phi_{l,k}^{\mathrm{ref}}} && (207)
\end{aligned}$$

the formulation (207) is much faster than (206). The complexity is:

- For (206), we get

| | | |
|---|---|---|
| $\forall j,nk$ | $\Phi_{j,nk}$ | $J \cdot N_p \cdot K \cdot N_p$ |
| $\forall j,k$ | $\sum_n \Phi_{j,nk} \cdot \tilde{u}_{j,n}$ | $J \cdot K \cdot N$ |
| | | $\Rightarrow \mathcal{O}(JKN^2)$ |

- For (207), we have

| | | |
|---|---|---|
| $\forall j,l$ | $\tilde{v}_{j,l}$ | $J \cdot N_p \cdot N_p$ |
| $\forall j,k$ | $\sum_l \tilde{v}_{j,l}\Phi_{l,k}^{\mathrm{ref}} \cdot \tilde{u}_{j,n}$ | $J \cdot K \cdot N_p$ |
| | | $\Rightarrow \mathcal{O}(JKN_p) or \mathcal{O}(JN_p^2)$ |

- For linear cells, using $B_{j,ln} = \frac{1}{\sqrt{J_j}}\delta_{ln}$, we can further optimize

$$\forall j,k : u_{j,k} = \sum_{l=0}^{N_p-1} \left( \frac{1}{\sqrt{J_j}} \tilde{u}_{j,n} \right) \Phi_{l,k}^{\mathrm{ref}} \tag{208}$$

- In terms of runtime, it is beneficial to use hardware-optimized libraries BLAS libraries, which offer e.g. optimized matrix-matrix multiplications. Therefore, formulas like (208) have to be written as matrix-matrix multiplications, i.e.

$$\begin{bmatrix} u_{0,0} & \cdots & u_{0,K-1} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ u_{J-1,0} & \cdots & u_{J-1,K-1} \end{bmatrix} = \begin{bmatrix} \tilde{v}_{0,0} & \cdots & \tilde{v}_{0,N_p-1} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \tilde{v}_{J-1,0} & \cdots & \tilde{v}_{J-1,N_p-1} \end{bmatrix} \cdot \begin{bmatrix} \Phi_{0,0}^{\mathrm{ref}} & \cdots & \Phi_{0,K-1}^{\mathrm{ref}} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \Phi_{N_p-1,0}^{\mathrm{ref}} & \cdots & \Phi_{N_p-1,K-1}^{\mathrm{ref}} \end{bmatrix} \tag{209}$$

This can be up to 20 times faster than a "naive" implementation using for-loops. (BLAS:Basic Linear Algebra Subsystem; BLAS stands for the definition of the standard and is a *reference implementation.* Optimized implementations of BLAS are e.g. Intel MKL, AMD ACML, ATLAS, ....)

## 5.2 Quadrature

- approximation of integration: nodes $\left(\underline{\xi}_0, \ldots, \underline{\xi}_{k-1},\right) =: \underline{\underline{\xi}}$, weights $(w_0, \ldots, w_{k-1}) =: \underline{w}$

$$\int_{\mathcal{K}^{\mathrm{ref}}} f(\underline{\xi})\, d\mathcal{V} \approx \sum_{k=0}^{K-1} f(\xi_k) w_k := \int_{(\underline{\underline{\xi}},\underline{w})}^{\mathrm{num}} f \tag{210}$$

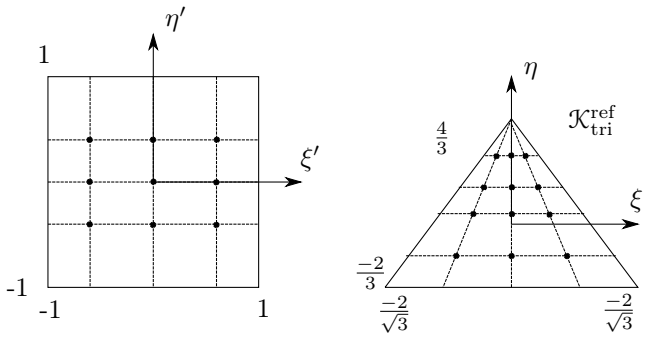
- the rule $\left(\underline{\underline{\xi}}, \underline{w}\right)$ has order $p$, if for all polynomials $f(\underline{\xi})$ with degree $f \le g$ we have $\int_{(\underline{\underline{\xi}},\underline{w})}^{\mathrm{num}} f = \int_{\mathcal{K}^{\mathrm{ref}}} f\, d\mathcal{V}$.
- In 1D: Gauss rules are most efficient: $p = 2K - 1$

**Rules for square and cube elements.** For $\mathcal{K}^{\text{ref}}_{\text{sq}}$, $\mathcal{K}^{\text{ref}}_{\text{cube}}$, tensorized rules: e.g., assume 1D-rule $\left(\underline{\xi}^{\text{like}}, \underline{w}^{\text{like}}\right)$ of order $p$, with $K$ nodes, then we have a rule $\left(\underline{\underline{\xi}}^{\text{sq}}, \underline{w}^{\text{sq}}\right)$ for $\mathcal{K}^{\text{ref}}_{\text{sq}}$ with
$\underline{\xi}^{\text{sq}}_{lK+k} = \left(\xi^{\text{line}}_l, \xi^{\text{line}}_k\right)$, $w^{\text{sq}}_{lK+k} = w^{\text{line}}_l, w^{\text{line}}_k$.

So, the efficient 1D-Gauss rules can also be used in 2D resp. 3D: $p = 2\sqrt[D]{K} - 1$.

**Rules for triangle and tetrahedron:** For $\mathcal{K}^{\text{ref}}_{\text{tri}}, \mathcal{K}^{\text{ref}}_{\text{tetra}}$ the situation is less optimal.

- One could use e.g. collapsed rules:




  with the transformation

$$\xi = \xi' \cdot (1 - \eta') \frac{1}{\sqrt{3}}, \quad \eta = \eta' + \frac{1}{3} \tag{211}$$

  and the integral transformation

$$\int_{\mathcal{K}^{\text{ref}}_{\text{tri}}} f \, dV = \int_{\xi'=-1}^{1} f\left(\left(\xi' \cdot (a - \eta') \frac{1}{\sqrt{3}}, \eta' + \frac{1}{3}\right)\right) \cdot \frac{1 - \eta'}{\sqrt{3}} d\eta' d\xi' \tag{212}$$

  Then, form the 1D rule $\left(\underline{\underline{\xi}}^{\text{tri}}, \underline{\underline{w}}^{\text{tri}}\right)$ with

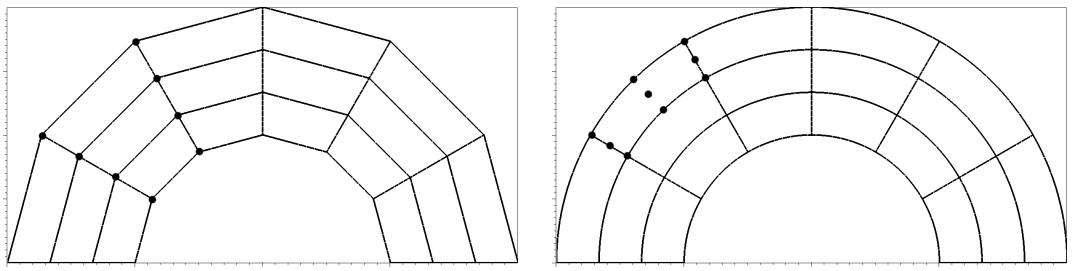
$$\underline{\xi}^{\text{tri}}_{lK+k} = \left(\xi^{\text{line}}_l (1 - \xi^{\text{line}}_k) \frac{1}{\sqrt{3}}, \xi^{\text{line}}_k + \frac{1}{3}\right) \qquad w^{\text{tri}}_{lK+k} = \left(w^{\text{line}}_l w^{\text{line}}_k \frac{1 - \xi^{\text{line}}_k}{\sqrt{3}}\right) \tag{213}$$

  Due to the Integral transformation term $\frac{1-\eta'}{\sqrt{3}}$, we one quadrature order is best, $p = 2\sqrt{K} - 2$.

- A disadvantage of collapsed rules is (1) that lots of nodes accumulate around one vertex, and (2) that due to the integral transformation the weights may differ by several magnitudes, possibly causing high round-off-errors.
- Rotationally symmetric rules for triangle and tetrahedron [1] with Gauss-like properties are still a research topic.
- " Poor man "-solution: construct rules where $K = Np$ ; not very efficient, but robust and symmetric.

## 5.3 Curved grids

For a high-order method, one needs a high-order representation of the geometry (see exercise 8). A low order method (e.g. finite volume, DG with $p \leq 1$) will not notice a difference between the following meshes; a high-order-method (DG, with $p \geq 2$) will produce different convergence rates on the following grid



[1] e.g. for $\mathcal{K}^{\text{ref}}_{\text{tri}}$: we get the same nodes after a rotation of $\frac{2}{3}\pi$, i.e. 120 degrees

On the left, elements are discretized with 5 nodes per element, the mappings $T_j$ have a *degree in each variable* of 1. On the right, 9 nodes per element are used, the mappings $T_j$ have*a degree in each variable* of 2.

**Degree of a polynomial** Consider the monomial

$$m := \xi^{\alpha_1} \cdot \eta^{\alpha_2} \tag{214}$$

- (In this lecture) we define the *degree* of $m$ as $\deg = \alpha_1 + \alpha_2$. (In other sources, this is sometimes called "absolute degree").
- (In this lecture) we define the *degree in each variable* $\deg'(m) = \max\{\alpha_1, \alpha_2\}$. (In other sources, this is sometimes called "degree" )
- Unfortunately, there is no common convention.

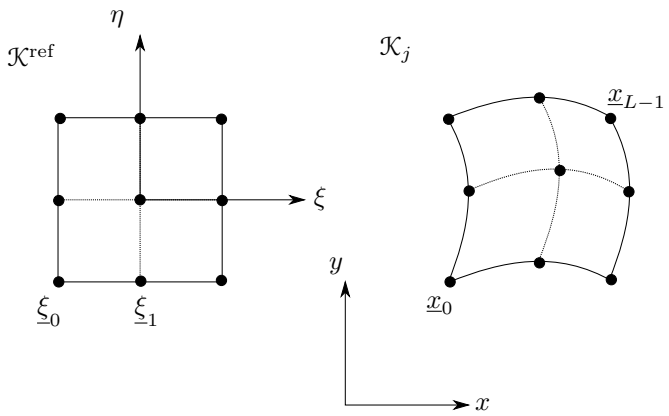
**Mappings form reference to physical elements**

- (affine) linear elements:

$$T_j(\underline{\xi}) = \underline{T_j} \cdot \underline{\xi} + O_j \tag{215}$$

  Jacobian and Jacobian determinant are easy and cheap to compute

- curved elements can be described by polynomials using a interpolation




  Nodes in the reference element $\underline{\xi}_0, \ldots, \underline{\xi}_{L-1}$ should be mapped to nodes in the physical space $\underline{x}_0, \ldots, \underline{x}_{L-1}$ so that

$$T_j(\underline{\xi}_l) = \underline{x}_l \tag{216}$$

- Notation (w.l.o.g. only in 2D, i.e. D=2)

$$T_j(\underline{\xi}) = \begin{bmatrix} T_{x,j}(\underline{\xi}) \\ T_{y,j}(\underline{\xi}) \end{bmatrix} = \begin{bmatrix} T_{x,j}(\xi, \eta) \\ T_{y,j}(\xi, \eta) \end{bmatrix} \tag{217}$$

- We use nodal polynomials $\phi_0, \ldots, \phi_{L-1}$ which fulfill the property

$$\phi_{l_1}(\xi_{l_2}) = \delta_{l_1, l_2}. \tag{218}$$

  Then, we can set

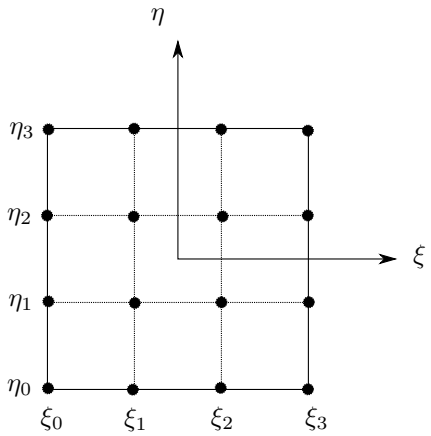
$$T_{x,j}(\underline{\xi}) = \sum_{l=0}^{L-1} \phi_l(\underline{\xi}) x_l. \tag{219}$$

  Obviously, we get

$$T_j(\underline{\xi}_l) = \underline{x}_l \tag{220}$$

**Properties of the nodal space**

- for Quadrilaterals, we can use tensor spaces:

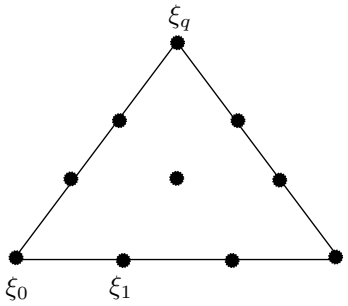



  use the Lagrange Polynomials $\ell_0, \ldots, \ell_3$ (see Eq. (10)) for the nodes $\xi_0 = -1, \xi_1 = \frac{-1}{3}, \xi_2 = \frac{1}{3}, \xi_3 = 1$. Then, we can define e.g.

$$\phi^{\text{quadN}}_{l_1*3+l_2}(\xi, \eta) = \ell_{l_1}(\xi) \cdot \ell_{l_2}(\eta) \tag{221}$$

- for triangles, we can e.g. compute modal polynomials from monomials:

$$m_0 = 1,\ m_1 = \xi,\ m_2 = \eta,\ m_3 = \xi^2,\ m_4 = \xi\eta,\ m_5 = \eta^2,\ m_6 = \eta^3,\ m_7 = \xi^2\eta,\ m_8 = \xi\eta^2,\ m_9 = \eta^9 \tag{222}$$




- Note that the number of monomials up to degree $p$ is the same as the number of nodes in a triangle (as shown above) with $p+1$ nodes on each side. We can make the ansatz

$$\phi^{\text{tri10}}_l(\underline{\xi}) = \sum_l \alpha_l m_l(\underline{\xi}) \tag{223}$$

  and obtain the constants $\alpha_l$ by solving the linear system

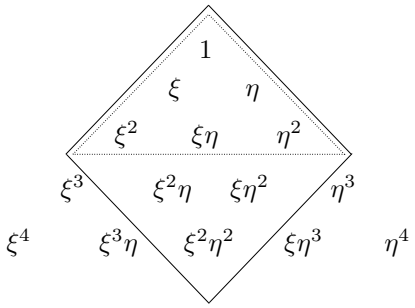
$$\begin{bmatrix} m_0(\underline{\xi}_0) & \ldots & m_q(\underline{\xi}_0) \\ \vdots & & \vdots \\ m_0(\underline{\xi}_q) & \ldots & m_q(\underline{\xi}_q) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_q \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{224}$$

- The so-called Vandermonde-matrix $V_L$ is invertible if the nodes $\underline{\xi}_l$ are pair-wise different, but it has a bad condition, i.e. the determinant close to zero. This, however also depends on the choice of nodes, blended nodes usually deliver a better condition number. (see Hesthaven and Warburton (2008), page 169 ff). If the Vandermonde- matrix is used to construct modal polynomials, it should be solved with high precision- either analytical or with enhanced numerical accuracy (using e.g. 16 Byte-floating point numbers).

- For both examples shown alone (quad16 and tri10) the modal basis functions

$$\phi_0^{\text{quad10}},\ \ldots,\ \phi_{15}^{\text{quad16}} \quad \text{resp.} \quad \phi_0^{\text{tri10}},\ \ldots,\ \phi_9^{\text{tri10}} \tag{225}$$

span a polynomial space. In the Pascal triangle




- $(\phi_l^{\text{tri10}})_l$ spar and space of dimension 10, containing all polynomials up to degree 3
- $(\phi_l^{\text{quad16}})_l$ span a space of dimension 16, containing all polynomials up to degree 3 in each variable.

**Invertibility**

- for higher order nonlinear mappings $T_j$, the inverse mapping $T_j^{-1}$ is usually not known. It can be computed e.g. by a Newton method, which is rather expensive. Therefore, the implementation should not rely on $T_j^{-1}$, Assume e.g. we want to compute

$$\tilde{f}_{j,m} := \Phi_{j,m}\, dV = \int_{\underline{x}\in\mathcal{K}_j} f(\underline{x}) \left( \sum_n B_{j,mn} \Phi_n^{\text{ref}}(T_j^{-1}(\underline{x}) \right) d\underline{x} \tag{226}$$

Since we "pull back " calculations onto the reference element, this is not a problem:

$$\tilde{f}_{jm} = \int_{\underline{\xi}\in\mathcal{K}^{\text{ref}}} \det(\partial T_j)(\underline{\xi}) f(T_j(\underline{\xi})) \cdot \left( \sum_n B_{j,mn} \Phi_n^{\text{ref}}(\underline{\xi}) \right) d\underline{\xi} \tag{227}$$

**Transformation of basis polynomials**

- on curved elements, the basis $\Phi_{j,m}$ is usually not polynomial in $x,\ y,\ z$ anymore. Assume the 1D example

$$T_j(\xi) = \left( \frac{\xi}{2} + \frac{3}{2} \right)^2 \tag{228}$$

which maps $k^{\text{line}} = (-1, 1)$ to $k_j = (1, 4)$. Obviously,

$$T_j^{-1}(x) = 2\left( \sqrt{x} - \frac{3}{2} \right) \tag{229}$$

If we assume

$$\Phi_1^{\text{line}}(\xi) = \sqrt{\frac{3}{2}}\xi \tag{230}$$

then we have (we momentarily assume $B_{j,mn} = \delta_{mn}$).

$$\Phi_{j1}(x) = \Phi_1^{\text{line}}(T_j^{-1}(x)) = \sqrt{3}\sqrt{2}(\sqrt{x} - \frac{2}{3}), \tag{231}$$

i.e. $\Phi_{j1}$ is a polynomial in $\sqrt{x}$!

- Alternatively, one can define the $\Phi_{j,n}$ without any reference element as polynomials in $x,\ y,\ z$ in this case, the basis functions are not "adapted " to the geometry and the transformation is usually worse.

## 5.4 Treatment of Volume terms

We want to compute the volume term in equation (83)

$$F_{j,m} := -\int_{\mathcal{K}_j} \underline{f} \cdot \nabla \Phi_{j,m} \, dV \tag{232}$$

we only focus on one cell $\mathcal{K}_j$, we omit the index $j$ for the remains of the section,

**Gradient of the test functions:** By using the chain rule, we get

$$\nabla_{\underline{x}} \Phi_n(x) = \sum_n B_{m,n} \nabla_{\underline{x}} \left( \Phi_n^{\text{ref}} \circ T^{-1} \right) \tag{233}$$

$$= \sum_n B_{m,n} \left( \partial T^{-1} \right)^T \nabla_{\underline{\xi}} \Phi_n^{\text{ref}} \circ T^{-1} \tag{234}$$

For the Jacobian $\partial(T^{-1})$ we can use the identity $\partial(T^{-1}) = (\partial T)^{-1}$. This is important for curved cells, since there is usually no analytic expression for $T^{-1}$.

**Volume Terms** Using integral transformations, we get

$$F_{j,m} = -\sum_n B_{m,n} \int_{\underline{\xi} \in \mathcal{K}^{\text{ref}}} \det(\partial T_j) \, (\partial T_j) \, (\underline{\xi}) \underline{f}(\underline{\xi}) \cdot \left[ \left( (\partial T)(\underline{\xi}) \right)^{-1,T} \cdot \nabla_{\underline{\xi}} \Phi_n^{\text{ref}}(\underline{\xi}) \right] dV = * \tag{235}$$

and using a quadrature rule $\left( \xi^{\text{ref}}, w^{\text{ref}} \right)$ with K nodes

$$* = -\sum_{n=0}^{N_{p-1}} B_{m,n} \sum_{K=0}^{K-1} \det(\partial T)_k \left( \sum_{d,e} f_{k,d} ((\partial T)^{-1})_{kde} \partial_{\xi e} \Phi_{n,k}^{\text{ref}} \right) w_k = * \tag{236}$$

where we use the notation $\det(\partial T)(\underline{\xi}_k) =: \det(\partial T)_k$, $f_{k,d}$ is the d-th component of $\underline{f}$ at node k, etc. Using the adjungate[2] matrix, which is defined as

$$\text{adj}(A) = \det(A) \cdot A^{-1}, \tag{237}$$

but simpler to compute one can simplify

$$* = -\sum_{n=0}^{N_{p-1}} B_{m,n} \sum_{K=0}^{K-1} \underbrace{\left( \sum_d f_{kd} \, \text{adj}(\partial T)_{kde} \right) w_k}_{=: f'_{kde}} \qquad \partial_{\xi e} \Phi_{n,k}^{\text{ref}} \tag{238}$$

For performance reasons it is important to factor out terms like $f'_{kde}$. In other words: it is more efficient to transform *one* flux to the reference element, than to transform $N_p$ test function gradients to the physical element.

**Choice of Quadrature order** If the flux $\underline{f}$ is polynomial, e.g. $\underline{f}(\underline{x}) = \underline{f}(u(\underline{x})) = u^2(\underline{x})$, the integral

$$\underline{f} \cdot \left( \text{adj}(\partial T)^T \cdot \nabla_{\xi} \Phi_n \right) \tag{239}$$

is polynomial in the reference coordinates $\underline{\xi}$ and can be integrated exactly if a sufficiently accurate quadrature rule is used. Given that $u$ is of degree $p$, in this example we get the following degree for the integrand:

| | |
|---|---|
| $\underline{f}$: | 2p |
| $\text{adj}(\partial T)^T$: | $(\deg(T) - 1) \cdot D$ |
| $\nabla_{\xi} \Phi$ | $p - 1$ |
| sum | $2p + (\deg(T) - 1) + p - 1$ |

[2] or classical adjoint, or adjunct matrix but *not* adjoint matrix!

# 6 Linear, scalar equations of second order

## 6.1 Prototype problems

**Poisson equation:** domain $\Omega \subseteq \mathbb{R}^D$, $D = 1, 2, 3$ (spatial dimension), boundary of $\Omega : \partial\Omega = \Gamma_D \cup \Gamma_N$

$$\left\{ \begin{array}{rcll} -\Delta u & = & g_\Omega & \text{in } \Omega \\ u & = & g_D & \text{on } \Gamma_D \quad \text{Dirichlet-boundary} \\ \nabla u \cdot \underline{n}_{\partial\Omega} & = & g_N & \text{on } \Gamma_N \quad \text{Neumann-boundary} \end{array} \right. \tag{240}$$

**Heat equation:**

$$\left\{ \begin{array}{rcll} \partial_t u - \mu\Delta u & = & g_\Omega & \text{in } \Omega \\ u & = & g_D & \text{on } \Gamma_D \\ \nabla u \cdot \underline{n}_{\partial\Omega} & = & g_N & \text{on } \Gamma_N \end{array} \right. \tag{241}$$

**Remarks:**

- $\Gamma_D$ and $\Gamma_N$ are disjoint, i. e. $\Gamma_N \cap \Gamma_D = \emptyset$
- it can be that either $\Gamma_N = \emptyset$ (only Dirichlet) or $\Gamma_D = \emptyset$ (only Neumann)
- only Neumann ($\Gamma_N = \partial\Omega$) is a special case:
  - solution only unique up to a constant, i. e. if $u$ is a solution, so is $u + const.$.
  - compatibility of $g_N$ and $g_\Omega$ has to be fulfilled

$$-\int_\Omega g_\Omega \, dV = \oint_{\partial\Omega} g_N \, dA, \tag{242}$$

$$\text{because } \int_\Omega g_\Omega \, dV = -\int_\Omega \Delta u \, dV = -\int_\Omega \operatorname{div}(\nabla u) \, dV = -\oint_{\partial\Omega} \nabla u \cdot \underline{n}_{\partial\Omega} \, dA = -\oint_{\partial\Omega} g_N \, dA.$$

- in the chapter: only Poisson and Heat equation, but methods and theory apply to wider class of problems.

**Major difference to scalar convection:**

- scalar conv. is *hyperbolic*, i.e. information travels at finite speed: If one adds distortion in one cell $\mathcal{K}_j$, in timestep $\vartheta$, in timestep $\vartheta + 1$ only direct neighbours of $\mathcal{K}_j$ are affected, in timestep $\vartheta + 2$ the neighbours of neighbours, etc. Therefore, explicit timestepping works well.
- Poisson problem is *elliptic*; Heat equation is *parabolic*; Information travels at infinite speed. Small, local change in boundary data $g_D$, $g_N$ affects solution in the whole domain, after infinitely short time. Therefore, we need linear solvers / implicit timestepping.

**Remark:** representation of the solution $u(\underline{x})$: we assume the basis functions $\Phi_{j,n}$ to be defined on the whole domain $\Omega$, so we can write

$$u(\underline{x}) = \sum_{j=0}^{J-1} \sum_{n=0}^{N_p-1} \Phi_{j,n}(\underline{x}) \cdot \tilde{u}_{j,n} = \underline{\Phi} \cdot \tilde{u} \tag{243}$$

where

$$\underline{\Phi} = \left(\Phi_{0,0}, \ldots, \Phi_{0,N_{p-1}}, \Phi_{1,0}, \ldots, \Phi_{J-2,N_{p-1}}, \Phi_{J-1,0}, \ldots, \Phi_{J-1,N_{p-1}}\right)$$

is a row vector of basis polynomials and $\tilde{u} = \left(\tilde{u}_{0,0}, \ldots, \tilde{u}_{J-1,N_{p-1}}\right)^T$ a column- vector in $\mathbb{R}^{J \cdot N_p \times 1}$ of DG coordinates. I.e., we assume that

$$\Phi_{j,n}(\underline{x}) = \left\{ \begin{array}{ll} \neq 0, \text{ polynomial in } \underline{x} & \text{in cell } \mathcal{K}_j \\ = 0 & \text{in } \Omega \setminus \mathcal{K}_j \end{array} \right.$$

**Questions:**

- we search for a linear system
$$M \cdot \tilde{u} = \tilde{f} \tag{244}$$
with $M \in \mathbb{R}^{J \cdot N_p \times J \cdot N_p}$ and $\tilde{f} \in \mathbb{R}^{J \cdot N_p}$ that is a discrete version of the Poisson equation with boundary conditions.
- how to compute matrix $M$?
- consistency: if the exact solution is polynomial, the DG method should find this exactly up to round-off-errors.
- stability: the matrix must be invertible, ($\det(M) \neq 0$), independent of polynomial order and grid resolution.

## 6.2 Variational formulation of the Poisson equation, continuous setting

DG-methods can be written in multiple ways, if explicit timestepping (like in previous chapter) is used, the methods are usually formulated cell-by-cell. For implicit timestepping or steady-state problems and for numerical analysis, the variational formulation is more convenient. First, a general look on variational form (also called weak form), for $\Gamma_D = \partial\Omega$ and $g_D = 0$, i.e. homogeneous Dirichlet boundary conditions.

Idea: multiply Poisson equation with test function, and perform partial integration of $\Omega$. To handle boundary conditions, we define a function space that only admits functions $u$ with $u|_{\partial\Omega} = 0$. (Sobolev-spaces, idea from Sergej Sobolev, est. 1950)

**Definition (Sobolev-Space):**

$$H_0^1(\Omega) := \{u \in L^2(\Omega);\ \|u\|_2 + \|\nabla u\|_2 < \infty;\ u|_{\partial\Omega} = 0\}. \tag{245}$$

**Weak formulation of Poisson problem:** Regarding the Poisson problem, we search for a weak solution $u$ in the Sobolev-space $H_0^1(\Omega)$: Find $u \in H_0^1(\Omega)$ so that

$$a(u, v) = \int_\Omega g_\Omega v \, dV \text{ for all } v \in H_0^1(\Omega) \tag{246}$$

with

$$a(u, v) := \int_\Omega \nabla u \cdot \nabla v \, dV. \tag{247}$$

(Remark: $-\int_\Omega \Delta u v \, dV = -\int_\Omega \operatorname{div}(\nabla u) v \, dV = \underbrace{-\oint_{\partial\Omega} (\nabla u \cdot \underline{n}_{\partial\Omega}) v \, dA}_{=0 \text{ since } v=0 \text{ on } \partial\Omega} + \int_\Omega \nabla u \cdot \nabla v \, dV = a(u, v)$.)

## 6.3 Global variational formulation, discrete setting

The goal is to find a formulation similar to (246) for the DG problem. This will allow us to answer the questions stated at the end of section 6.1. We start with a generic, time-independent equation

$$\operatorname{div}(\underline{f}(u)) = g_\Omega \tag{248}$$

As in section 2.3, we multiply by a test function $\Phi_{jm}$, integrate over $\mathcal{K}_j$, apply integration-by-parts and introduce a numerical flux $\hat{F}$ to obtain

$$\oint_{\partial\mathcal{K}_j} \hat{F}(u_j^-, u_j^+, \underline{n}_{\partial\mathcal{K}_j}) \Phi_{jm} \, dA - \int_{\mathcal{K}_j} \underline{f}(u_j) \cdot \nabla \Phi_{jm} \, dV = \int_{\mathcal{K}_j} g_\Omega \Phi_{jm} \, dV. \tag{249}$$

This is a cell-by-cell-formulation. To obtain a global formulation we have to sum over all cells.

In addition, to replace the test-function $\Phi_{jm}$ with an arbitrary test function $v$, we use the representation $v = \sum_j \sum_m \tilde{v}_{jm}\Phi_{jm}$. Therefore, we also multiply equation (249) with a constant $\tilde{v}_{jm}$ and sum over $m$. First, for the right-hand-side of equation (249)

$$\sum_j \sum_m \tilde{v}_{jm} \int_{\mathcal{K}_j} g_\Omega \Phi_{jm}\, dV = \int_\Omega g_\Omega (\sum_j \sum_m v_{jm}\Phi_{jm})\, dV = \int_\Omega g_\Omega v\, dV \tag{250}$$

Next, the volume part of the left-hand-side of (249). This works exactly as for the right-hand-side. However, we have to be careful about the interpretation of the gradient.

**Definition (broken gradient $\nabla_h$ )** For some function $f$ which is sufficiently smooth within the cells of $\mathfrak{K}_h h$, but may be discontinuous at the cell boundaries, i.e.

$$f \in \mathcal{C}^1 \left(\Omega \setminus (\cup_j \partial\mathcal{K}_j)\right) \tag{251}$$

we define

$$\nabla_h f = \begin{cases} 0 & \text{on } \cup_j \partial\mathcal{K}_j \\ \nabla f & \text{elsewhere} \end{cases} \tag{252}$$

To keep notation simple, we may omit the $h$-index of the broken gradient $\nabla_h$ in all weak formulations.

**Example (broken Gradient)** We consider a grid, consisting of two cells, $\mathcal{K}_0 = (-1, 0)$, $\mathcal{K}_1 = (0, 1)$, $\mathcal{K}_h = \mathcal{K}_0, \mathcal{K}_1$, and the function

$$v(x) = \begin{cases} -1 & x \le 0 \\ x & x > 0 \end{cases} \quad \in \mathbb{P}_1(\mathfrak{K}_h h). \tag{253}$$

Obviously, $v(x)$ has a jump at $x = 0$, so $\nabla v$ is not defined at $x = 0$. (One could, of course, use distribution theory to define the gradient. This is, however, not useful in the DG context.) However,

$$\nabla_h v(x) = \begin{cases} 0 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}, \tag{254}$$

since the point $x = 0$ is at a cell boundary and therefore omitted.

Using the broken gradient, we can take the sum of the volume part of (249) over all $j$ and $m$ to obtain

$$-\sum_j \sum_m \tilde{v}_{jm} \int_{\mathcal{K}_j} \underline{f}(u_j) \cdot \nabla\Phi_{jm}\, dV = \int_\Omega \underline{f}(u)(\sum_j \sum_m \tilde{v}_{jm} \nabla_h \Phi_{jm})\, dV = \int_\Omega \underline{f}(u) \cdot \nabla_h v\, dV \tag{255}$$

(Using the notation convention, we may write $\nabla v$ instead of $\nabla_h v$). Next, we will consider the surface part; we therefore define

- The sets of all edges $\Gamma$ and all inner edges $\Gamma_i$

$$\Gamma := \cup_j \partial\mathcal{K}_j \text{ and } \Gamma_i = \Gamma \setminus \partial\Omega \tag{256}$$

- a normal field on $\Gamma$, $\underline{n}_\Gamma$: on $\partial\Omega$, $\underline{n}_\Gamma = \underline{n}_{\partial\Omega}$; on $\Gamma_i$, we have to pick one normal (out of two options) For each edge $\mathbb{E} \subseteq \Gamma$, we therefore have an "in" and an "out" – cell.
- We re-define the limits $u^+$ and $u^-$ (eqs. (78) and (79), section 2.3) with respect to $\Gamma$:

$$u^-(\underline{x}) = \lim_{\substack{\epsilon\to 0 \\ \epsilon>0}} (u(\underline{x} - \underline{n}_\Gamma \epsilon)) \tag{257}$$

$$u^+(\underline{x}) = \lim_{\substack{\epsilon\to 0 \\ \epsilon>0}} (u(\underline{x} + \underline{n}_\Gamma \epsilon)) \tag{258}$$

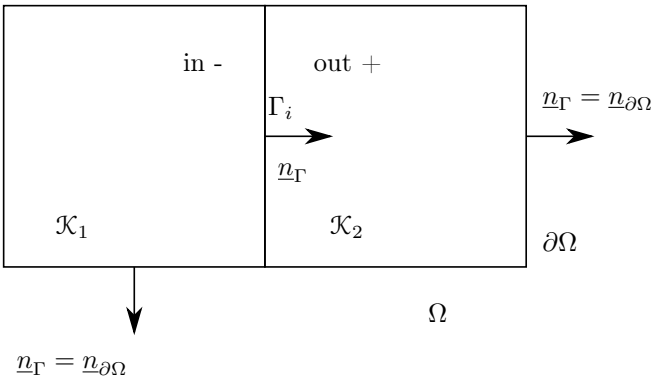
$$\tag{259}$$

Note that on $\Gamma_i$, the value of the DG-Fields is *not* unique; the limit on the in-side and on the out-side are usually different.

- This re-definition of in- and out-values also affects the average- and the jump-operator (113) and (114). We also define jump and average on the boundary:

$$\begin{aligned} \text{on } \Gamma_i : [\![u]\!] &:= (u^- - u^+) \quad & \{u\} &:= \frac{1}{2}(u^- + u^+) \\ \text{on } \partial\Omega : [\![u]\!] &:= u^- & \{u\} &:= u^- \end{aligned}$$

- Illustration for two cells:




- Note that

$$\hat{F}(a, b, \underline{n}_{\partial\mathcal{K}}) = \begin{cases} \hat{F}(a, b, \underline{n}_\Gamma) & \underline{n}_{\partial\mathcal{K}} = \underline{n}_\Gamma \\ -\hat{F}(a, b, \underline{n}_\Gamma) & \underline{n}_{\partial\mathcal{K}} = -\underline{n}_\Gamma \end{cases} \tag{260}$$

- For any discontinuous function $v$, we have

$$\sum_j \oint_{\partial\mathcal{K}_j} v \underline{n}_{\partial\mathcal{K}_j} \mathrm{dS} = \oint_\Gamma [\![v]\!] \, \underline{n}_\Gamma \mathrm{dS} \tag{261}$$

Note that this is independent of the choice on $\underline{n}_\Gamma$. Finally, we are ready to sum up the surface part of the left-hand side of (249) and obtain:

$$\sum_j \sum_m \tilde{v}_{jm} \oint_{\partial\mathcal{K}_j} \hat{F} \Phi_{jm} \, dA = \int_\Gamma \hat{F} \, [\![v]\!] \, dA \tag{262}$$

combining all above, we obtain

$$\underbrace{\oint_\Gamma \hat{F}(u^-, u^+, \underline{n}_\Gamma) \, [\![v]\!] \, dA - \int_\Omega \underline{f}(u) \cdot \nabla v \, dV}_{=: a_h(u,v)} = \int_\Omega g_\Omega v \, dV \tag{263}$$

we obtain the global variational formulation: find $u \in \mathbb{P}_p(\mathcal{K}_h)$ so that

$$a_h(u, v) = \int g_\Omega v \, dV \quad \forall v \in \mathbb{P}_p(\mathcal{K}_h) \tag{264}$$

## 6.4 The Lax-Milgram theorem

**A naive approach** As a motivation for a deeper study of theory, we will come up with a naive discretization of the Poisson problem and see that this fails. We notice that $\Delta u = \mathrm{div}(\nabla u)$, therefore we use in the (248)

$$\underline{f} = -\nabla u, \tag{265}$$

$$\hat{F} = -\{\nabla u\} \cdot \underline{n}_\Gamma, \tag{266}$$

inserting that into (263) yields

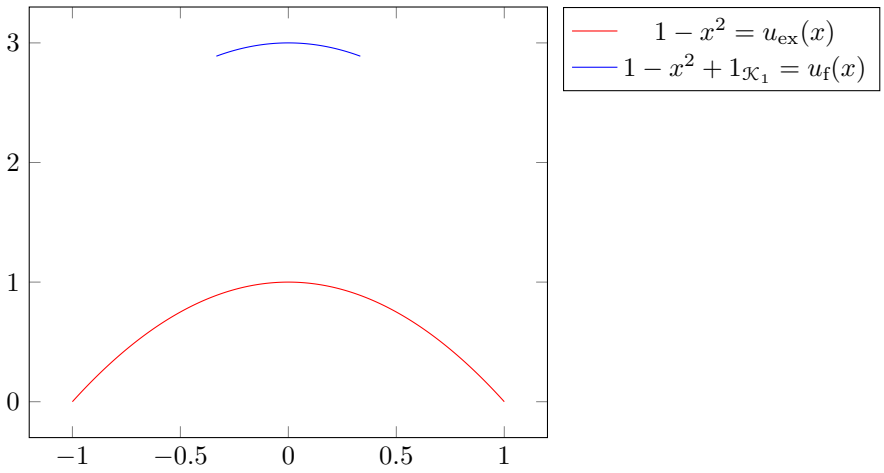
$$\underbrace{-\oint_\Gamma \{\nabla u\} \cdot \underline{n}_\Gamma \, [\![v]\!] \, dA + \int_\Omega \nabla u \cdot \nabla v \, dV}_{=: a_{\mathrm{naive}}(u,v)} = \int_\Omega g_\Omega v \, dV \tag{267}$$

**Example(instability of the naive discretization)** 1D-Poisson-Problem $-\Delta u = 2$ in the $\Omega = (-1, 1)$; 3 cells $\mathcal{K}_0 = (-1, -\frac{1}{3})$, $\mathcal{K}_1 = (-\frac{1}{3}, \frac{1}{3})$, $\mathcal{K}_2 = (\frac{1}{3}, 1)$; boundary conditions $u(-1) = u(1) = 0$; exact solution $u_{\text{ex}}(x) = 1 - x^2$; in DG-space with $p = 2$, we should be able to get this solution exactly up to round-off-errors.

We will see that the discretization is *consistent*, but *not stable*.

- *Consistency:* $u_{\text{ex}}$ is a (exact) solution $\Leftrightarrow a_{\text{naive}}(u_{\text{ex}}, v) = \int f v \, dV$ should hold for every test function $v$.
- *Stability* will be guarantee the uniqueness of the numerical solution. (more precise: see below)

The naive form is completely insensitive to jumps between cells. We add a constant $c$ in cell $\mathcal{K}_2$, $u_{\text{f}}(x) = u_{ex}(x) + 1_{\mathcal{K}_2}(x) \cdot c$ ($1_S(x) = 1$ if $x \in S$, 0 else)




Then, $a_{\text{naive}}(u_{\text{f}}, v) = \int_\Omega f v \, dV$ for all $v$; i.e. the discretization allows infinitely many, wrong solutions $\rightarrow$ not stable resp. unstable.

We see that it is not so obvious whether a discretization for the Laplace operator works or not. To get a better understanding, some theory is required. One of the most important concepts is coercivity. If this property is fulfilled, the Lax-Milgram-theorem states that the variational problem has a unique solution.

**Definition: coercivity.** a bilinear form $a(-,-)$ is *coercive*, if there is a constant $\gamma > 0$ so that in *some suitable norm* $|||-|||_*$ the inequality

$$a(u, u) \geq \gamma \, |||u|||_*^2 \; \forall \; u \tag{268}$$

holds.

**Theorem: Lax-Milgram.** If a bilinear form $a(-,-)$ is coercive, the solution to the variational problem $\left(a(u, v) = \int g_\Omega v \, dV \; \forall v\right)$ is unique.

**Note:** To prove coercivity, one is free to choose a norm that is convenient for the proof. This is usually not the $L^2$-, or the $H^1$-norm. E.g., for showing the coercivity of the symmetric interior penalty method (see below, section 6.5), the so-called SIP-norm is introduced:

$$|||u|||_{\text{sip}} := \left( \|\nabla u\|^2_{L^2(\Omega)^D} + \oint_\Gamma \frac{1}{h_e} [\![u]\!]^2 \, dA \right)^{\frac{1}{2}} \tag{269}$$

It is, however not obvious that $|||u|||_{\text{sip}}$ actually is a norm.

**Remarks:** we point out some connections between norms, bilinear forms and matrices:

- definiteness can also be defined for bilinear forms: $a(-,-)$ is positiv definite $\Leftrightarrow \forall u, u \neq 0 : a(u, u) > 0$

- a positive definite bilinear form induces a norm: $|||u|||_a := \sqrt{a(u,v)}$
- in a finite dimensional space, a positive definite bilinear form can be related to a positive definite matrix:

$$u = \sum_{j,n} \tilde{u}_{j,n}\Phi_{j,n},\ v = \sum_{i,m} \tilde{v}_{i,m}\Phi_{i,m},\ \underline{\underline{M}}_{(i,m)(j,n)} = a\left(\Phi_{j,n},\ \Phi_{i,m}\right) \tag{270}$$

- a matrix $\underline{\underline{M}}$ is positive definite, if, and only if for all $\underline{x} \neq 0$ we have $\underline{x}^T \underline{\underline{M}}\underline{x} \geq 0$.
- for a *symmetric* matrix $\underline{\underline{M}}$, this is equivalent to that all eigenvalues are $> 0$
- for a un-symmetric matrix, the spectrum gives no information about definiteness, e.g. $\underline{\underline{M}} = \left(\begin{smallmatrix} 1 & -3 \\ 0 & 1 \end{smallmatrix}\right)$ has the double eigenvalue 1, but $[1\ 1]\underline{\underline{M}}\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right] = -1$.
- if $\underline{\underline{M}} \in \mathbb{R}^{L\times L}$ also induces a bilinear form on $\mathbb{R}^L$, $a_{\underline{\underline{M}}}(\underline{x},\underline{y}) = \underline{x}\cdot\underline{\underline{M}}\cdot\underline{y}$ and if $\underline{\underline{M}}$ is positive definite, $a_{\underline{\underline{M}}}(-,-)$ is positive definite: In this case $\underline{\underline{M}}$ also induces a norm $|||\underline{x}|||_{\underline{\underline{M}}} = \sqrt{\underline{x}^T\underline{\underline{M}}\underline{x}}$

## 6.5 Symmetric Interior Penalty (SIP)

One possibility for a Discretization that is *stable* and *consistent*. (Alternatives: Local non-symmetric interior penalty.)

$$a_{\text{sip}}(u,v) = \int_\Omega \underbrace{\nabla u\cdot\nabla v}_{\text{Volume term}}\ dV - \oint_{\Gamma\setminus\Gamma_N} \underbrace{\{\nabla u\}\cdot n_\Gamma \llbracket v\rrbracket}_{\text{consistency term}} + \underbrace{\{\nabla v\}\cdot \underline{n}_\Gamma \llbracket u\rrbracket}_{\text{symmetry term}}\ dA + \oint_{\Gamma\setminus\Gamma_N} \underbrace{\eta \llbracket u\rrbracket \llbracket v\rrbracket}_{\text{penalty term}}\ dA \tag{271}$$

- penalty term prevents spurious solutions
- penalty factor $\eta(x)$ must be chosen large enough, otherwise method is unstable; in each cell, one might pick

$$\eta = c\cdot p^2 \frac{|\partial\mathcal{K}|}{|\mathcal{K}|}, c \approx 1 \tag{272}$$

  and on $\Gamma$

$$\eta = \max(\eta^-,\eta^+)$$

- $a_{\text{sip}}(-,-)$ is symmetric: reflects symmetry of $a(-,-)$ and is good for solvers.

**Stability of the SIP-form:** According to the Lax-Milgram-theorem, we have to show coercivity of the SIP-form $a_{\text{sip}}(-,-)$, i.e. we have to show that there is a constant so that

$$a_{\text{sip}}(u,u) \geq \gamma |||u|||^2_{\text{sip}} \quad \text{for all u} \in \mathbb{P}_p(\mathfrak{K}_h h) \tag{273}$$

We see that ( for $\Gamma_N = \emptyset$)

$$a_{sip}(u,u) = \underbrace{\int_\Omega \nabla u\cdot\nabla u\, dV}_{=\|\nabla u\|^2_{L^2(\Omega)}} - \oint_\Gamma \{\nabla u\}\cdot n_\Gamma \llbracket u\rrbracket + \{\nabla u\}\cdot\underline{n}_\Gamma \llbracket u\rrbracket\ dA + \oint_\Gamma \eta \llbracket u\rrbracket^2\ dA \tag{274}$$

and the inequality (273) reduces to

$$\|\nabla u\|^2_{L^2(\Omega)} - 2\oint_\Gamma \{\nabla u\}\cdot\underline{n}_\Gamma \llbracket u\rrbracket\ dA + \oint_\Gamma \eta \llbracket u\rrbracket^2\ dA \geq \gamma\|\nabla u\|^2_{L^2(\Omega)} + \gamma\oint \frac{1}{h_l}\llbracket u\rrbracket^2\ dA \tag{275}$$

The critical part is

$$(1-\gamma)\|\nabla u\|^2_{L^2(\Omega)} - 2\oint_\Gamma \{\nabla u\}\cdot\underline{n}\llbracket u\rrbracket\ dA + \oint_\Gamma (\eta - \frac{\gamma}{h_l})\llbracket u\rrbracket^2\ dA \geq 0 \tag{276}$$

The basic idea of the proof (of coercivity of the SIP-norm) is to show that if $\eta$ is large enough, the above inequality is fulfilled for every $u$ in the DG-space. This is sometimes called: "obtaining control over the gradients by choosing a sufficiently large penalty". The actual proof is rather technical and will be skipped, see e.g. Hillewaert (2013) or Di Pietro and Ern (2012). Its foundation, however, are the so-called inverse trace inequalities.

**Theorem (inverse trace inequalities)** The surface integral of a polynomial $v$ over $\partial\mathcal{K}$ can be bound by its volume integral, i.e.

$$\oint_{\partial\mathcal{K}} v^2\, dA \leq \eta \int_{\mathcal{K}} v^2\, dV \quad \forall v \in \mathbb{P}_p(\{\mathcal{K}\}) \tag{277}$$

where the constant $\eta$ behaves as

$$\eta = \frac{p^2}{h} c(\mathcal{K}) \tag{278}$$

The factor $c(\mathcal{K})$ depends on the shape of cell $\mathcal{K}$, and $h$ is the diameter of $\mathcal{K}$. For an overview of shape factors for different type of elements, see Hillewaert (2013).

## 6.6 Implementation of implicit methods

From the generic variational formulation (264), resp. from the SIP-form (271) we can find matrix $\underline{\underline{M}}_{\text{sip}}$ of the Poisson system. Remember that the trial function $u$ and the test function $v$ can be represented as

$$u = \sum_{j=0}^{J-1} \sum_{n=0}^{N_p-1} \Phi_{j,n} \tilde{u}_{j,n} \quad \text{and } v = \sum_{j=0}^{J-1} \sum_{n=0}^{N_p-1} \Phi_{j,n} \tilde{v}_{j,n} \tag{279}$$

Since each $v \in \mathbb{P}(K_h)$ be represented by the finite basis $\underline{\Phi}$, (264) is equivalent to: find $u \in \mathbb{P}_P(K_h)$ so, that

$$a\left(\sum_{j,n} \Phi_{j,n} \tilde{u}_{j,n}, \Phi_{i,m}\right) = \int g_\Omega \Phi_{i,m}\, dV \text{ for } 0 \leq i < J,\ 0 \leq m < N_p \tag{280}$$

Considering that $a_{\text{sip}}(-,-)$ is linear, we can write the system

$$\underbrace{\begin{bmatrix} a_{\text{sip}}(\Phi_{0,0},\Phi_{0,0}) & \dots & a_{\text{sip}}(\Phi_{J-1,N_{P-1}},\Phi_{0,0}) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ a_{\text{sip}}(\Phi_{0,0},\Phi_{J-1,N_{P-1}}) & \dots & a_{\text{sip}}(\Phi_{J-1,N_{P-1}},\Phi_{J-1,N_{P-1}}) \end{bmatrix}}_{=:\underline{\underline{M}}_{\text{sip}}} \cdot \underbrace{\begin{bmatrix} \tilde{u}_{0,0} \\ \vdots \\ \vdots \\ \tilde{u}_{J-1,N_p-1} \end{bmatrix}}_{=:\tilde{u}} = \underbrace{\begin{bmatrix} \int g_\Omega \cdot \Phi_{0,0}\, dV \\ \vdots \\ \vdots \\ \int g_\Omega \cdot \Phi_{J-1,N_p-1}\, dV \end{bmatrix}}_{=:\tilde{g}} \tag{281}$$

Now we define a multi-index notation: for numbers $j, n$ we write

$$(j\ n) := jN_p + n \tag{282}$$

It maps all combinations of $j$ and $n$, $(0 \leq j \leq J-1,\ 0 \leq n \leq N_p - 1)$ to indices in the range of 0 to $JN_p - 1$. Then we can write

$$\underline{\underline{M}}_{\text{sip},(i,m)(j,n)} = a_{\text{sip}}(\Phi_{j,n}, \Phi_{i,m}) \tag{283}$$

$$\tilde{u}_{(j,n)} = \tilde{u}_{j,n} \tag{284}$$

$$\tilde{g}_{(i,m)} = \int_\Omega g_\Omega \Phi_{i,m}\, dV \tag{285}$$

From a mathematical point-of-view, there are many alternatives to the multi-index definition (282) , since it just needs to be a bijective mapping between pairs $j, n$ and the numbers $0, \dots, JN_p - 1$. From an implementation point-of-view, however it is better to have a *fast rotation* polynomial index and a *slow rotation* cell-index, since this choice yields a block structure of $\underline{\underline{M}}_{\text{sip}}$, which is beneficial for performance.

To implement a DG-method efficiently, it must be possible to write this in the form

$$a_{\text{sip}}(u,v) = a^{\text{Vol}}(u,v) + a^{\text{Edg}}(u,v) \tag{286}$$

$$a^{\text{Vol}}(u,v) = \sum_j \int_{\mathcal{K}_j} f^{\text{Vol}}(\underline{x}, u, v, \nabla_h u, \nabla_h v)\, dV \tag{287}$$

$$a^{\text{Edg}}(u,v) = \sum_{E \subseteq \Gamma} \oint_E f^{\text{Edg}}(\underline{x}, \underline{n}_\Gamma, u^-, u^+, v^-, v^+, \nabla_h u^-, \nabla_h u^+, \nabla v^-, \nabla v^+)\, dV \tag{288}$$

Therefore, we can split the matrix into a volume- and edge-part, i.e.

$$\underline{\underline{M}} = \underline{\underline{M}}^{\mathrm{Vol}} + \underline{\underline{M}}^{\mathrm{Edg}}, \tag{289}$$

with

$$\underline{\underline{M}}^{\mathrm{Vol}}_{(i,m)(j,n)} = a^{\mathrm{Vol}}(\Phi_{j,n}, \Phi_{i,m}), \tag{290}$$

$$\underline{\underline{M}}^{\mathrm{Edg}}_{(i,m)(j,n)} = a^{\mathrm{Edg}}(\Phi_{j,n}, \Phi_{i,m}). \tag{291}$$

By an *efficient* implementation, we mean that the computational cost for computing the matrix $M$ scales linear with the number of cells $J$. This is only possible if we can exploit the *locality* of the DG method.

**Assembly of the volume part:** to note that *off-diagonal-blocks* of $M$ are zero, i.e.

$$a^{\mathrm{Vol}}(\Phi_{j,n}, \Phi_{i,m}) = 0 \quad \text{for } i \neq j \tag{292}$$

$$\underline{\underline{M}}_{(j,-),(i,-)} = 0 \quad \text{for } i \neq j \tag{293}$$

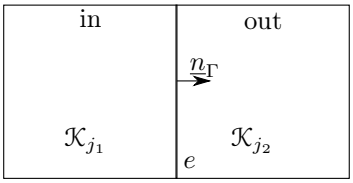
**Assembly of the edge part** E.g. consider the edge term

$$a^{\mathrm{Edg}}(u,v) = \oint_{\Gamma} [\![u]\!]\,[\![v]\!]\,dS =$$

$$= \oint_{\Gamma^{\mathrm{Int}}} (u^{\text{-}} - u^{+})(v^{\text{-}} - v^{+}) dS =$$

$$= \underbrace{\sum_{e \subset \Gamma} \oint_{e} u^{\text{-}}v^{\text{-}} - u^{\text{-}}v^{+} - u^{+}v^{\text{-}} + u^{+}v^{+}\, dV}_{=: a_e^{\mathrm{Edg}}(\ldots)}. \tag{294}$$

We consider one edge $e$ which is located between cells $\mathcal{K}_{j_1}$ and $\mathcal{K}_{j_2}$:

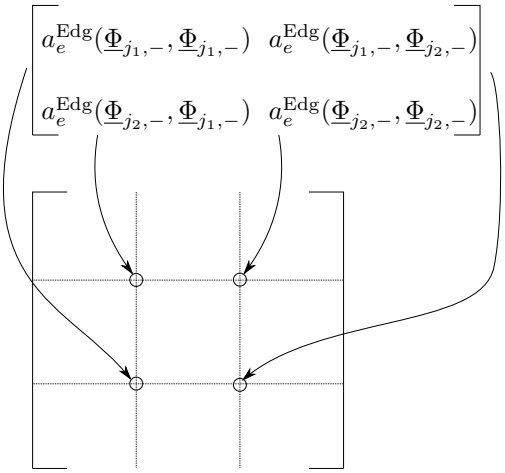



For the trial- and the test-functions $u$ and $v$, we use the basis functions $\Phi_{j,n}$, obviously for $a^{\mathrm{Edg}}(\Phi_{l_1,n}, \Phi_{l_2,m}) \neq 0$ we must fulfill ($l_1 = j_1$ or $l_1 = j_2$) and ($l_2 = j_1$ or $l_2 = j_2$). Therefore, the integration over $e$ produces 4 blocks

$$M^{\mathrm{Edg}}_{(j_1,n),(j_1,m)} += a_e^{\mathrm{Edg}}(\Phi_{j_1,n}, \Phi_{j_1,m}) \quad M^{\mathrm{Edg}}_{(j_1,n),(j_2,m)} += a_e^{\mathrm{Edg}}(\Phi_{j_1,n}, \Phi_{j_2,m}) \tag{295}$$

$$M^{\mathrm{Edg}}_{(j_2,n),(j_1,m)} += a_e^{\mathrm{Edg}}(\Phi_{j_2,n}, \Phi_{j_1,m}) \quad M^{\mathrm{Edg}}_{(j_2,n),(j_2,m)} += a_e^{\mathrm{Edg}}(\Phi_{j_2,n}, \Phi_{j_2,m}) \tag{296}$$

For an edge $e$:




### 6.7 The heat equation

For the heat equation we search an $u(t,-) : \mathbb{R}_{>0} \to \mathbb{P}_p(\mathcal{K})$ so that

$$\int_\Omega \partial_t u \; v \, dV + a_{\mathrm{sip}}(u, v) = 0 \; \forall \; v \in \mathbb{P}_p(\mathcal{K}) \tag{297}$$

using that $u(t,x) = \sum_{j,n} \tilde{u}_{j,n}(t) \cdot \Phi_{j,n}(\underline{x})$ and using a test function $\Phi_{i,m}$, we get

$$\sum_{j,n} (\partial_t \tilde{u}_{j,n}) \int_\Omega \Phi_{j,n} \Phi_{i,m} \, dV + \sum_{j,m} a_{\mathrm{sip}} \left( \Phi_{j,n} \Phi_{i,m} \right) \tilde{u}_{j,n} = 0 \tag{298}$$

In matrix notation, using $(\underline{\underline{M}}_{\mathrm{mass}})_{(j,n)(i,m)} := \int_\Omega \Phi_{j,n} \Phi_{i,m} \, dV$ we get

$$\underline{\underline{M}}_{\mathrm{mass}} \partial_t \tilde{u} + \underline{\underline{M}}_{\mathrm{sip}} \tilde{u} = 0 \tag{299}$$

If we assume an orthonormal basis, we have $\underline{\underline{M}}_{\mathrm{mass}} = \mathbf{1}$. For time discretization, implicit methods are preferred:

- implicit Euler, BDF-schemes
- Crank-Nicolson (also called implicit trapezoidal rule)
- implicit Runge-Kutta

## 7 Poisson equation as a system

Obviously, it is also possible to discretize the Poisson equation as a system of first-order-PDE's, introducing a vector field $\underline{\sigma}$:

$$\underline{\sigma} + \nabla u = 0, \quad \text{in } \Omega \tag{300}$$

$$\operatorname{div}(\underline{\sigma}) = g_\Omega, \text{ in } \Omega \tag{301}$$
$$u = g_D, \text{ on } \Gamma_D \tag{302}$$
$$-\underline{\sigma} \cdot \underline{n}_{\partial\Omega} = g_N, \text{ on } \Gamma_N \tag{303}$$

resp. in matrix-notation:

$$\begin{bmatrix} \mathbf{1} & \nabla \\ \operatorname{div} & 0 \end{bmatrix} \cdot \begin{bmatrix} \underline{\sigma} \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ g_\Omega \end{bmatrix} \tag{304}$$

**DG-Discretization of the Poisson system** Multiply (300) with a test function $\underline{\tau}$ and integrating over a $\mathcal{K}_j$

$$\int_{\mathcal{K}_j} \underline{\sigma} \cdot \underline{\tau} \, dV + \int_{\mathcal{K}_j} \nabla u \cdot \underline{\tau} \, dV = 0 \tag{305}$$

Integrating by parts:

$$\int_{\mathcal{K}_j} \underline{\sigma} \cdot \underline{\tau} \, dV + \oint_{\partial\mathcal{K}_j} u \, \underline{n}_{\partial\mathcal{K}_j} \cdot \underline{\tau} \, dA - \int_{\mathcal{K}_j} \operatorname{div}(\underline{\tau}) u \, dV = 0 \tag{306}$$

We replace $u$ in the surface integral with

$$u = \begin{cases} \{u\} & \text{on } \Gamma_\mathrm{i} \\ g_D & \text{on } \Gamma_D \\ \{u\} = \{u^-\} & \text{on } \Gamma_N \end{cases}$$

and sum over all $\mathcal{K}_j$ in the grid

$$\int \underline{\sigma\tau} \, dV + \underbrace{\oint_{\Gamma\setminus\Gamma_D} \{u\} \, [\![\underline{\tau}]\!] \cdot \underline{n}_\Gamma \, dA - \int_\Omega \operatorname{div}(\underline{\tau}) \cdot u \, dV}_{=b(u,\underline{\tau})} = \underbrace{\oint_{\Gamma_D} -g_D \, [\![\underline{\tau}]\!] \cdot \underline{n}_\Gamma \, dA}_{=:r_D(\underline{\tau})} \tag{307}$$

In the same fashion, we derive a weak form for the second equation (301). Multiply by test function $v$, integrate over $\mathcal{K}_j$

$$\int_{\mathcal{K}_j} \operatorname{div} \underline{\sigma} \cdot v \, dV = \int_{\mathcal{K}_j} g_\Omega v \, dV \tag{308}$$

... apply integration by parts, use a central-difference flux

$$\widehat{\underline{\sigma} \cdot \underline{n}} = \begin{cases} \{\sigma\} \cdot \underline{n} & \text{on } \Gamma_\mathrm{i} \\ \sigma^- \cdot \underline{n} & \text{on } \Gamma_D \\ g_N & \text{on } \Gamma_N \end{cases}$$

and sum over all cells

$$\underbrace{\oint_{\Gamma\setminus\Gamma_N} \{\underline{\sigma} \cdot \underline{n}_\Gamma\} \, [\![v]\!] \, dA - \int_\Omega \underline{\sigma} \nabla v \, dV}_{=:c(\underline{\sigma},v)} = \int_\Omega g_\Omega v \, dV + \underbrace{\oint_{\Gamma_N} -g_N v \, dA}_{=:r_N(v)} \tag{309}$$

To discretize $u$ and $\underline{\sigma}$, we use DG-spaces of order $k$ and $p$. Finally we arrive at the following DG-discretization: Find $\underline{\sigma} \in \mathbb{P}_p(\mathcal{K})^D$ and $u \in \mathbb{P}_k(\mathcal{K})$ so that

$$\int \underline{\sigma} \cdot \underline{\tau} \, dV + b(u, \underline{\tau}) = r_D(\underline{\tau}) \qquad \forall \underline{\tau} \in \mathbb{P}_p(\mathcal{K})^D \tag{310}$$

$$c(\underline{\sigma}, v) = \int_\Omega g_\Omega v \, dV + r_N(v) \quad \forall v \in \mathbb{P}_k(\mathcal{K}) \tag{311}$$

**Symmetry of the formulation** (310)**-**(311)**:** One can indeed show that $b(u, \underline{\tau}) = -c(\underline{\tau}, u)$

$$\begin{aligned} b(u, \underline{\tau}) = &\oint_{\Gamma\setminus\Gamma_D} \{u\} \, [\![\underline{\tau}]\!] \cdot \underline{n}_\Gamma \, dA - \int_\Omega \operatorname{div}(\underline{\tau}) \cdot \underline{u} \, dV \\ &= \sum_j \left( \oint_{\partial\mathcal{K}_j} \{u\} \cdot \underline{\tau}^- \cdot \underline{n}_{\partial\mathcal{K}_j} \, dA - \int_{\mathcal{K}_j} \operatorname{div}(\underline{\tau}) u \, dV \right) \end{aligned}$$

$$= \sum_j \left( \oint_{\partial \mathcal{K}_j} \frac{1}{2} \left(u^- + u^+\right) \underline{\tau}^- \cdot \underline{n}_{\partial \mathcal{K}} \, dA - \left( \oint_{\partial \mathcal{K}_j} \underline{\tau}^- \cdot \underline{n}_{\partial \mathcal{K}} \cdot u^- \, dA - \int_{\mathcal{K}_j} \underline{\tau} \cdot \nabla u \, dV \right) \right)$$
$$= -\sum_j \left( \oint_{\partial \mathcal{K}_j} \frac{1}{2} \left(u^- - u^+\right) \underline{\tau}^- \cdot \underline{n}_{\partial \mathcal{K}} \, dA + \int_{\mathcal{K}_j} \underline{\tau} \cdot \nabla u \, dV \right) = -c(\underline{\tau}, u) \tag{312}$$

The last step is a consequence of the identity

$$\frac{-1}{2} \underbrace{\left(u^1 - u^2\right)}_{=[\![u]\!]} \underline{\tau}^1 \cdot \underbrace{\underline{n}_\Gamma}_{\underline{n}_{\partial \mathcal{K}_1}} + \frac{-1}{2} \underbrace{\left(u^2 - u^1\right)}_{-[\![u]\!]} \underline{\tau}^2 \cdot \underbrace{\left(-\underline{n}_\Gamma\right)}_{=\underline{n}_{\partial \mathcal{K}_2}} = \frac{-1}{2} [\![u]\!] \, \underline{n}_\Gamma \cdot \left(\tau^1 + \tau^2\right) = - [\![u]\!] \, \underline{u}_\Gamma \cdot \{\underline{\tau}\} \tag{313}$$

for some point on $\Gamma_i$, bounding to cells $\mathcal{K}_1$ and $\mathcal{K}_2$, $\left(\underline{n}_{\partial \mathcal{K}_2} = -\underline{n}_\Gamma, \underline{n}_{\partial \mathcal{K}_1} = \underline{n}_\Gamma\right)$ with respect to $\mathcal{K}_1$ $u^- = u^1$ and $u^+ = u^2$, with respect to $\mathcal{K}_2$ $u^- = u^1$ and $u^+ = u^2$.
Using this identity, one can sum equations (300) and (311), to obtain the following representation: Find $(\underline{\sigma}, u) \in \mathbb{P}_p(\mathcal{K})^D \times \mathbb{P}_k(\mathcal{K})$ so that

$$\underbrace{\int_\Omega \underline{\sigma} \cdot \underline{\tau} \, dV + b(u, \underline{\tau}) - b(v, \underline{\sigma})}_{=:B((\underline{\sigma},u),(\underline{\tau},v))} = \int_\Omega g_\Omega v \, dV + r_D(\underline{\tau}) + r_N(v) \qquad \forall (\underline{\tau}, v) \in \mathbb{P}_p(\mathcal{K})^D \times \mathbb{P}_k(\mathcal{K}) \tag{314}$$

## 7.1 Stability of the system-formulation

Unfortunately, the form $B(-,-)$ is not coercive; it is only partially coercive:

$$B\left((\underline{\sigma}, v), (\underline{\sigma}, v)\right) = \int_\Omega \underline{\sigma} \cdot \underline{\sigma} \, dV + b(v, \underline{\sigma}) - b(v, \underline{\sigma}) = \|\underline{\sigma}\|^2_{L^2(\Omega)} \tag{315}$$

It is obvious that there exists a coercivity constant $\gamma > 0$ so that

$$B\left((\underline{\sigma}, v), (\underline{\sigma}, v)\right) \geq \gamma \left\|\underline{\sigma}\right\|^2 \tag{316}$$

However, we fail to realize

$$B\left((\underline{\sigma}, v), (\underline{\sigma}, v)\right) \geq \gamma |||(\underline{\sigma}, v)|||^2_*, \tag{317}$$

since e.g. for any $v$ with $|||(0, v)|||_* > 0$

$$B\left((0, v), (0, v)\right) = 0 \leq \gamma |||(0, v)|||^2_* \tag{318}$$

Therefore, the Lax-Milgram-theorem does not apply. This does not necessarily mean that the Problem is ill-posed. It is, however still possible to prove that the formulation (314) coercivity resp. is a sufficient condition, but it is a necessary condition.

**Theorem: Banach-Nečas-Babuška (BNB), inf-sup-condition.** The variational problem:

$$a(u, v) = \int f v \, dV \qquad \forall v$$

has a unique solution (is well-posed) *if, and only if* such that

1. there exists a constant $C_{\text{Sta}} > 0$ so that for all $v$

$$C_{\text{Sta}} |||v|||_* \leq \sup_{w \neq 0} \frac{a(v, w)}{|||w|||_*} \tag{319}$$

2. if $a(v, w) = 0 \; \forall v$, this implies that $w = 0$.

Since the condition 319 can be reformulated as

$$C_{\text{Sta}} \leq \inf_{v \neq 0} \sup_{w \neq 0} \frac{a(v, w)}{|||v|||_* |||w|||_*} \tag{320}$$

this is also called *inf-sup* condition.

## 7.2 The inf-sup-condition for a general saddle-point problem

A complete proof of the inf-sup-stability of the Poisson problem (314), or the Stokes-problem (361) is beyond the scope of this lecture. However, we give the sketch of the proof for the generic saddle-point problem: find $(\underline{u}, \psi)$ so that

$$a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) = f_1(\underline{v}) \quad \forall \underline{v} \tag{321}$$

$$-b(q, \underline{u}) = f_2(q) \quad \forall q \tag{322}$$

$$\tag{323}$$

resp.

$$B((\underline{u}, p), (\underline{v}, q)) = f_1(\underline{v}) + f_2(q) \quad \forall (\underline{v}, q) \tag{324}$$

where

$$B((\underline{u}, p), (\underline{v}, q)) = a(\underline{u}, \underline{v}) + b(p, \underline{v}) - b(q, \underline{u}). \tag{325}$$

To show inf-sup-stability, we have to proof that there exists a constant $\gamma > 0$ so that

$$\gamma |||(\underline{v}, q)|||_{\text{sdl}} \leq \sup_{(\underline{w}, r) \neq 0} \frac{B((\underline{v}, q)(\underline{w}, r))}{|||(\underline{v}, q)|||_{\text{sdl}}} \quad \forall (\underline{v}, q) \tag{326}$$

in some suitable norm $||| - |||_{\text{sdl}}$.

**Remarks(saddle-point problems).** Systems like (321), (322), if discretized lead to matrices of the form

$$\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix} \tag{327}$$

Such matrices are very often called saddle-point matrices. The actual definition of a saddle-point matrix is that it has Eigenvalues and negative sign. For coercive problems, on the other hand, all Eigenvalues have the same sign.

**Assumptions and definitions** .

- Since the saddle-point system is reminiscent of a Stokes-problem, we call variables $\underline{u}, \underline{v}, \underline{w}$ *velocities*, which are members of a *velocity space*. Analogously, the pressure-type variables $\psi$, $q$, $r$ are members of a *pressure space*. For the coarse sketch of the proof, it is not necessary to define these spaces.
- $a(-,-)$ is symmetric and positive definite, thereby it is coercive
- $a(-,-)$ thereby induces the velocity-norm $|||\underline{u}|||_a := \sqrt{a(\underline{u}, \underline{u})}$ on the velocity-space
- We also assume a suitable pressure norm $||| - |||_p$ which we do not specify in detail, on the pressure space
- We choose the norm $|||(\underline{u}, \psi)|||_{\text{sdl}} = (|||u|||_a^2 + |||\psi|||_p^2)^{1/2}$ The most critical part is the stability of the pressure. We assume that there exists an estimate

  $$\|q\|_p \lesssim \sup_{\underline{w} \neq 0} \frac{b(q, \underline{w})}{|||\underline{w}|||_a} \quad \forall q \tag{328}$$

  Showing such an estimate usually is the most difficult part of the proof. It heavily depends on the choice for velocity and pressure space and on the form $a(-,-)$.
- For ease of notification, we make use of the *approximate less*-notification: e.g.

  $$X \lesssim Z \tag{329}$$

  means that there exists a constant $\gamma \geq 0$, independent of the expressions $X$ and $Z$, so that

  $$\gamma X \lesssim Z. \tag{330}$$

  This allows us to avoid the definition of e.g. coercivity constants.

- Recall Young's inequality: For two numbers $a$, $b$ we have

$$ab \leq \frac{a^2}{2} + \frac{b^2}{2}. \tag{331}$$

Using the $\lesssim$-notation, one e.g. argues that

$$ab + b^2 \leq \frac{a^2}{2} + \frac{b^2}{2} + b^2 \lesssim a^2 + b^2. \tag{332}$$

- For the Poisson System (314)

$$a(\underline{u}, \underline{v}) = \int_\Omega \underline{u} \cdot \underline{v} \, dV \tag{333}$$

and thus

$$|||\underline{u}|||_a = |||\underline{u}|||_{L^2(\Omega)} \tag{334}$$

- We further assume *boundness* of a $a(-,-)$, i.e.

$$\sup_{\underline{w} \neq 0} \frac{a(v,w)}{|||w|||_a} \lesssim |||v|||_a \tag{335}$$

Using the partial coercivity (316), one obtains

$$|||v|||_a^2 = B((\underline{v}, q)(\underline{v}, q) \lesssim \underbrace{\sup_{(\underline{w}, r) \neq 0} \frac{B((\underline{v}, q)(\underline{w}, r))}{|||(\underline{w}, r)|||_{\text{sdl}}}}_{=:S} |||(\underline{v}, q)|||_{\text{sdl}} \tag{336}$$

In the pressure stability assumption (328), we substitute $b(q, \underline{w}) = B((\underline{v}, q), (\underline{w}, 0)) - a(\underline{v}, \underline{w})$ and obtain

$$|||q|||_p \lesssim \sup_{\underline{w} \neq 0} \left[ \frac{-a(\underline{v}, \underline{w})}{|||w|||_a} + \frac{B_h((\underline{v}, q)(\underline{w}, 0))}{\underbrace{|||w|||_a}_{=|||(\underline{w},0)|||_{\text{sdl}}}} \right] \lesssim \underbrace{\sup_{(\underline{w} \neq 0} \frac{a(\underline{v}, \underline{w})}{|||w|||_a}}_{\lesssim |||v|||_a \text{ due to bound (335)}} + S \tag{337}$$

We take the square of this estimate and use Young's inequality to obtain:

$$|||q|||_p^2 \lesssim |||v|||_a^2 + 2|||v|||_a S + S^2 \lesssim |||v|||_a^2 + S^2 \tag{338}$$

By adding (336) and (338), we obtain

$$|||v|||_a^2 + |||q|||_p^2 \lesssim \underbrace{S|||(\underline{v}, q)|||_{\text{sdl}}}_{\lesssim S^2 + |||(\underline{v},q)|||_{\text{sdl}}} + |||v|||_a^2 + S^2 \tag{339}$$

and finally ( due to the $\lesssim$-notation, we can substract $|||v|||_a^2$ form the right-hand-side: $\alpha|||v|||_a + ... \leq |||v_a||| + .. \Leftrightarrow (\alpha - 1)|||v|||_a + ... \leq ...$) and finally obtain the inf-sup-condition:

$$|||(\underline{v}, q)|||_{\text{sdl}}^2 \lesssim S^2 \tag{340}$$

**Final remarks** The proof can be applied to show the well-posedness of the weak continuous problem or of the discrete problem. If the velocity and pressure space are DG, the proof might become more complex, since the pressure stability estimate (328) contains additional terms.

# 8 Incompressible flows

## 8.1 The continuous setting

**Equations and boundary conditions, strong form:** Steady Stokes, note similarity to the Poisson equation (300)-(301):

$$-\frac{1}{\text{Re}} \Delta \underline{u} + \nabla \psi \ = \underline{g}_\Omega \tag{341}$$

$$\operatorname{div}(\underline{u}) \qquad = 0 \tag{342}$$

Steady Navier Stokes:

$$-\frac{1}{\mathrm{Re}}\Delta\underline{u} + \operatorname{div}(\underline{u}\otimes\underline{u}) + \nabla\psi \;= \underline{g}_\Omega \tag{343}$$

$$\operatorname{div}(\underline{u}) \qquad = 0 \tag{344}$$

Instationary Navier-Stokes:

$$\partial_t\ \underline{u} - \frac{1}{\mathrm{Re}}\Delta u + \operatorname{div}(\underline{u}\otimes\underline{u}) + \nabla\psi \;= \underline{g}_\Omega \tag{345}$$

$$\operatorname{div}(\underline{u}) \qquad = 0 \tag{346}$$

Boundary conditions:

$$\underline{u} = \underline{u}_D \quad \text{on } \Gamma_D \quad \text{(Dirichlet)} \tag{347}$$

$$\left(-\frac{1}{\mathrm{Re}}\ \nabla\underline{u} + \mathbf{1}_p\psi\right)\underline{n}_{\partial\Omega} = 0 \qquad \text{on } \Gamma_N \quad \text{(Neumann)} \tag{348}$$

If we have only Dirichlet boundary conditions, the pressure $\psi$ is only unique up to a constant value, i.e. if $(\underline{u},\psi)$ is a solution for the Stokes/ Navier-Stokes equation, so is $(\underline{u},\psi+\text{const.})$. An additional condition is required to fix the pressure, e.g. $p(\underline{x}_0)=0$ or $\int_\Omega p\,dV = 0$.

**Weak form of the Stokes equation:** Start with one component of equation (341)

$$\operatorname{div}\left(-\frac{1}{\mathrm{Re}}\nabla u_d + \underline{e}_d\ \psi\ \right) = g_{\Omega,d} \tag{349}$$

multiply with test function $v_d$ and integrate over $\Omega$:

$$\int_\Omega \operatorname{div}\left(-\frac{1}{\mathrm{Re}}\nabla u_d + \underline{e}_d\ \psi\ \right) v_d\,dV = \int_\Omega g_d v_d\,dV, \tag{350}$$

perform integration by parts and assume that

$$v_d = 0 \text{ on } \Gamma_D, \tag{351}$$

$$-\frac{1}{\mathrm{Re}}\nabla u_d + \underline{e}_d\psi = 0 \text{ on } \Gamma_N \tag{352}$$

obtain

$$\underbrace{\oint_{\partial\Omega}\left(-\frac{1}{\mathrm{Re}}\nabla u_d + \underline{e}_d\psi\right)\cdot\underline{n}_{\partial\Omega}v_d dA}_{=0 \text{ due to boundary conditions}} - \int_\Omega\left(-\frac{1}{\mathrm{Re}}\nabla u_d + \underline{e}_d\psi\right)\cdot\nabla v_d\,dV = \int_\Omega g_{\Omega,d}v_d\,dV \tag{353}$$

take the sum over $d$:

$$\underbrace{\int_\Omega \frac{1}{\mathrm{Re}}\nabla\underline{u} : \nabla\underline{v}\,dV}_{=:a(\underline{u},\underline{v})} + \underbrace{\int_\Omega -\psi\operatorname{div}(\underline{v})\,dV}_{=:b(\psi,\underline{v})} = \int \underline{g}_\Omega\cdot\underline{v}\,dV \tag{354}$$

We used the following notations: $0\le d\le D-1$, $\underline{u}=(u_0,u_1)$ for $D=2$ resp. $\underline{u}=(u_0,u_1,u_2)$ for $D=3$, analog for $v_d$ resp. $\underline{v}$ for $D=2$, we further have

$$\nabla\underline{u} = \begin{bmatrix}\nabla u_0\\ \nabla u_1\end{bmatrix} = \begin{bmatrix}\frac{\partial u_0}{\partial x} & \frac{\partial u_0}{\partial y}\\ \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y}\end{bmatrix}, \tag{355}$$

and

$$\sum_d \nabla u_D\cdot\nabla v_d =: \nabla\underline{u} : \underline{v} = \begin{bmatrix}\nabla u_0\\ \nabla u_1\end{bmatrix} : \begin{bmatrix}\nabla v_0\\ \nabla v_1\end{bmatrix} = \nabla u_0\cdot\nabla v_0 + \nabla u_1\cdot\nabla v_1 \tag{356}$$

For the pressure part,

$$\sum_d (\underline{e}_d \psi) \cdot \nabla v_d = (\psi, 0) \cdot \nabla v_0 + (0, \psi) \cdot \nabla v_1 = \psi \frac{\partial v_0}{\partial x} + \psi \frac{\partial v_1}{\partial y} = \psi \operatorname{div}(\underline{v}) \tag{357}$$

For the continuity equation, (342), we use a test function $w$ and

$$\underbrace{\int_\Omega \operatorname{div}(\underline{u}) \cdot w \, dV = 0}_{=:c(\underline{u},w)} \tag{358}$$

and, by integration by parts and assuming $w|_{\partial\Omega} = 0$, observe the symmetry $c(\underline{u}, \psi) = -b(\psi, \underline{u})$.

Finally, we have the weak form of the continuous setting: find $(\underline{u}, \psi)$ so that

$$a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) = \int \underline{g}_\Omega \cdot \underline{v} \, dV \qquad \forall \ \underline{v} \tag{359}$$

$$-b(w, \underline{u}) = 0 \qquad \forall \ w \tag{360}$$

resp.:

$$\underbrace{a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) - b(w, \underline{u})}_{=:B((\underline{u},\psi),(\underline{v},w))} = \int \underline{g}_\Omega \cdot \underline{v} \, dV \qquad \forall \ (\underline{v}, w) \tag{361}$$

**Boundary conditions for the weak form:** So far, we have not specified the function spaces for the test functions $(\underline{v}, w)$ and the solution $(\underline{u}, \psi)$. Furthermore, we have assumed that the boundary integrals vanish, see (353), i.e.

$$\oint_{\partial\Omega} \left( \left( \frac{-1}{\mathrm{Re}} \nabla \underline{u} + \mathbf{1}\psi \right) \cdot \underline{n}_{\partial\Omega} \right) \cdot \underline{v} \, dA = 0 \tag{362}$$

This is indeed justified, at least for homogeneous boundary conditions, i.e. $\underline{u}_D = 0$.

- on $\Gamma_D$, a homogeneous boundary condition is enforced via the function space, we admit only functions with $u|_{\Gamma_D} = 0$. Using a function space,

  $$\mathbb{H}^1_0(\Omega) := \{u; \|u\|_2 + \|\nabla u\|_2 < \infty; \ u|_{\Gamma_D} = 0\} \tag{363}$$

  and pick test function $\underline{v}$ and trial function $\underline{u}$ form that space, i.e. $\underline{u}, \underline{v} \in \left(\mathbb{H}^1_0(\Omega)\right)^D$ ensures that

  $$\oint_{\partial\Omega} \left( \left( \frac{-1}{\mathrm{Re}} \nabla \underline{u} + \mathbf{1}\psi \right) \cdot \underline{n}_{\partial\Omega} \right) \cdot \underline{v} \, dV = 0 \tag{364}$$

- on $\Gamma_N$, we must have $\left(\frac{-1}{\mathrm{Re}} \nabla \underline{u} + \mathbf{1}\psi\right) \cdot \underline{n}_{\partial\Omega} = 0$ and admit that $\underline{v}|_{\Gamma_N} \neq 0$, otherwise the boundary condition cannot be satisfied. If we would admit only functions with $\underline{v}|_{\Gamma_N} = 0$, there is no control over the value of $\left(\frac{-1}{\mathrm{Re}} \nabla \underline{u} + \mathbf{1}\psi\right) \cdot \underline{n}_{\partial\Omega}$ on $\Gamma_N$. I.e., we would have $a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) = 0$ even if the Neumann boundary condition is not fulfilled.

Note that the treatment of Dirichlet boundary conditions is different for the weak form in the continuous setting and for the DG discretization. For the former (and for continuous finite elements), boundary conditions are realized by a restriction to the space of test- and trial-functions. For the latter, boundary conditions are realized via penalty terms.

Inhomogeneous boundary conditions are transfered to the right-hand-side. Instead of solving e.g.

$$\left\{ \begin{array}{rcll} \frac{-1}{\mathrm{Re}} \Delta \underline{u} + \nabla \psi & = & -\underline{g}_\Omega & \text{in } \Omega \\ \underline{u} & = & \underline{u}_D & \text{on } \Gamma_D \end{array} \right.$$

one defines a continuation of $\underline{u}_D$, which is defined on $\Omega$ (not just on $\Gamma_D$) and sufficiently smooth and via transformation $\underline{u} = \underline{u}^* + \underline{u}_D$ reduces (8.1) to

$$\left\{ \begin{array}{rcll} \frac{-1}{\mathrm{Re}} \Delta \underline{u}^* + \nabla \psi & = & \underbrace{-\underline{g}_\Omega - \frac{1}{\mathrm{Re}} \Delta \underline{u}_D}_{:=f^*} & \text{in } \Omega \\ \underline{u}^* & = & 0 & \text{on } \Gamma_D \\ & \cdots & & \end{array} \right.$$

since such transformation is always possible,

- in other words; the inhomogeneous boundary condition is transferred to the right-hand-side
- it is sufficient for theoretical investigations to look only on homogeneous problems.

**Weak form of the Navier- Stokes equation** We have to add the convectional part $\operatorname{div}(\underline{u} \otimes \underline{u})$ to equation (359) resp. (361). We define the trilinear form

$$t(\underline{c}, \underline{u}, \underline{v}) := \int_\Omega \operatorname{div}(\underline{c} \otimes \underline{u}) \cdot \underline{v} \, dV \tag{365}$$

(Notation, e.g. in 2D: $\underline{u} \otimes \underline{v} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} \begin{bmatrix} v_0 & v_1 \end{bmatrix} = \begin{bmatrix} u_0 \underline{v}^T \\ u_1 \underline{v}^T \end{bmatrix} = \begin{bmatrix} u_0 v_0 & u_0 v_1 \\ u_1 v_0 & u_1 v_1 \end{bmatrix}$, $\operatorname{div} \begin{bmatrix} \underline{v}_0^T \\ \underline{v}_1^T \end{bmatrix} = \begin{bmatrix} \operatorname{div}(\underline{v}_0) \\ \operatorname{div}(\underline{v}_1) \end{bmatrix}$) Then, the weak form of the equation ((359) - (360)) is

$$\begin{aligned} t(\underline{u}, \underline{u}, \underline{v}) + a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) &= \int_\Omega \underline{f} \cdot \underline{v} \, dV & (366) \\ -b(w, \underline{u}) &= 0 & (367) \end{aligned}$$

resp.

$$\underbrace{t(\underline{u}, \underline{u}, \underline{v}) + B((\underline{u}, \psi), (\underline{v}, w))}_{=:\mathrm{Ns}((\underline{u},\psi),(\underline{v},w))} = \int_\Omega \underline{f} \underline{v} \, dV \quad \forall (\underline{v}, w) \tag{368}$$

**Weak form of the Instationary Navier-Stokes equation**

$$\begin{aligned} (\partial_t \underline{u}, \underline{v}) + t(\underline{u}, \underline{u}, \underline{v}) + a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) &= \textstyle\int_\Omega \underline{g}_\Omega \cdot \underline{v} \, dV \qquad \forall \ \underline{v} & (369) \\ -b(w, \underline{u}) &= 0 \qquad \forall \ w & (370) \end{aligned}$$

resp.

$$(\partial_t \underline{u}, \underline{v}) + \mathrm{Ns}((\underline{u}, \psi), (\underline{v}, w)) = \int_\Omega \underline{g}_\Omega \cdot \underline{v} \, dV \quad \forall (\underline{v}, w) \tag{371}$$

**Energy conservation**

- One can show that the incompressible Navier-Stokes-equation are *dissipative*: if there is no in-flow, and no additional forcing, the total kinetic energy in the system always decreases; the kinetic energy is conserved, if we omit the diffusion term $\frac{-1}{\mathrm{Re}} \Delta \underline{u}$.
- Total kinetic energy:

  $$E(\underline{u}) := \frac{1}{2} \int_\Omega \underline{u} \cdot \underline{u} \, dV \tag{372}$$

  (we assume: $\Omega$ is either infinite, i.e. $\Omega = \mathbb{R}^D$, or a closed box, i.e. $\underline{u} \cdot \underline{n}_{\partial\Omega} = 0$ on $\partial\Omega$)
- we show the *conservation* resp. the *dissipation of energy*,

  $$\partial_t E(\underline{u}) = -a(\underline{u}, \underline{u}) \leq 0 \tag{373}$$

  if the trilinear form $t(-,-,-)$ is *skew-symmetric*:

  $$\forall \ \underline{v}, \underline{u} \ \forall \ \underline{w} \text{ with } \operatorname{div}(\underline{w}) = 0 : \ t(\underline{w}, \underline{u}, \underline{v}) = -t(\underline{w}, \underline{v}, \underline{u}). \tag{374}$$

  Proof:

  $$\begin{aligned} \partial_t E(\underline{u}) = \partial_t \left( \frac{1}{2} \int_\Omega \underline{u} \cdot \underline{u} \, dV \right) = (\partial_t \underline{u}, \underline{u}) = \\ = \underbrace{-t(\underline{u}, \underline{u}, \underline{u})}_{=0 \text{ (skew symm.)}} - \underbrace{b(\psi, \underline{u})}_{=0 \text{ (continuity eq.)}} - \underbrace{a(\underline{u}, \underline{u})}_{\geq 0 \text{ (positive definiteness}} = \\ = -a(\underline{u}, \underline{u}) \leq 0. \end{aligned} \tag{375}$$

  since $t(\underline{w}, \underline{u}, \underline{v}) = -t(\underline{w}, \underline{v}, \underline{u})$ implies $t(\underline{w}, \underline{u}, \underline{u}) = 0$.

- We have to show that the skew-symmetry of the trilinear form $t(-,-,-)$, i.e. $t(\underline{c},\underline{u},\underline{w}) = 0$ for $\mathrm{div}(\underline{c}) = 0$; First we use the product rule to see

$$\mathrm{div}(\underline{c}\otimes\underline{u})\cdot\underline{v} = (\underline{uv})\underbrace{\mathrm{div}(\underline{c})}_{=0} + ((\underline{c}\cdot\nabla)\underline{u})\cdot\underline{v} \tag{376}$$

Since, in semi-component notation we see that

$$\mathrm{div}(\underline{c}\otimes\underline{u})\cdot\underline{v} = \sum_d \mathrm{div}(\underline{c}u_d)v_d = \sum_d (\mathrm{div}(\underline{c})u_d v_d + (\underline{c}\cdot\nabla u_d)v_d) \tag{377}$$

Furthermore, we see that

$$((\underline{c}\cdot\nabla)\underline{u})\cdot\underline{v} = \mathrm{div}((\underline{u}\cdot\underline{c})\underline{c}) - (\underline{u}\cdot\underline{v})\,\mathrm{div}(\underline{c}) - ((\underline{c}\cdot\nabla)\underline{v})\cdot\underline{u} \tag{378}$$

since

$$\begin{aligned}\sum_i\sum_j (\partial_i(u_j v_j c_i) - (u_j v_j)\partial_i c_i - c_j\partial_j v_i u_i) = \\ = \sum_i\sum_j (\partial_i c_i\cdot(u_j v_j) + c_i\partial_i(u_j v_j) - (u_j v_j)\partial_i c_i - c_j\partial_j v_i u_i) = \\ = \sum_i\sum_j c_i\partial_i u_j v_j + \sum_i\sum_j c_i u_j\partial_i v_j - \sum_{ij} c_j\partial_j v_i u_i\end{aligned} \tag{379}$$

Finally, using all the identities and integration by parts, we get:

$$\begin{aligned}t(\underline{c},\underline{u},\underline{v}) = \int_\Omega \mathrm{div}(\underline{c}\otimes\underline{u})\cdot\underline{v}\,dV = \\ = \int_\Omega (\underline{u}\cdot\underline{v})\underbrace{\mathrm{div}(\underline{c})}_{=0}\,dV + \int_\Omega (\underline{c}\cdot\nabla)\underline{u})\cdot\underline{v}\,dV = \\ = \int_\Omega \mathrm{div}((\underline{u}\cdot\underline{u})\underline{c})\,dV - \int_\Omega (\underline{u}\cdot\underline{v})\underbrace{\mathrm{div}(\underline{c})}_{=0}\,dV - \int_\Omega ((\underline{c}\cdot\nabla)\underline{v})\underline{u}\,dV = \\ = \int_{\partial\Omega} (\underline{u}\cdot\underline{v})\underbrace{\underline{c}\cdot\underline{n}_{\partial\Omega}}_{=0}\,dS - \int_\Omega \mathrm{div}(\underline{c}\otimes\underline{v})\cdot\underline{u}\,dV = \\ = -t(\underline{c},\underline{v},\underline{u})\end{aligned} \tag{380}$$

So,

$$\begin{aligned}t(\underline{c},\underline{v},\underline{u}) &= -t(\underline{c},\underline{v},\underline{u}) &(381)\\ 2t(\underline{c},\underline{v},\underline{u}) &= 0 &(382)\end{aligned}$$

**Energy conservation** As usual, we treat the temporal discretization with typical time-stepping methods (implicit Euler, BDF, Crank-Nicolson); (alternatively, one could also use space-time-DG). For these methods, space- and time- discretization are independent,i.e. it is not important whether we do time- discretization first and space- discretization second, or the other way around. In both ways, we end up with the same method.

- Implicit Euler for (371)

$$\frac{1}{\Delta t}(\underline{u}^1 - \underline{u}^0, \underline{v}) + \mathrm{Ns}((\underline{u}^1,\psi^1)(\underline{v},w)) = \int_\Omega \underline{f}\cdot\underline{v}\,dV\ \forall(\underline{v},w) \tag{383}$$

- Crank-Nicolson

$$\frac{1}{\Delta t}(\underline{u}^1 - \underline{u}^0, \underline{v}) + \frac{1}{2}\mathrm{Ns}((\underline{u}^1,\psi^1)(\underline{v},w)) + \frac{1}{2}\mathrm{Ns}((\underline{u}^0,\psi^0)(\underline{v},w)) = \int_\Omega \underline{f}\cdot\underline{v}\,dV\ \forall(\underline{v},w) \tag{384}$$

**Energy conservation for Crank Nicolson**

$$\begin{aligned}
\frac{1}{\Delta t}(E^1 - E^0) &= \frac{1}{\Delta t}(\underline{u}^1, \underline{u}^1) - \frac{1}{\Delta t}(\underline{u}^0, \underline{u}^0) \\
&= \underbrace{\frac{-1}{2}\mathrm{Ns}((\underline{u}^1, \psi^1)(\underline{u}^1, \psi^1))}_{=0} + \frac{-1}{2}\mathrm{Ns}((\underline{u}^0, \psi^0)(\underline{u}^1, \psi^1)) + \underbrace{\frac{1}{\Delta t}(\underline{u}^0, \underline{u}^1) - \frac{1}{\Delta t}(\underline{u}^0, \underline{u}^0)}_{=\frac{1}{\Delta t}(\underline{u}^1 - \underline{u}^0, \underline{u}^0)} \\
&= \frac{-1}{2}\mathrm{Ns}((\underline{u}^0, \psi^0)(\underline{u}^1, \psi^1)) + \frac{-1}{2}\mathrm{Ns}((\underline{u}^1, \psi^1)(\underline{u}^0, \psi^0)) + \underbrace{\frac{-1}{2}\mathrm{Ns}((\underline{u}^0, \psi^0)(\underline{u}^0, \psi^0))}_{=0} \\
&= \frac{-1}{2}(t(\underline{u}^0, \underline{u}^1, \underline{u}^1, ) - b(\psi^0, \underline{u}^1) + b(\psi^1, \underline{u}^0) + t(\underline{u}^1, \underline{u}^1, \underline{u}^0) - b(\psi^1, \underline{u}^0) + b(\psi^0, \underline{u}^1)) = 0 \quad (385)
\end{aligned}$$

## 8.2 Steady Stokes, discrete setting

We reuse the SIP from section 6.5 and the Poisson system from section 7. As for the Poisson system, one has to fulfill the inf-sup condition to (see section 7.2) in order to guarantee well-posedness of the problem. In the context of the Stokes-equation, this is also known as the *LBB-condition (Ladyzhenskaya-Babuška-Brezzi)*.

**Mixed order discretizations**

- The polynomial degree of velocity is $p$, for pressure it is $p-1$.
- Stability is only known/proven for triangular grids, without hanging nodes, $p \leq 3$, see Girault et al. (2005).
- Seems to work also on other grids and for higher orders.
- We search for $(\underline{u}, \psi) \in \mathbb{P}_p(\mathcal{K}_h)^D \times \mathbb{P}_{p-1}(\mathcal{K}_h)$, so that

$$\begin{aligned}
a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) &= \int_\Omega \underline{f} \cdot \underline{v}\, dV \qquad \forall \underline{v} \in \mathbb{P}_p(\mathcal{K}_h)^D \quad (386) \\
-b(\tau, \underline{u}) &= 0 \qquad \forall \tau \in \mathbb{P}_{p-1}(\mathcal{K}_h) \quad (387)
\end{aligned}$$

- Definition of diffusive terms:

$$a(\underline{u}, \underline{v}) := \frac{1}{\mathrm{Re}} \sum_d a_{\mathrm{sip}}(u_d, v_d) \quad (388)$$

**Equal order discretization:**

- The same polynomial order $p$ for velocity and pressure.
- Requires an additional stabilization term, stability proof Di Pietro and Ern (2012)
- We search for $(\underline{u}, \psi) \in \mathbb{P}_p(\mathcal{K}_k)^D \times \mathbb{P}_p(\mathcal{K}_k)$, so that

$$\begin{aligned}
a(\underline{u}, \underline{v}) + b(\psi, \underline{v}) &= \int_\Omega \underline{f} \cdot \underline{v}\, dV \quad \forall \underline{v} \in \mathbb{P}_p(\mathcal{K}_h)^D \quad (389) \\
-b(\tau, \underline{u}) + \mathrm{Re}\ s(\psi, \tau) &= 0 \qquad \forall \tau \in \mathbb{P}_p(\mathcal{K}_h) \quad (390)
\end{aligned}$$

  where the pressure stabilization term is defined as

$$s(\psi, \tau) := \int_{\Gamma_{\mathrm{int}}} \eta_s \left[\!\left[\psi\right]\!\right] \left[\!\left[\tau\right]\!\right] dA \quad (391)$$

  and $\eta_s$ equal to the area of the respective cell-face.

### 8.3 Steady Navier-Stokes, discrete setting

**Discretization of the convective part:** We need an approximation to the trilinear form $t(-,-,-)$.

- Skew- Symmetric form, see Di Pietro and Ern (2012)

$$t(\underline{c},\underline{v},\underline{u}) = \int_\Omega (\underline{c}\cdot\nabla\underline{v})\cdot\underline{u} + \frac{1}{2}\operatorname{div}(\underline{c})(\underline{v}\cdot\underline{u})\,dV + \oint_{\Gamma\setminus\Gamma_D} (\{\underline{c}\}\cdot\underline{n}_\Gamma)\,([\![\underline{v}]\!]\cdot\{\underline{u}\})\,dS - \oint_\Gamma \frac{1}{2}\,[\![\underline{c}]\!]\cdot\underline{n}_\Gamma\,\{\underline{u}\cdot\underline{v}\}\,dA, \quad (392)$$

  (Temam's modification at discrete level)

- Flux-based form:

$$t(\underline{c},\underline{v},\underline{u}) = \oint_\Gamma \hat{\underline{F}}\cdot\underline{u}\,dA - \int_\Omega (\underline{c}\otimes\underline{v}) : \nabla\underline{u}\,dV \quad (393)$$

  For the flux $\hat{\underline{F}}$ one might e.g. use an upwind-flux as presented in section 3.3. The upwind-direction can be identified based on the velocity $\{\underline{u}\}$.

**Treatment of Nonlinear Terms** We have to solve: find $(\underline{u},\psi)\in\mathbb{P}_p^D(\mathcal{K}_h)\times\mathbb{P}_p(\mathcal{K}_h)$ so that

$$t(\underline{u},\underline{u},\underline{v}) + a(\underline{u},\underline{v}) + b(\psi,\underline{v}) = \int_\Omega \underline{f}\cdot\underline{v}\,dV \quad \forall\underline{v}\in\mathbb{P}_p(\mathcal{K}_h)^D \quad (394)$$

$$-b(\tau,\underline{u}) + \text{Re}\; s(\psi,\tau) = 0 \quad \forall\tau\in\mathbb{P}_p(\mathcal{K}_h) \quad (395)$$

Obviously, this problem is nonlinear. Therefore it has to be solved iteratively. We search for a sequence

$$(\underline{u}^1,\psi^1),(\underline{u}^2,\psi^2)\cdots\to(\underline{u},\psi) \quad (396)$$

which converges to the solution $(\underline{u},\psi)$ of the problem ((394), (395)). In the $\vartheta$-th iteration ($\vartheta = 1,2,\dots$) we solve

$$t(\underline{u}^{\vartheta-1},\underline{u}^*,\underline{v}) + a(\underline{u}^\vartheta,\underline{v}) + b(\psi^*,\underline{v}) = \int_\Omega \underline{g}_\Omega\cdot\underline{v}\,dV \quad \forall\underline{v}\in\mathbb{P}_p(\mathcal{K}_h)^D \quad (397)$$

$$-b(\tau,\underline{u}^*) + \text{Re}\; s(\psi^*,\tau) = 0 \quad \forall\tau\in\mathbb{P}_p(\mathcal{K}_h) \quad (398)$$

and define

$$\underline{u}^\vartheta = \underline{u}^{\vartheta-1}(\alpha_u - 1) + \underline{u}^*\cdot\alpha_u \quad (399)$$

$$\psi^\vartheta = \psi^{\vartheta-1}(\alpha_\psi - 1) + \psi^*\cdot\alpha_\psi \quad (400)$$

The under-relaxation factors $\alpha_u$ and $\alpha_\psi$, $0<\alpha_u,\ \alpha_\psi\le 1$ can be tuned to improve the convergence behavior of the method. As initial-values one might just use $\underline{u}^0 = 0$, $\psi^0 = 0$. Most of the time, one might just use $\underline{a}_u = \alpha_\psi = 1$. In this case, the iterative procedure can be interpreted as a Picard-iteration or a fixpoint-iteration. Alternatively, it may be called Oseen iteration.

# References


Bernardo Cockburn and Chi-Wang Shu. Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*. Springer, Berlin; New York, 1st edition, 2012. ISBN 978-3-642-22980-0.

Gregor Gassner, C. Altmann, F. Hindenlang, M. Staudenmeier, and C. D. Munz. Explicit Discontinuous Galerkin Schemes with Adaptation in Space and Time. In *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adative and HP-multigrid methods*, volume 1. Von Karman Institute for Fluid Dynamics, Rhode Saint Genese, Belgium, 2010. ISBN 978-2-930389-98-2.

Vivette Girault, Béatrice Rivière, and Mary F. Wheeler. A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier-Stokes problems. *Mathematics of Computation*, 74:53–84, 2005. ISSN 1088-6842.

Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer, Berlin, 1st edition, 2008. ISBN 0-387-72065-0.

Koen Hillewaert. *Development of the Discontinuous Galerkin Method for high-resolution, large scale CFD and acoustics in industrial geometries*. PhD thesis, Universit/'e catholique de Louvain, Louvain-la-Neuve, Belgium, 2013.

Randall J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, New York, 2002. ISBN 978-0-521-81087-6 978-0-521-00924-9.

Eleuterio F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer, Berlin/Heidelberg, 3rd edition, 2009. ISBN 978-3-540-25202-3.

Tim Warburton. A Low-Storage Curvilinear Discontinuous Galerkin Method for Wave Problems. *SIAM Journal on Scientific Computing*, 35(4):A1987–A2012, 2013. ISSN 1064-8275, 1095-7197. doi: 10.1137/120899662.