# Application of Discontinuous Galerkin methods to the unsteady, compressible Navier-Stokes equations

## A review of the state of the art

Björn Müller
Seminar Strömungsmechanik, Kontinuumsmechanik und geophysikalische Mechanik
Fachgebiet für Strömungsdynamik (FDY)
TU Darmstadt

March 3, 2010

# Outline



Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

Time discretization

Outlook on some advanced features

# Overview

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

Time discretization

Outlook on some advanced features

# Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main goal: Compute higher order solutions of partial differential equations

# Motivation



- Main goal: Compute higher order solutions of partial differential equations
- If possible, the order of the approximation should be adjustable on each element separately (so called *hp-adaptivity*)

## Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main goal: Compute higher order solutions of partial differential equations
- If possible, the order of the approximation should be adjustable on each element separately (so called *hp-adaptivity*)
- Main idea: Intelligent combination of the favorable properties of the Finite Volume Method (FVM) and the Finite Element Method (FEM)
  $\rightarrow$ Discontinuous Galerkin (DG, sometimes also called DG-FEM)

# Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main goal: Compute higher order solutions of partial differential equations
- If possible, the order of the approximation should be adjustable on each element separately (so called *hp-adaptivity*)
- Main idea: Intelligent combination of the favorable properties of the Finite Volume Method (FVM) and the Finite Element Method (FEM)
  $\rightarrow$ Discontinuous Galerkin (DG, sometimes also called DG-FEM)

| Method | High-order accuracy | hp-adaptivity | Conservativity |
|---|---|---|---|
| FVM | ($\checkmark$) | x | $\checkmark$ |
| FEM | $\checkmark$ | ($\checkmark$) | ($\checkmark$) |
| DG | $\checkmark$ | $\checkmark$ | $\checkmark$ |

Figure: Comparison of the different approaches

# The general idea of DG



- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells

# The general idea of DG

- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials

# The general idea of DG

- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials
- We allow discontinuities at the cell interfaces

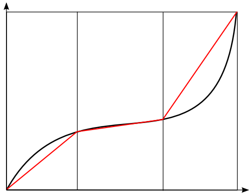
# The general idea of DG



- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials
- We allow discontinuities at the cell interfaces
- Thus, the recovered solution can be interpreted as a generalized Finite Volume solution

# The general idea of DG

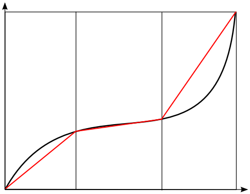
- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials
- We allow discontinuities at the cell interfaces
- Thus, the recovered solution can be interpreted as a generalized Finite Volume solution



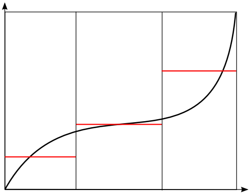
(a) First order FEM

# The general idea of DG

- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials
- We allow discontinuities at the cell interfaces
- Thus, the recovered solution can be interpreted as a generalized Finite Volume solution



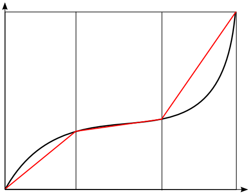
(a) First order FEM



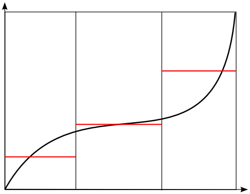
(b) Zeroth order DG (FVM)

# The general idea of DG

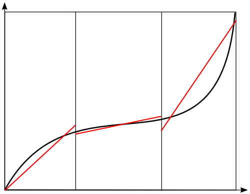
- We want to approximate the solution of a given partial differential equation inside a domain that is subdivided into a number of disjoint cells
- In each cell, we represent the solution as a weighted sum of basis polynomials
- We allow discontinuities at the cell interfaces
- Thus, the recovered solution can be interpreted as a generalized Finite Volume solution



(a) First order FEM



(b) Zeroth order DG (FVM)



(c) First order DG

# Workflow



- The general workflow is similar to the FEM

# Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

- The general workflow is similar to the FEM
  1. Multiplication of the equation by a test function

# Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

- The general workflow is similar to the FEM
  1. Multiplication of the equation by a test function
  2. Integration and requiring that the local residual is orthogonal to all test functions

# Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

- The general workflow is similar to the FEM
  1. Multiplication of the equation by a test function
  2. Integration and requiring that the local residual is orthogonal to all test functions
  3. Application of Gauss' theorem

# Workflow

- The general workflow is similar to the FEM
  1. Multiplication of the equation by a test function
  2. Integration and requiring that the local residual is orthogonal to all test functions
  3. Application of Gauss' theorem
  4. Specification of a *numerical flux* so that a unique value at interfaces can be recovered

# Workflow

TECHNISCHE UNIVERSITÄT DARMSTADT

- The general workflow is similar to the FEM
  1. Multiplication of the equation by a test function
  2. Integration and requiring that the local residual is orthogonal to all test functions
  3. Application of Gauss' theorem
  4. Specification of a *numerical flux* so that a unique value at interfaces can be recovered
- For the sake of simplicity, this procedure will be introduced by means of a scalar consveration law in the following

## Example:
## Application to a scalar conservation law



- Considered equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) = 0 \tag{1}$$

# Example:
# Application to a scalar conservation law

TECHNISCHE UNIVERSITÄT DARMSTADT

- Considered equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) = 0 \qquad (1)$$

- The function $\vec{f}$ is an arbitrary non-linear function

# Example:
# Application to a scalar conservation law

TECHNISCHE UNIVERSITÄT DARMSTADT

- Considered equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) = 0 \tag{1}$$

- The function $\vec{f}$ is an arbitrary non-linear function
- Multiplication by the test function $\Phi$ and integration over the cell $K$ yields

$$\int_K \left( \frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) \right) \Phi d\vec{x} = 0 \tag{2}$$

# Example:
# Application to a scalar conservation law

TECHNISCHE UNIVERSITÄT DARMSTADT

- Considered equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) = 0 \tag{1}$$

- The function $\vec{f}$ is an arbitrary non-linear function
- Multiplication by the test function $\Phi$ and integration over the cell $K$ yields

$$\int_K \left( \frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) \right) \Phi d\vec{x} = 0 \tag{2}$$

- After application of Gauss' theorem:

$$\int_K \frac{\partial u}{\partial t} \Phi d\vec{x} = - \int_{\partial K} \left( \vec{f}(u) \cdot \vec{n} \right) \Phi ds + \int_K \vec{f}(u) \cdot \nabla \Phi d\vec{x} \tag{3}$$

# Example continued: Approximation of the solution

TECHNISCHE UNIVERSITÄT DARMSTADT

- (Modal) Approximation of order $p$ of the exact solution in cell $K$:

$$u(\vec{x}, t) \approx \tilde{u}(\vec{x}, t) = \sum_{i=1}^{N} \tilde{u}_i^K(t) \cdot \varphi_i(\vec{x}) \tag{4}$$

# Example continued: Approximation of the solution

TECHNISCHE UNIVERSITÄT DARMSTADT

- (Modal) Approximation of order $p$ of the exact solution in cell $K$:

$$u(\vec{x}, t) \approx \tilde{u}(\vec{x}, t) = \sum_{i=1}^{N} \tilde{u}_i^K(t) \cdot \varphi_i(\vec{x}) \tag{4}$$

- (Modal) Basis functions: Polynomials $\varphi_i$ with

$$\text{degree}(\varphi_i) \leq p \quad \forall i \tag{5}$$

satisfying the orthogonality condition

$$\int_K \varphi_i \cdot \varphi_j \, d\vec{x} = \delta_{ij} \quad \forall i, j \tag{6}$$

# Example continued: Approximation of the solution

TECHNISCHE UNIVERSITÄT DARMSTADT

- (Modal) Approximation of order $p$ of the exact solution in cell $K$:

$$u(\vec{x}, t) \approx \tilde{u}(\vec{x}, t) = \sum_{i=1}^{N} \tilde{u}_i^K(t) \cdot \varphi_i(\vec{x}) \tag{4}$$

- (Modal) Basis functions: Polynomials $\varphi_i$ with

$$\mathrm{degree}(\varphi_i) \leq p \ \ \forall i \tag{5}$$

satisfying the orthogonality condition

$$\int_K \varphi_i \cdot \varphi_j \, d\vec{x} = \delta_{ij} \ \ \forall i, j \tag{6}$$

- Usually, the polynomials $\varphi$ are also chosen as test functions $\phi$

# Example continued: Handling of the discontinuity

- Notation: Quantities tagged with “-” are associated with the value *inside* the considered cell while quantities with “+” are associated with the adjacent cell

# Example continued:
# Handling of the discontinuity

- Notation: Quantities tagged with “-” are associated with the value *inside* the considered cell while quantities with “+” are associated with the adjacent cell
- After insertion of the approximation into the weak formulation:

$$\int_K \frac{\partial \tilde{u}}{\partial t} \Phi d\vec{x} = -\int_{\partial K} \left( \vec{f}(\tilde{u}) \cdot \vec{n} \right) \Phi^- ds + \int_K \vec{f}(\tilde{u}) \cdot \nabla \Phi d\vec{x} \tag{7}$$

## Example continued: Handling of the discontinuity

- Notation: Quantities tagged with “-” are associated with the value *inside* the considered cell while quantities with “+” are associated with the adjacent cell
- After insertion of the approximation into the weak formulation:

$$\int_K \frac{\partial \tilde{u}}{\partial t} \Phi d\vec{x} = -\int_{\partial K} \left( \vec{f}(\tilde{u}) \cdot \vec{n} \right) \Phi^- ds + \int_K \vec{f}(\tilde{u}) \cdot \nabla \Phi d\vec{x} \tag{7}$$

- This expression is not uniquely defined because we allow discontinuities

# Example continued: Handling of the discontinuity

TECHNISCHE UNIVERSITÄT DARMSTADT

- Notation: Quantities tagged with “-” are associated with the value *inside* the considered cell while quantities with “+” are associated with the adjacent cell
- After insertion of the approximation into the weak formulation:

$$\int_K \frac{\partial \tilde{u}}{\partial t} \Phi d\vec{x} = -\int_{\partial K} \left( \vec{f}(\tilde{u}) \cdot \vec{n} \right) \Phi^- ds + \int_K \vec{f}(\tilde{u}) \cdot \nabla \Phi d\vec{x} \tag{7}$$

- This expression is not uniquely defined because we allow discontinuities
- Introduction of a so called *flux function*:

$$\vec{f}(\tilde{u}) \cdot \vec{n} \approx g(\tilde{u}^-, \tilde{u}^+) \tag{8}$$

## Example continued: Handling of the discontinuity

TECHNISCHE UNIVERSITÄT DARMSTADT

- Notation: Quantities tagged with "-" are associated with the value *inside* the considered cell while quantities with "+" are associated with the adjacent cell
- After insertion of the approximation into the weak formulation:

$$\int_K \frac{\partial \tilde{u}}{\partial t} \Phi d\vec{x} = - \int_{\partial K} \left( \vec{f}(\tilde{u}) \cdot \vec{n} \right) \Phi^- ds + \int_K \vec{f}(\tilde{u}) \cdot \nabla \Phi d\vec{x} \tag{7}$$

- This expression is not uniquely defined because we allow discontinuities
- Introduction of a so called *flux function*:

$$\vec{f}(\tilde{u}) \cdot \vec{n} \approx g(\tilde{u}^-, \tilde{u}^+) \tag{8}$$

- The choice of $g$ defines the numerical properties of the whole scheme!

## Example continued:
## The flux functions

- Many choices for the flux function are possible

# Example continued: The flux functions



- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method

# Example continued: The flux functions

- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:

## Example continued: The flux functions

- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:
  - Mean value flux

# Example continued: The flux functions



- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:
  - Mean value flux
  - Upwind formulations

# Example continued:
# The flux functions



- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:
  - Mean value flux
  - Upwind formulations
  - Dozens of problem-dependent formulations

# Example continued: The flux functions



- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:
  - Mean value flux
  - Upwind formulations
  - Dozens of problem-dependent formulations
- Actually, the topic of flux functions is quite extensive and will not be discussed here any further

# Example continued: The flux functions



- Many choices for the flux function are possible
- They are the most important factor in terms of consistency, stability and accuracy of the method
- Existing flux functions:
  - Mean value flux
  - Upwind formulations
  - Dozens of problem-dependent formulations
- Actually, the topic of flux functions is quite extensive and will not be discussed here any further
- Note: Another possibility of deriving a DG scheme is the so called *strong formulation*. For details see [Gassner2009b] or [Hesthaven2007]

# Overview

# Overview

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations
- Formulation of the equations
- Application of the DG approach

Time discretization

Outlook on some advanced features

# Notation



| Symbol ($i, j = 1, 2, 3$) | Physical interpretation | Unit |
|---|---|---|
| $x_i$ | Cartesian coordinates | $m$ |
| $u_i$ | Velocity components | $\frac{m}{s}$ |
| $p$ | Pressure | $\frac{N}{m^2}$ |
| $\tau_{ij}$ | Components of the stress tensor | $\frac{N}{m^2}$ |
| $\rho$ | Mass density | $\frac{kg}{m^3}$ |
| $e$ | Specific inner energy | $\frac{J}{kg}$ |
| $q_i$ | Components of the heat flow | $\frac{J}{m^2}$ |
| $\mu$ | Dynamic viscosity | $\frac{Ns}{m^2}$ |
| $T$ | Absolute temperature | $K$ |
| $\lambda$ | Thermal conductivity | $\frac{W}{Km}$ |

Figure: The notation used for the formulation of the Navier-Stokes equations

# Assumptions

- Newtonian fluid ($i, j = 1, 2, 3$)

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \tag{9}$$

# Assumptions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Newtonian fluid ($i, j = 1, 2, 3$)

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \tag{9}$$

- Fourier's law of heat conduction ($i = 1, 2, 3$):

$$\frac{\partial q_i}{\partial t} = -\lambda \frac{\partial T}{\partial x_i} \tag{10}$$

# Standard formulation for a Newtonian fluid

- Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_j} = 0 \qquad (11)$$

# Standard formulation for a Newtonian fluid

TECHNISCHE UNIVERSITÄT DARMSTADT

- Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_j} = 0 \tag{11}$$

- Momentum

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{12}$$

# Standard formulation for a Newtonian fluid

TECHNISCHE UNIVERSITÄT DARMSTADT

- Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_j} = 0 \tag{11}$$

- Momentum

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{12}$$

- Energy

$$\rho \left( \frac{\partial e}{\partial t} + u_i \frac{\partial e}{\partial x_i} \right) = \frac{\partial}{\partial x_i} \left( \lambda \frac{\partial T}{\partial x_i} \right) - p \frac{\partial u_i}{\partial x_i} + 2\mu \left( \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j} - \frac{1}{3} \left( \frac{\partial u_i}{\partial x_i} \right)^2 \right) \tag{13}$$

# Standard formulation for a Newtonian fluid

TECHNISCHE UNIVERSITÄT DARMSTADT

- Continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_j} = 0 \tag{11}$$

- Momentum

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{12}$$

- Energy

$$\rho \left( \frac{\partial e}{\partial t} + u_i \frac{\partial e}{\partial x_i} \right) = \frac{\partial}{\partial x_i} \left( \lambda \frac{\partial T}{\partial x_i} \right) - p \frac{\partial u_i}{\partial x_i} + 2\mu \left( \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j} - \frac{1}{3} \left( \frac{\partial u_i}{\partial x_i} \right)^2 \right) \tag{13}$$

- We still need relations for $T$ and $p$

# Exemplary material laws

- $T$ and $p$ are determined using a problem-dependent material law

# Exemplary material laws

- $T$ and $p$ are determined using a problem-dependent material law
- For a thermally and calorically ideal gas (e.g. air in most configurations)

$$p = \rho R T \tag{14}$$

$$e = c_v T \tag{15}$$

with *the ideal gas constant R* is the standard choice

# Exemplary material laws

TECHNISCHE UNIVERSITÄT DARMSTADT

- $T$ and $p$ are determined using a problem-dependent material law
- For a thermally and calorically ideal gas (e.g. air in most configurations)

$$p = \rho R T \tag{14}$$

$$e = c_v T \tag{15}$$

  with *the ideal gas constant* $R$ is the standard choice
- Additionally, the viscosity $\mu$ will usually depend on the temperature $T$

# Exemplary material laws

TECHNISCHE UNIVERSITÄT DARMSTADT

- $T$ and $p$ are determined using a problem-dependent material law
- For a thermally and calorically ideal gas (e.g. air in most configurations)

$$p = \rho R T \tag{14}$$

$$e = c_v T \tag{15}$$

with *the ideal gas constant R* is the standard choice

- Additionally, the viscosity $\mu$ will usually depend on the temperature $T$
- For example for ideal gases usually *Sutherland's law*

$$\mu = \mu_0 \left( \frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S}{T + S} \tag{16}$$

with the *reference temperature* $T_0$, the *reference viscosity* $\mu_0 = \mu(T_0)$ and the *Sutherland temperature S* is used

# Dimensionless formulation



| Physical quantity | Dimensionless formulation |
|---|---|
| Cartesian coordinates | $\hat{x}_i = \frac{x_i}{L}$ |
| Time | $\hat{t} = \frac{t u_\infty}{L}$ |
| Mass density | $\hat{\rho} = \frac{\rho}{\rho_\infty}$ |
| Components of the momentum | $\hat{\rho}\hat{u}_i = \frac{\rho u_i}{\rho_\infty u_\infty}$ |
| Pressure | $\hat{p} = \frac{p}{\rho_\infty u_\infty^2}$ |
| Energy per volume | $\hat{\rho}\hat{E} = \frac{\rho e + u_i u_i}{\rho_\infty u_\infty^2}$ |
| Absolute temperature | $\hat{T} = \frac{T}{T_\infty}$ |
| Thermal conductivity | Depends on the law for $p$ and $T$ |
| Components of the stress tensor | Depends on the law for $\hat{\mu}$ |

Figure: Dimensionless quantities (for $i = 1, 2, 3$) for the characteristic quantities $L$, $\rho_\infty$, $u_\infty$, $T_\infty$

# The general conservation form

- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*

# The general conservation form

- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*
- That is, conserved quantities (like the mass) are expressed as divergences

# The general conservation form

TECHNISCHE UNIVERSITÄT DARMSTADT

- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*
- That is, conserved quantities (like the mass) are expressed as divergences
- For DG such a formulation is mandatory!

# The general conservation form

- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*
- That is, conserved quantities (like the mass) are expressed as divergences
- For DG such a formulation is mandatory!
- Abstract (dimensionless) formulation of a second order conservation law:

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re} \frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \tag{17}$$

# The general conservation form



- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*
- That is, conserved quantities (like the mass) are expressed as divergences
- For DG such a formulation is mandatory!
- Abstract (dimensionless) formulation of a second order conservation law:

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re} \frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \tag{17}$$

- The $F_i$ represent the *convectice fluxes* while the $G_i$ denote the *dissipative fluxes*

# The general conservation form

- For numerical calculations it is often useful to write conserved quantities in the so called *conservation form*
- That is, conserved quantities (like the mass) are expressed as divergences
- For DG such a formulation is mandatory!
- Abstract (dimensionless) formulation of a second order conservation law:

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re}\frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \qquad (17)$$

- The $F_i$ represent the *convectice fluxes* while the $G_i$ denote the *dissipative fluxes*
- For the application of the DG approach, only weak assumptions for concerning the structure of $F_i$ and $G_i$ are needed

# Navier Stokes in conservation form



- In our case:

## Navier Stokes in conservation form

TECHNISCHE UNIVERSITÄT DARMSTADT

- In our case:
  - Vector of unknowns

$$U(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{t}) = \begin{pmatrix} \hat{\rho} \\ \hat{\rho}\hat{u}_1 \\ \hat{\rho}\hat{u}_2 \\ \hat{\rho}\hat{u}_3 \\ \hat{\rho}\hat{E} \end{pmatrix} \tag{18}$$

## Navier Stokes in conservation form

- In our case:
  - Vector of unknowns

$$U(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{t}) = \begin{pmatrix} \hat{\rho} \\ \hat{\rho}\hat{u}_1 \\ \hat{\rho}\hat{u}_2 \\ \hat{\rho}\hat{u}_3 \\ \hat{\rho}\hat{E} \end{pmatrix} \tag{18}$$

  - Convective and diffusive fluxes:

$$F_i(U) = \begin{pmatrix} \hat{\rho}\hat{u}_i \\ \hat{\rho}\hat{u}_i\hat{u}_1 + \delta_{1i}\hat{p} \\ \hat{\rho}\hat{u}_i\hat{u}_2 + \delta_{2i}\hat{p} \\ \hat{\rho}\hat{u}_i\hat{u}_3 + \delta_{3i}\hat{p} \\ \hat{u}_i(\hat{\rho}\hat{E} + \hat{p}) \end{pmatrix} \quad G_i(U, \nabla U) = \begin{pmatrix} 0 \\ \hat{\tau}_{i1} \\ \hat{\tau}_{i2} \\ \hat{\tau}_{i3} \\ \hat{u}_j\hat{\tau}_{ji} - \hat{\lambda}\frac{\partial \hat{T}}{\partial \hat{x}_i} \end{pmatrix} \tag{19}$$

## Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

- Formulation of the equations
- Application of the DG approach

Time discretization

Outlook on some advanced features

# Second order systems: Naive approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main difference compared to the scalar example: *Second order terms*

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re} \frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \tag{20}$$

# Second order systems: Naive approach

- Main difference compared to the scalar example: *Second order terms*

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re}\frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \tag{20}$$

- Naive approach:

# Second order systems: Naive approach



- Main difference compared to the scalar example: *Second order terms*

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re} \frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \tag{20}$$

- Naive approach:
  - Introduce auxiliary variables and write equation 20 as a first order system:

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re} \frac{\partial G_i(U, V)}{\partial \hat{x}_i} = 0 \tag{21}$$

$$V_{ji} - \frac{\partial U_j}{\partial \hat{x}_i} = 0 \quad \text{for } j = 1, \dots, 5 \tag{22}$$

# Second order systems: Naive approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main difference compared to the scalar example: *Second order terms*

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re}\frac{\partial G_i(U, \nabla U)}{\partial \hat{x}_i} = 0 \qquad (20)$$

- Naive approach:
  - Introduce auxiliary variables and write equation 20 as a first order system:

$$\frac{\partial U}{\partial \hat{t}} + \frac{\partial F_i(U)}{\partial \hat{x}_i} - \frac{1}{Re}\frac{\partial G_i(U, V)}{\partial \hat{x}_i} = 0 \qquad (21)$$

$$V_{ji} - \frac{\partial U_j}{\partial \hat{x}_i} = 0 \quad \text{for } j = 1, \ldots, 5 \qquad (22)$$

  - Reformulation of equation 22 in conservation form:

$$V_{ji} - \frac{\partial \delta_{ik} U_j}{\partial \hat{x}_k} = 0 \quad \text{for } k = 1, \ldots, 3 \qquad (23)$$

# Alternative approach

- Main drawback of the naive approach:

# Alternative approach

- Main drawback of the naive approach:
  - Explosion of the number of unknowns

# Alternative approach

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had 5 variables

# Alternative approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had 5 variables
  - Now we have $5 + 5 \cdot 3 = 20$ variables

# Alternative approach

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had 5 variables
  - Now we have $5 + 5 \cdot 3 = 20$ variables
- Alternative: Ultra weak formulation (introduced in [Gassner2009b])

# Alternative approach

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had 5 variables
  - Now we have $5 + 5 \cdot 3 = 20$ variables
- Alternative: Ultra weak formulation (introduced in [Gassner2009b])
  - Idea: Apply a second spatial integration by parts

## Alternative approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had $5$ variables
  - Now we have $5 + 5 \cdot 3 = 20$ variables
- Alternative: Ultra weak formulation (introduced in [Gassner2009b])
  - Idea: Apply a second spatial integration by parts
  - As a result, an additional surface integral has to be treated and so called adjoint fluxes have to be introduced

# Alternative approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main drawback of the naive approach:
  - Explosion of the number of unknowns
  - Before the transformation, we had 5 variables
  - Now we have $5 + 5 \cdot 3 = 20$ variables
- Alternative: Ultra weak formulation (introduced in [Gassner2009b])
  - Idea: Apply a second spatial integration by parts
  - As a result, an additional surface integral has to be treated and so called adjoint fluxes have to be introduced
  - The approach seems quite promising but not much literature is available by now

# Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

# Types of discretization



- In general, the DG method allows the application of all common time discretization schemes

# Types of discretization

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:

# Types of discretization

TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit

# Types of discretization

TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly

# Types of discretization

TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step

# Types of discretization

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case

# Types of discretization

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit

# Types of discretization



- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms

# Types of discretization

TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms
    - Seems to be used less frequently than the other options

# Types of discretization

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms
    - Seems to be used less frequently than the other options
    - Main advantage: Large time-steps in low Mach-number flows

# Types of discretization

TECHNISCHE UNIVERSITÄT DARMSTADT

- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms
    - Seems to be used less frequently than the other options
    - Main advantage: Large time-steps in low Mach-number flows
  - Fully Explicit

# Types of discretization



- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the “standard” case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms
    - Seems to be used less frequently than the other options
    - Main advantage: Large time-steps in low Mach-number flows
  - Fully Explicit
    - All terms are treated explicitly

# Types of discretization



- In general, the DG method allows the application of all common time discretization schemes
- On the whole, one generally distinguishes between three types of schemes:
  - Semi-implicit
    - Usually linear terms are treated implicitly (e.g. using an Implicit-Euler or a Crank-Nicolson scheme) while all others are treated explicitly
    - The terms treated explicitly limit the maximum time-step
    - Can be considered the "standard" case
  - Fully Implicit
    - Nonlinear terms are linearized and treated implicitly as the linear terms
    - Seems to be used less frequently than the other options
    - Main advantage: Large time-steps in low Mach-number flows
  - Fully Explicit
    - All terms are treated explicitly
    - Main advantage: Low computational cost per time-step

# The restriction of the time-step length

TECHNISCHE UNIVERSITÄT DARMSTADT

- The length of a time-step using an explicit scheme is usually limited by the so called *CFL condition*

# The restriction of the time-step length

- The length of a time-step using an explicit scheme is usually limited by the so called *CFL condition*
- This condition generally depends on the applied scheme and the minimum cell size $\Delta x$

# The restriction of the time-step length



- The length of a time-step using an explicit scheme is usually limited by the so called *CFL condition*
- This condition generally depends on the applied scheme and the minimum cell size $\Delta x$
- For the one-dimensional heat equation, for example, one has

$$\Delta t_{\mathrm{max}} \propto \Delta x^2 \tag{24}$$

# The restriction of the time-step length

- The length of a time-step using an explicit scheme is usually limited by the so called *CFL condition*
- This condition generally depends on the applied scheme and the minimum cell size $\Delta x$
- For the one-dimensional heat equation, for example, one has

$$\Delta t_{\text{max}} \propto \Delta x^2 \tag{24}$$

- But in the context of DG methods, this becomes much more restrictive:

$$\Delta t_{\text{max}} \propto \frac{\Delta x^2}{(2N+1)^2} \tag{25}$$

(where $N$ is the polynomial degree of the approximation)

# The restriction of the time-step length

TECHNISCHE UNIVERSITÄT DARMSTADT

- The length of a time-step using an explicit scheme is usually limited by the so called *CFL condition*
- This condition generally depends on the applied scheme and the minimum cell size $\Delta x$
- For the one-dimensional heat equation, for example, one has

$$\Delta t_{\max} \propto \Delta x^2 \tag{24}$$

- But in the context of DG methods, this becomes much more restrictive:

$$\Delta t_{\max} \propto \frac{\Delta x^2}{(2N+1)^2} \tag{25}$$

  (where $N$ is the polynomial degree of the approximation)
- E.g. in [Gassner2009c] similar formulations for our case are shown and can be used to determine the time-step restriction

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)
- But: For low Mach number flows, for example, the time-step restriction is dominated by the speed fast *acoustic* waves

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)
- But: For low Mach number flows, for example, the time-step restriction is dominated by the speed fast *acoustic* waves
- Conclusion

## Dealing with time-step restrictions



- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)
- But: For low Mach number flows, for example, the time-step restriction is dominated by the speed fast *acoustic* waves
- Conclusion
  - There is no optimal setting for all situations

# Dealing with time-step restrictions



- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)
- But: For low Mach number flows, for example, the time-step restriction is dominated by the speed fast *acoustic* waves
- Conclusion
  - There is no optimal setting for all situations
  - High Mach number and/or highly dynamic $\rightarrow$ Fully explicit or semi-implicit

# Dealing with time-step restrictions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implicit methods often are nearly unconditionally stable which seems to make them superior
- But: Stability is not enough. For accurate results in fully unsteady flow simulations (e.g, think of turbulence), the time-step cannot be too large without losing information
- On the other hand, explicit methods are by far less computationally expensive (especially when local time-stepping can be used)
- But: For low Mach number flows, for example, the time-step restriction is dominated by the speed fast *acoustic* waves
- Conclusion
  - There is no optimal setting for all situations
  - High Mach number and/or highly dynamic $\rightarrow$ Fully explicit or semi-implicit
  - Low Mach number and/or steady $\rightarrow$ Fully implicit or semi-implicit

# Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

Time discretization

Outlook on some advanced features

- Local time-stepping
- Hp-adaptivity

# Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

Time discretization

Outlook on some advanced features

- Local time-stepping
- Hp-adaptivity

# The concept of local-time-stepping



- Usually a small number of tiny cells limits the maximum time-step size for the whole simulation

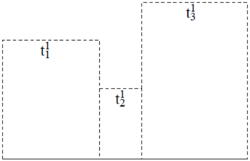
# The concept of local-time-stepping

TECHNISCHE UNIVERSITÄT DARMSTADT

- Usually a small number of tiny cells limits the maximum time-step size for the whole simulation
- In explicit DG schmes it is possible to apply this very small time-step only where it is really needed (using so called ADER or CERK schemes)

## The concept of local-time-stepping

- Usually a small number of tiny cells limits the maximum time-step size for the whole simulation
- In explicit DG schmes it is possible to apply this very small time-step only where it is really needed (using so called ADER or CERK schemes)

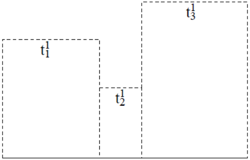



(a) Three cells with their maximum time-step

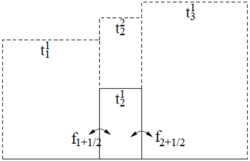
Figure: Visualization of the local-time-stepping procedure (taken from [Gassner2009c])

# The concept of local-time-stepping

- Usually a small number of tiny cells limits the maximum time-step size for the whole simulation
- In explicit DG schmes it is possible to apply this very small time-step only where it is really needed (using so called ADER or CERK schemes)




(a) Three cells with their maximum time-step




(b) Local advancement of the cell with the sharpest restriction

Figure: Visualization of the local-time-stepping procedure (taken from [Gassner2009c])

# Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Introduction to the Discontinuous Galerkin framework

Application of the DG approach to the compressible Navier-Stokes equations

Time discretization

Outlook on some advanced features

- Local time-stepping
- Hp-adaptivity

# Non-conforming meshes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Automatic grid adaptation in critical regions is desirabe for many applications

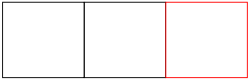
# Non-conforming meshes



- Automatic grid adaptation in critical regions is desirabe for many applications
- The main problem: It is generally not trivial to split single while retaining a *conforming* mesh

# Non-conforming meshes

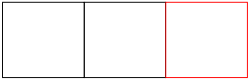
- Automatic grid adaptation in critical regions is desirabe for many applications
- The main problem: It is generally not trivial to split single while retaining a *conforming* mesh



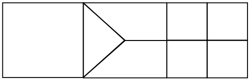
(a) Example of a mesh where the red cell should be refined

# Non-conforming meshes

- Automatic grid adaptation in critical regions is desirabe for many applications
- The main problem: It is generally not trivial to split single while retaining a *conforming* mesh



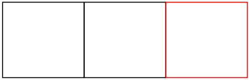
(a) Example of a mesh where the red cell should be refined



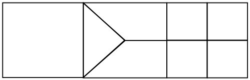
(b) Example of a conforming, mixed element refinement

# Non-conforming meshes

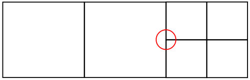
- Automatic grid adaptation in critical regions is desirabe for many applications
- The main problem: It is generally not trivial to split single while retaining a *conforming* mesh



(a) Example of a mesh where the red cell should be refined



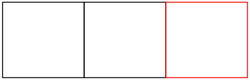
(b) Example of a conforming, mixed element refinement



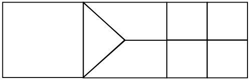
(c) Example of a non-conforming refinement with one hanging node (red)

# Non-conforming meshes

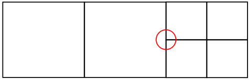
- Automatic grid adaptation in critical regions is desirabe for many applications
- The main problem: It is generally not trivial to split single while retaining a *conforming* mesh



(a) Example of a mesh where the red cell should be refined



(b) Example of a conforming, mixed element refinement



(c) Example of a non-conforming refinement with one hanging node (red)

- Due to the simple interaction between cells ($\rightarrow$ fluxes), DG schemes can (relatively) easily cope with so called *hanging nodes*

# Local order adaptation: Motivation

- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell

# Local order adaptation: Motivation

- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:

# Local order adaptation: Motivation



- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*

# Local order adaptation: Motivation



- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*
  - In regions with temporary strong dynamic changes one might want to increase the polynomial order

# Local order adaptation: Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*
  - In regions with temporary strong dynamic changes one might want to increase the polynomial order
- Main issues: Development of general, reliable error indicators and efficient data representation

# Local order adaptation: Motivation



- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*
  - In regions with temporary strong dynamic changes one might want to increase the polynomial order
- Main issues: Development of general, reliable error indicators and efficient data representation
- Popular choices in the context of CFD:

# Local order adaptation: Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*
  - In regions with temporary strong dynamic changes one might want to increase the polynomial order
- Main issues: Development of general, reliable error indicators and efficient data representation
- Popular choices in the context of CFD:
  - Detection of shocks: Gradient based detectors

# Local order adaptation: Motivation



- Adjacent cells only communicate through fluxes $\rightarrow$ There is no need for the approximation to be of the same polynomial degree in every cell
- There are cases where either the local increase or decrease of the polynomial order can be beneficial:
  - Near shocks, for example, one might want to decrease the order to avoid the so called *Gibbs phenomenon*
  - In regions with temporary strong dynamic changes one might want to increase the polynomial order
- Main issues: Development of general, reliable error indicators and efficient data representation
- Popular choices in the context of CFD:
  - Detection of shocks: Gradient based detectors
  - Detection of boundary layers: Vorticity based detectors

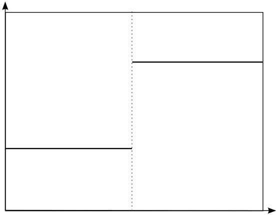
# Local order adaptation: Order reduction near shocks

TECHNISCHE UNIVERSITÄT DARMSTADT

- Problem: Unphysical oscillations of higher order approximations near jumps

# Local order adaptation: Order reduction near shocks

- Problem: Unphysical oscillations of higher order approximations near jumps

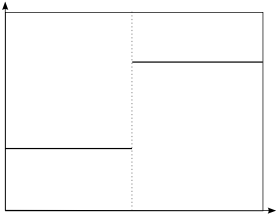


(a) Example of a discontinuous exact solution

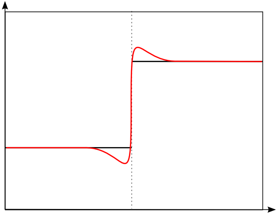
# Local order adaptation: Order reduction near shocks



- Problem: Unphysical oscillations of higher order approximations near jumps



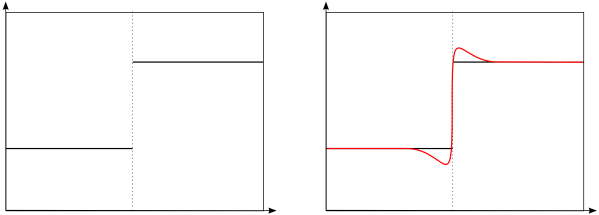
(a) Example of a discontinuous exact solution



(b) Gibbs phenomenon for a higher order approxmation

# Local order adaptation: Order reduction near shocks

- Problem: Unphysical oscillations of higher order approximations near jumps

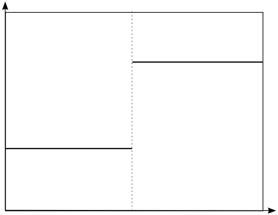


(a) Example of a discontinuous exact solution    (b) Gibbs phenomenon for a higher order approxmation

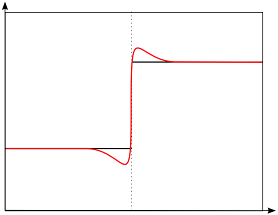
- In FVM, for example, usually *slope limiters* are applied to avoid this behaviour

# Local order adaptation: Order reduction near shocks

- Problem: Unphysical oscillations of higher order approximations near jumps



(a) Example of a discontinuous exact solution



(b) Gibbs phenomenon for a higher order approxmation

- In FVM, for example, usually *slope limiters* are applied to avoid this behaviour
- The simple order reduction in DG is a much “cleaner” approach since these limiters decrease the local order of exactness anyway

# Local order adaptation: Order elevation



- Regions with strong gradients have to be resolved very accurately
  $\rightarrow$ e.g. extremely fine meshes in boundary layers

# Local order adaptation: Order elevation

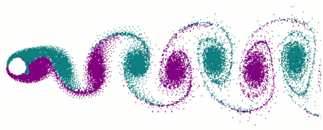
TECHNISCHE UNIVERSITÄT DARMSTADT

- Regions with strong gradients have to be resolved very accurately
  $\rightarrow$ e.g. extremely fine meshes in boundary layers
- Drawback: Critical regions must be defined a priori and/or cannot easily be adapted in case of dynamic changes

# Local order adaptation: Order elevation

TECHNISCHE UNIVERSITÄT DARMSTADT

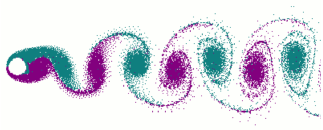
- Regions with strong gradients have to be resolved very accurately
  $\rightarrow$ e.g. extremely fine meshes in boundary layers
- Drawback: Critical regions must be defined a priori and/or cannot easily be adapted in case of dynamic changes
- Example: Kárman vortex street



# Local order adaptation:
# Order elevation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Regions with strong gradients have to be resolved very accurately
  $\rightarrow$ e.g. extremely fine meshes in boundary layers
- Drawback: Critical regions must be defined a priori and/or cannot easily be adapted in case of dynamic changes
- Example: Kárman vortex street

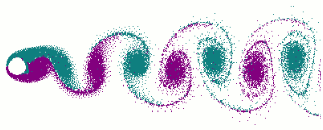


  - Whirls should be resolved very accurately but they move through the domain!

# Local order adaptation: Order elevation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Regions with strong gradients have to be resolved very accurately
  $\rightarrow$ e.g. extremely fine meshes in boundary layers
- Drawback: Critical regions must be defined a priori and/or cannot easily be adapted in case of dynamic changes
- Example: Kárman vortex street



  - Whirls should be resolved very accurately but they move through the domain!
  - DG approach: Dynamically increase the local order near whirls and decrease it again if the whirl has moved on

## The end

Thank you for your attention!

Any questions?

# References I

[Dumbser2005]
M. Dumbser
*Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*
Stuttgart, 2005

[Gassner2009b]
G. Gassner
*Discontinuous Galerkin methods for the unsteady compressible Navier-Stokes equations*
Stuttgart, 2009

# References II

TECHNISCHE UNIVERSITÄT DARMSTADT


[Gassner2009c]
G. Gassner, C. Altmann, F. Hindenlang, M. Staudenmeier and C. D. Munz
*Explicit Discontinuous Galerkin Schemes with Adaptation in Space and Time*
Stuttgart, 2009

[Hesthaven2007]
J. S. Hesthaven and T. Warburton
*Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*
Springer, Berlin, 2007

[Prandtl2008]
L. Prandtl
*Führer durch die Strömungslehre Grundlagen und Phänomene*
Teubner, Wiesbaden, 2008