LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

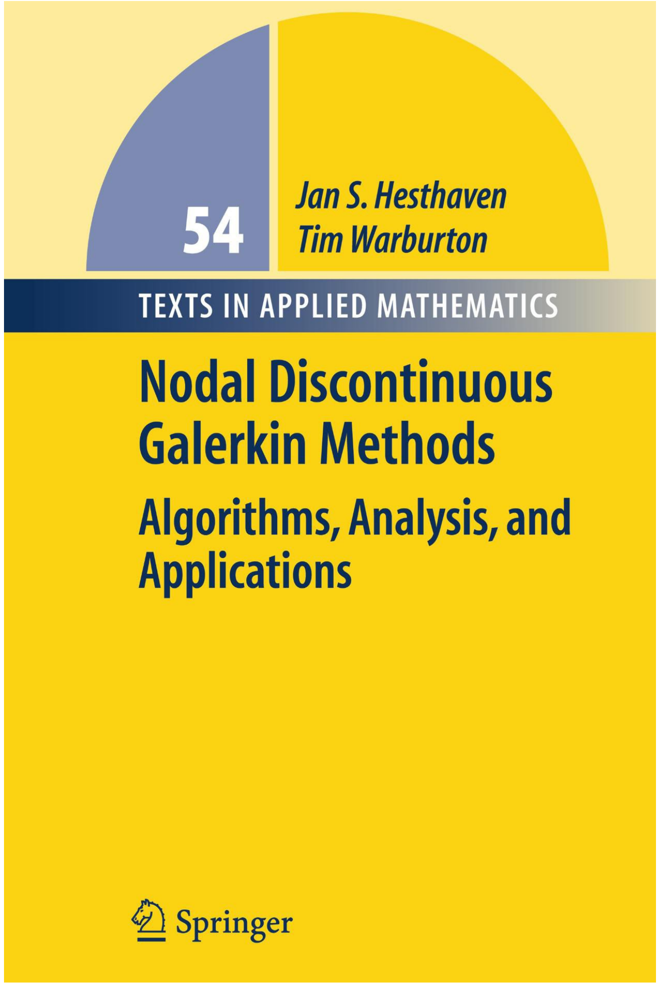
An introduction to the Discontinuous Galerkin Method

December 3 2014, Computational Seismology

Djamel Ziane & Stephanie Wollherr

Overview

1. Motivation
2. Discretization in space
3. Discretization in time



Motivation

Finite Difference Method

- **Pros:**
  - Simple and fast to implement
  - Explicit in time

- **Cons:**
  - no geometrical flexibility

Motivation

Finite Volume Method

- **Pros:**
  - Complex geometries
  - Energy conservation
  - Local communication, easy to parallelize
  - Explicit in time

- **Cons:**
  - inability to extend to higher-order accuracy

Motivation

Finite Element Method

- **Pros:**
  - Higher order accuracy can be combined with unstructured mesh

- **Cons:**
  - Implicit in time
    - Inversion of the global mass matrix (non-diagonal) for a explicit scheme
  - Global mass and stiffness matrix complicates parallelization

Motivation

Discontinuous Galerkin method: Finite Element + Finite Volume

Motivation

Discontinuous Galerkin method: Finite Element + Finite Volume

Pros:

- flexibility in grid discretization → complex geometries
- Local operations:
  - Local mass and stiffness matrices
  - Local approximation order
  - Easy to parallelize
- Energy conservation due the flux concept
- Diagonal mass matrix: explicit time scheme

Motivation

Discontinuous Galerkin method: Finite Element + Finite Volume

Pros:

- flexibility in grid discretization → complex geometries
- Local operations:
  - Local mass and stiffness matrices
  - Local approximation order
  - Easy to parallelize
- Energy conservation due the flux concept
- Diagonal mass matrix: explicit time scheme

Cons:

- Large number of degrees of freedom
  - High computational cost

The DG method: **1. Discretization in space**

$$\partial_t u(x,t) + \mu \; \partial_x u(x,t) = 0$$

Linear scalar wave equation

The DG method: **1. Discretization in space**

$$\partial_t u(x,t) + \mu \; \partial_x u(x,t) = 0$$

Linear scalar wave equation

- K non-overlapping elements with $\mathrm{D}^k = [x_l^k, x_r^k]$
- <u>local</u> representation of the solution for $x \in D_k$:

The DG method: 1. Discretization in space

$$\partial_t u(x,t) + \mu\ \partial_x u(x,t) = 0$$

Linear scalar wave equation

- K non-overlapping elements with $\mathrm{D}^k = [x_l^k, x_r^k]$
- local representation of the solution for $x \in D_k$:

$$u_h^k(x,t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t)\ P_n(x)$$

modal — **Legendre polynomials** not depending on special grid points inside the element

$$= \sum_{i=1}^{N_p} u_h^k(x_i,t)\ l_i^k(x)$$

nodal — Interpolating **Lagrange polynomials** at additional grid points inside the element

The DG method: 1. Discretization in space

- global solution: $$u(x,t) \approx u_h(x,t) = \bigoplus_{k=1}^{K} \underbrace{u_h^k(x,t)}_{\text{Local solutions}}$$

- For discretization in space: same approach as for FE methods:
  - Multiplication by an arbitrary test function $\phi_j(x)$
  - Integration over each element D_k

The Discontinuous Galerkin Method

The DG method: **1. Discretization in space**

- global solution: $$u(x,t) \approx u_h(x,t) = \bigoplus_{k=1}^{K} \underline{u_h^k(x,t)}$$

Local solutions

- For discretization in space: same approach as for FE methods:
  - Multiplication by an arbitrary test function $\phi_j(x)$
  - Integration over each element D_k

But: no continuity assumption for the solution at the boundary of an element!!!

The DG method: 1. Discretization in space

$$\boxed{\partial_t u(x,t) + \mu\ \partial_x u(x,t) = 0}$$

$$\int_{D^k} \partial_t u_h^k(x,t)\underline{\phi_j(x)}\ dx + \int_{D^k} \underline{\mu\ \partial_x u_h^k(x,t)\underline{\phi_j(x)}\ dx} = \ 0$$

The Discontinuous Galerkin Method

The DG method: 1. Discretization in space

$$\boxed{\partial_t u(x,t) + \mu\ \partial_x u(x,t) = 0}$$

$$\int_{D^k} \partial_t u_h^k(x,t)\underline{\phi_j(x)}\ dx + \int_{D^k} \underline{\mu\ \partial_x u_h^k(x,t)\underline{\phi_j(x)}\ dx} = \ 0$$

- Integration by parts:

$$\int_{D^k} \underline{\partial_x u_h^k(x,t)\phi_j(x)\ dx} =$$

$$-\int_{D_k} u_h^k(x,t)\cdot\partial_x\phi_j(x)\ dx + \int_{\partial D_k} u_h^k(x,t)\cdot\phi_j(x)\cdot\mathbf{n}\ dx$$

$\mathbf{n} =$ outer popinting normal

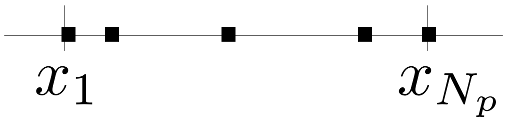
The DG method: 1. Discretization in space

Left hand side: $$\int_{D^k} \partial_t u_h^k(x,t)\phi_j(x)\ dx - \int_{D^k} \mu u_h^k(x,t)\partial_x\phi_j(x)\ dx$$

The DG method: 1. Discretization in space

Left hand side: $$\int_{D^k} \partial_t u_h^k(x,t)\phi_j(x)\ dx - \int_{D^k} \mu u_h^k(x,t)\partial_x \phi_j(x)\ dx$$

- Inserting the local (**nodal**) representation of the solution $u_h^k(x,t) = \sum_{i=1}^{N_p} u_h^k(x_i,t)\ l_i^k(x)$

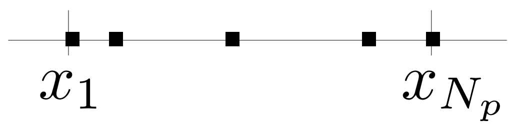


The Discontinuous Galerkin Method

The DG method: 1. Discretization in space

Left hand side:
$$\int_{D^k} \partial_t u_h^k(x,t)\phi_j(x)\ dx - \int_{D^k} \mu u_h^k(x,t)\partial_x \phi_j(x)\ dx$$

- Inserting the local (**nodal**) representation of the solution $u_h^k(x,t) = \sum_{i=1}^{N_p} u_h^k(x_i,t)\ l_i^k(x)$



- Lagrange polynomials as test functions $l_j^k(x)$

$$\sum_{i=1}^{N_p} \left(\partial_t \cdot u_h^k(x_i,t) \int_{D^k} l_i^k(x) l_j^k(x)\ dx - \mu u_h^k(x_i^k,t) \int_{D_k} l_i^k(x)\partial_x l_j^k(x) dx \right)$$

Entries of the mass matrix

Entries of the stiffness matrix S

The DG method: 1. Discretization in space

$$\partial_t u_h^k(t) \cdot M^k - \mu \cdot u_h^k(t) \cdot S^k$$

Local matrices for every element!

with $$[M^k]_{ij} = \left[\int_{D_k} l_i^k(x)\ l_j^k(x)\ dx \right]_{ij}$$

Mass matrix of the k-th element

$$[S^k]_{ij} = \left[\int_{D_k} l_i^k(x) \partial_x l_j^k(x)\ dx \right]_{ij}$$

Stiffness matrix of the k-th element

$$u_h^k(t) = (u^k(x_1, t), \ldots\ , u^k(x_{N_p} t))^T$$

(time dependent) coefficients,
N+1 DOF for each element

The DG method: 1. Discretization in space

Right hand side:
(without the source f)

$$\int_{\partial D_k} \mu \cdot u_h^k(x,t) \cdot l_j^k(x) \cdot \mathbf{n} \; dx$$

1D: outward pointing normal $\mathbf{n} = \pm 1$

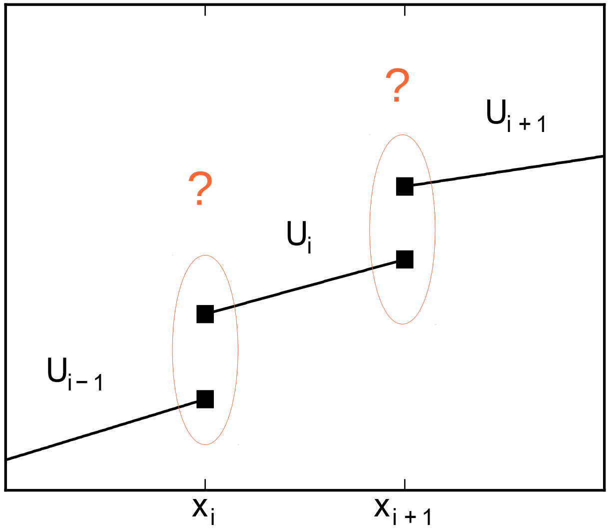
The DG method: 1. Discretization in space

Right hand side:
(without the source f)

$$\int_{\partial D_k} \mu \cdot u_h^k(x,t) \cdot l_j^k(x) \cdot \mathbf{n} \; dx$$

1D: outward pointing normal $\mathbf{n} = \pm 1$

How to calculate the values at the boundary of an element?



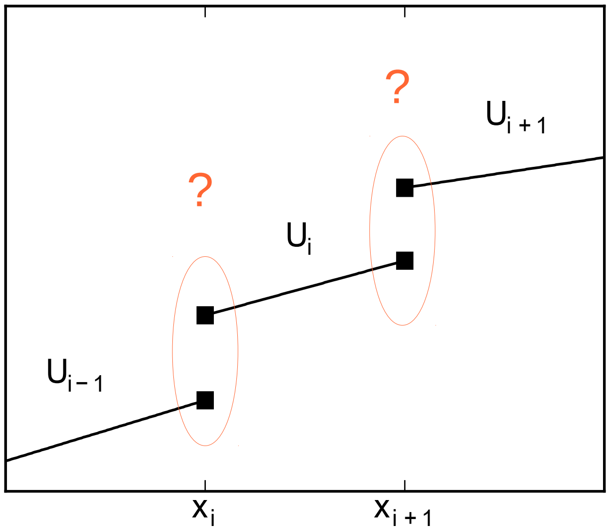
The DG method: 1. Discretization in space

Right hand side:
(without the source f)

$$\int_{\partial D_k} \mu \cdot u_h^k(x,t) \cdot l_j^k(x) \cdot \mathbf{n} \; dx$$

1D: outward pointing normal $\mathbf{n} = \pm 1$

How to calculate the values at the boundary of an element?



⇒ Use of numerical fluxes from the Finite Volume Method!

We introduce the numerical flux $(u_h^k)^*$ that we use instead of the two distinct values at the boundaries:

$$\int_{\partial D_k} \mu \cdot (u_h^k)^* \cdot l_j^k(x) \cdot \mathbf{n} \; dx \overset{\text{1D}}{=} \left[(\mu \cdot u)^* l_j^k(x) \cdot n \right]_{x_l^k}^{x_r^k}$$

The DG method: 1. Discretization in space

The (local) semidiscrete scheme (right hand side like in SE methods)

$$\partial_t u_h^k(t) \cdot M^k - \mu \cdot u_h^k(t) \cdot S^k = -[(\mu \cdot u)^* l_j^k(x) \cdot n]_{x_l^k}^{x_r^k}$$

The DG method: 1. Discretization in space

The (local) semidiscrete scheme (right hand side like in SE methods)

$$\partial_t u_h^k(t) \cdot M^k - \mu \cdot u_h^k(t) \cdot S^k = -[(\mu \cdot u)^* l_j^k(x) \cdot n]_{x_l^k}^{x_r^k}$$

- **Local** mass matrix M and stiffness matrix S

→ **how to calculate M and S?**

The DG method: 1. Discretization in space

The (local) semidiscrete scheme (right hand side like in SE methods)

$$\partial_t u_h^k(t) \cdot M^k - \mu \cdot u_h^k(t) \cdot S^k = -[(\mu \cdot u)^* l_j^k(x) \cdot n]_{x_l^k}^{x_r^k}$$

- **Local** mass matrix M and stiffness matrix S

→ how to calculate M and S?

- Additionally: **flux term** for the boundary

→ how to calculate the flux?

The DG method: 1. Discretization in space

The (local) semidiscrete scheme (right hand side like in SE methods)

$$\partial_t u_h^k(t) \cdot M^k - \mu \cdot u_h^k(t) \cdot S^k = -[(\mu \cdot u)^* l_j^k(x) \cdot n]_{x_l^k}^{x_r^k}$$

- **Local** mass matrix M and stiffness matrix S

→ how to calculate M and S?

2.

- Additionally: **flux term** for the boundary

→ how to calculate the flux?

1.

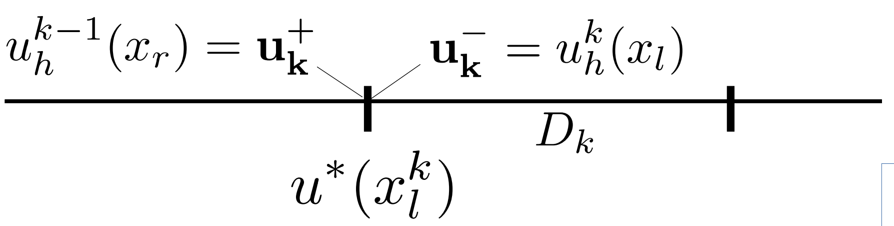
The Discontinuous Galerkin Method

1. The numerical flux

$$\int_{\partial D_k} \mu \cdot (u_h^k)^* \cdot l_j^k(x) \cdot \mathbf{n} \; dx \overset{\text{1D}}{=} \left[(\mu \cdot u)^* l_j^k(x) \cdot n \right]_{x_l^k}^{x_r^k}$$

→ The choice of the numerical flux is the heart of the DG scheme**!**

Basic idea: mimic the dynamic of the wave equation;
the flow of information from one element into the neighbouring element:



- interior information
+ exterior information

1. The numerical flux

Examples:

$$u_h^{k-1}(x_r) = \mathbf{u}_\mathbf{k}^+ \qquad \mathbf{u}_\mathbf{k}^- = u_h^k(x_l)$$

$$D_k$$

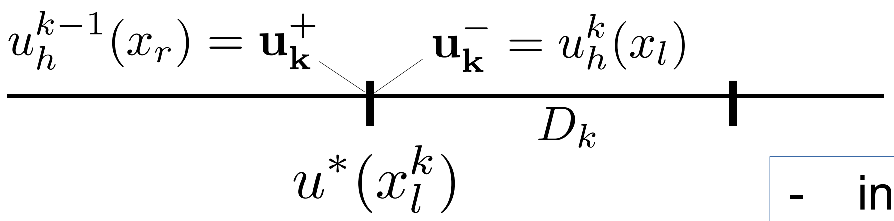
$$u^*(x_l^k)$$

- interior information
+ exterior information

→ easiest choice: **central flux** (average) $u^*(x_{BD}^k) = \frac{1}{2}(u_k^- + u_k^+)$

1. The numerical flux

Examples:



- interior information
+ exterior information

→ easiest choice: **central flux** (average) $u^*(x_{BD}^k) = \frac{1}{2}(u_k^- + u_k^+)$

→ **upwind flux:** always takes the information where it's coming from

$$u^*(x_{BD}^k) = \frac{\mu}{2}(u_k^- + u_k^+) + \frac{|\mu|}{2}(n^- u_k^- + n^+ u_k^+)$$

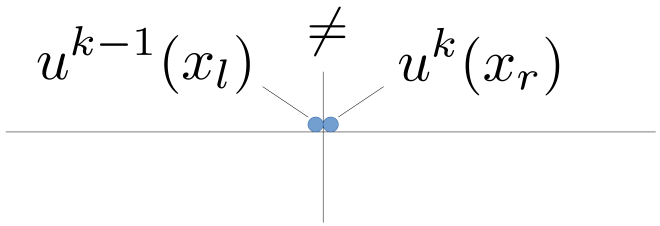
Left boundary element:

$$u^*(x_l^k) = \frac{\mu}{2}(u^k(x_l) + u^{k-1}(x_r)) + \frac{|\mu|}{2}((-1) \cdot u^k(x_l) + (1) \cdot u^{k-1}(x_r))$$

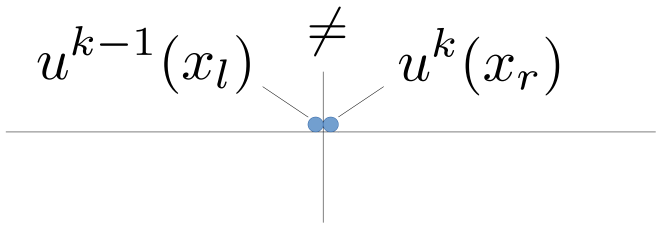
$$= \begin{cases} \mu \; u^k(x_l) & \text{if} \quad \mu \leq 0 \\ \mu \; u^{k-1}(x_r) & \text{if} \quad \mu > 0 \end{cases}$$

wave velocity μ

1. The numerical flux



1. The numerical flux



General concepts:

We want to solve a boundary problem with discontinuous initial conditions
→ **Riemann problem**:

- Flux concepts as a (numerical) solution to Riemann problems
  For example: Godunov flux (like in SeisSol)

➡ Solution is strongly dependent on the choice of flux!

➡ How to find the “best” flux for a problem?

2. Calculation of the mass and stiffness matrix

Local Mass Matrix: $$M_{i,j} = \int_{D_k} \phi_i^k(x) \cdot \phi_j^k(x)\ dx$$

Local Stiffness Matrix: $$K_{i,j} = \int_{D_k} \phi_i^k(x) \cdot \partial_x \phi_j^k(x)\ dx$$

→ Numerical Integration

→ Mapping to a reference element (intervall [-1,1])

→ Basis functions ϕ

2. Caluclation of the mass and stiffness matrix

Step 1: What about the integral?

→ numerical integration: **Gaussian quadrature**

For the Gauß-Lobatto-Legendre points (GLL) x_i and the corresponding weights w_i it holds:

$$\int_{\Omega} f(x)\ dx \approx \sum_{i=1}^{N} w_i\ f(x_i)$$

→ the same points where we define the nodal solution!

2. Caluclation of the Mass and Stiffness matrix

Step 2: Mapping to a reference element, in 1D [-1,1]

Affine mapping $$x(r) = x_l^k + \frac{(1+r)}{2} \cdot dx \qquad r \in [-1,1]$$

Transformation of the integral to a reference interval [-1,1] $$J = \frac{dx}{d\xi}$$

In 1D: $J = \frac{x_r - x_l}{1-(-1)}$

$$\Rightarrow \quad M_{i,j} = \int_{-1}^{1} \phi_i^k(\xi) \cdot \phi_j^k(\xi) J \; d\xi$$

$$K_{i,j} = \int_{-1}^{1} \phi_i^k(\xi) \cdot \partial_x \phi_j^k(\xi) \; d\xi$$

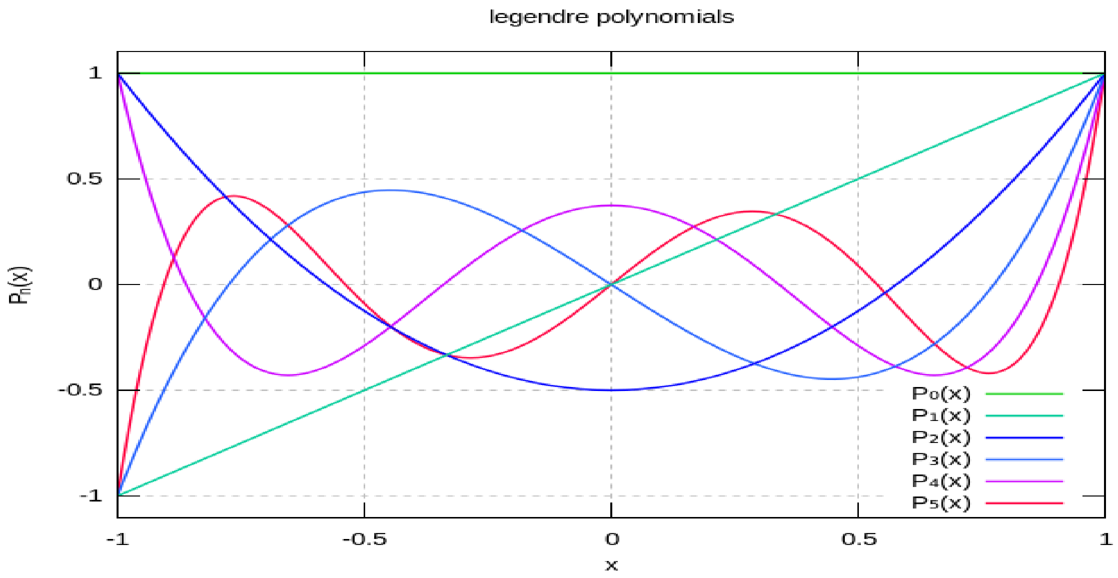
Step 3: basis functions for the modal solution

$$u_h^k(x,t) = \sum_{n=1}^{N_p} u_n^k(t) \cdot P_n(x)$$

Legendre polynomials

$$P_n(x) = \frac{1}{2^n!} \frac{d^n}{dx^n}[(x^2-1)^n]$$

$$\int_{-1}^{1} P_m(x) P_n(x) dx = \frac{2}{2n+1} \delta_{mn}$$



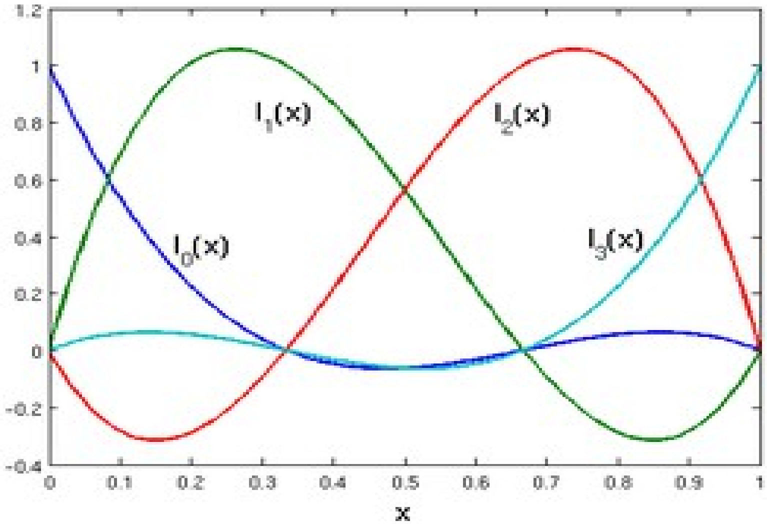
Step 3: basis functions for the nodal solution

$$\boxed{u_h^k(x,t) = \sum_{i=1}^{N_p} u^k(x_i,t) \cdot l_i(x)}$$

Lagrange polynomials

$$L(x_i) = \sum_{j=0}^{k} y_j l_j(x_i) = \sum_{j=0}^{k} y_j \delta_{ij} = y_i$$

$$l_j(x) = \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{x - x_0}{x_j - x_0} \cdot \ldots \cdot \frac{x - x_k}{x_j - x_k}$$



$$\rightarrow \quad l_j(x_i) = \prod_{\substack{m \ne j \\ m=0}} = \delta_{ji}$$

Caluclation of the Mass and Stiffness matrix
(exemplary for the nodal representation)

Caluclation of the Mass and Stiffness matrix (exemplary for the nodal representation)

$$M_{ij}^k = \int_{-1}^{1} l_i^k(\xi) l_j^k(\xi)\ J\ d\xi = \sum_{m=1}^{N_p} w_m\ l_i^k(x_m) l_j^k(x_m)\ J$$

Entries of the local mass matrix

Diagonal matrix!!!

$$= \sum_{m=1}^{N_p} w_m \delta_{im}\ \delta_{jm}\ J = \begin{cases} w_i\ J & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Caluclation of the Mass and Stiffness matrix (exemplary for the nodal representation)

$$M_{ij}^k = \int_{-1}^{1} l_i^k(\xi) l_j^k(\xi)\ J\ d\xi = \sum_{m=1}^{N_p} w_m\ l_i^k(x_m) l_j^k(x_m)\ J$$

Diagonal matrix!!!

Entries of the local mass matrix

$$= \sum_{m=1}^{N_p} w_m \delta_{im}\ \delta_{jm}\ J = \begin{cases} w_i\ J & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$S_{i,j} = \int_{-1}^{1} l_i^k(\xi) \cdot \partial_x l_j^k(\xi)\ d\xi = \sum_{m=1}^{N_p} w_m\ l_i^k(x_m) \cdot \partial_x l_j^k(x_m)$$

$$= \sum_{m=1}^{N_p} w_m \delta_{im} \cdot \partial_x l_j^k(x_m) = w_i \cdot \partial_x l_j^k(x_i)$$

Entries of the local Stiffness matrix

The Discontinuous Galerkin Method

Putting everything together:

The ingredients for the implementation

- **GLL-points** and their corresponding **weights** → Gaussian quadrature
- **Lagrange polynomials** for these points or **Legendre polynomials**
  → Mass matrix (diagonal!) and stiffness matrix
- **first derivative** of the Lagrange polynomials → Stiffness matrix
- **upwind flux** calculations

So we have everything to calculate the ODE for every timestep!

Next: solving the ODE in time

The DG method: 2. Discretization in time

- After discretization in space we have the semidiscrete ODE

$$\frac{du_h^k}{dt} = L_k(t, u_h)$$

with the right hand side

$$L_k(t, u_h) = (M^k)^{-1}(S^k \cdot u_h^k(t) - (F_l^k + F_r^k) \cdot u_h^k(t))$$

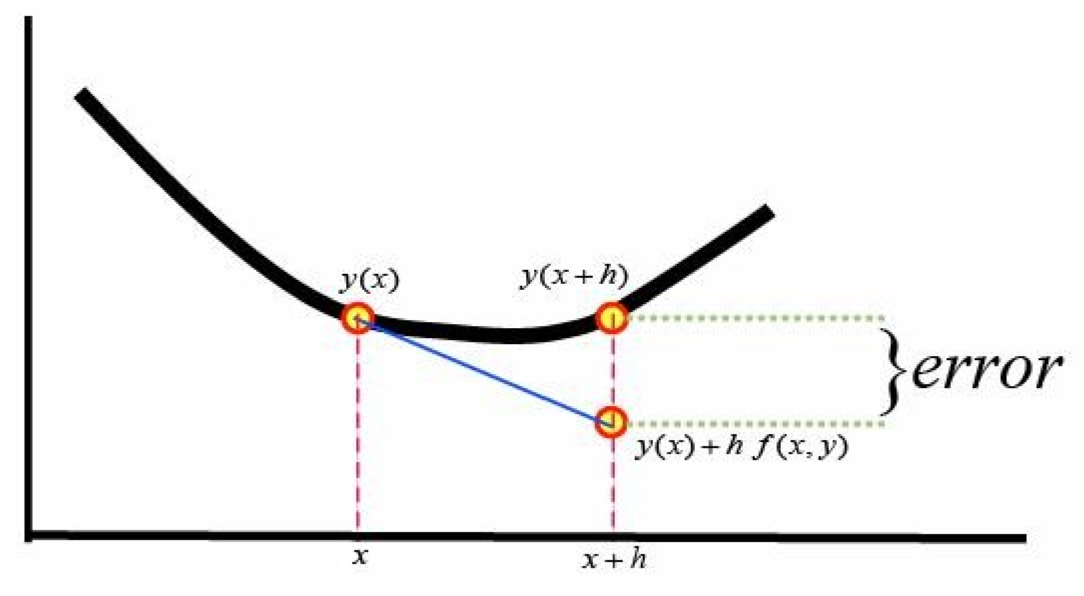
M^k	Mass matrix
S^k	Stiffness matrix
F_l^k	Flux from the left side boundary
F_r^k	Flux from the right side boundary

The DG method: 2. Discretization in time

Different schemes to solve this equation numerically:

1) Euler Method (Finite Difference)

$$u_h(t_{n+1}) = u_h(t_n) + \Delta t \cdot L(t_n, u_h(t_n)) + O(h^2)$$

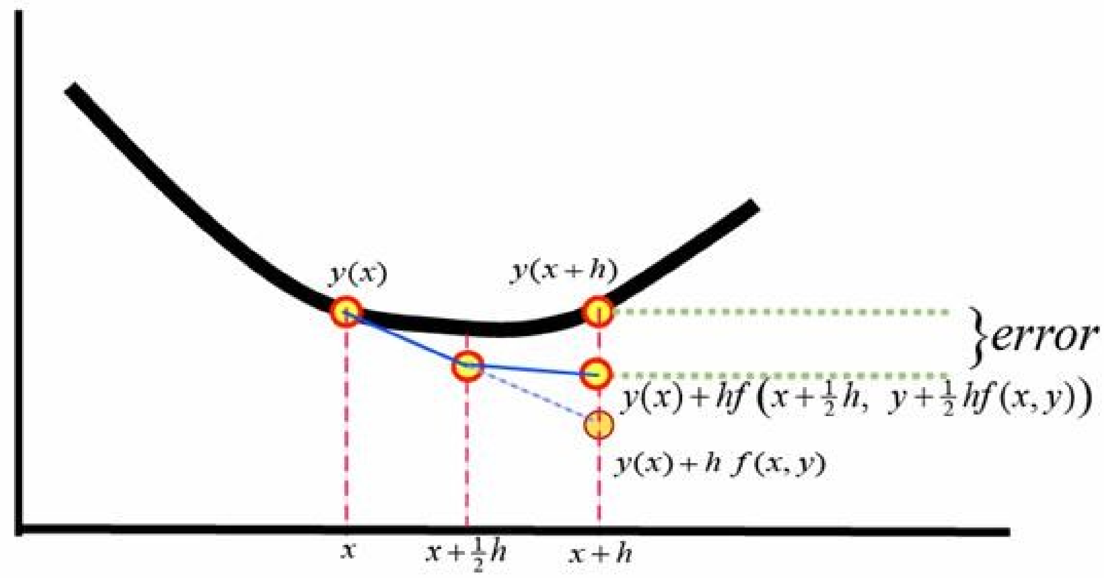


The DG method: 2. Discretization in time

Different schemes to solve this equation numerically:

2) second order Runge -Kutta

$$u_h(t_{n+1}) = u_h(t_n) + \Delta t \cdot L(t_n + \frac{h}{2}, u_h(t_n) + \frac{h}{2} L(t_n, u_h(t_n))) + O(h^3)$$



The DG method: 2. Discretization in time

3) classical Runge -Kutta

- explicite four step Runge-Kutta algorithm

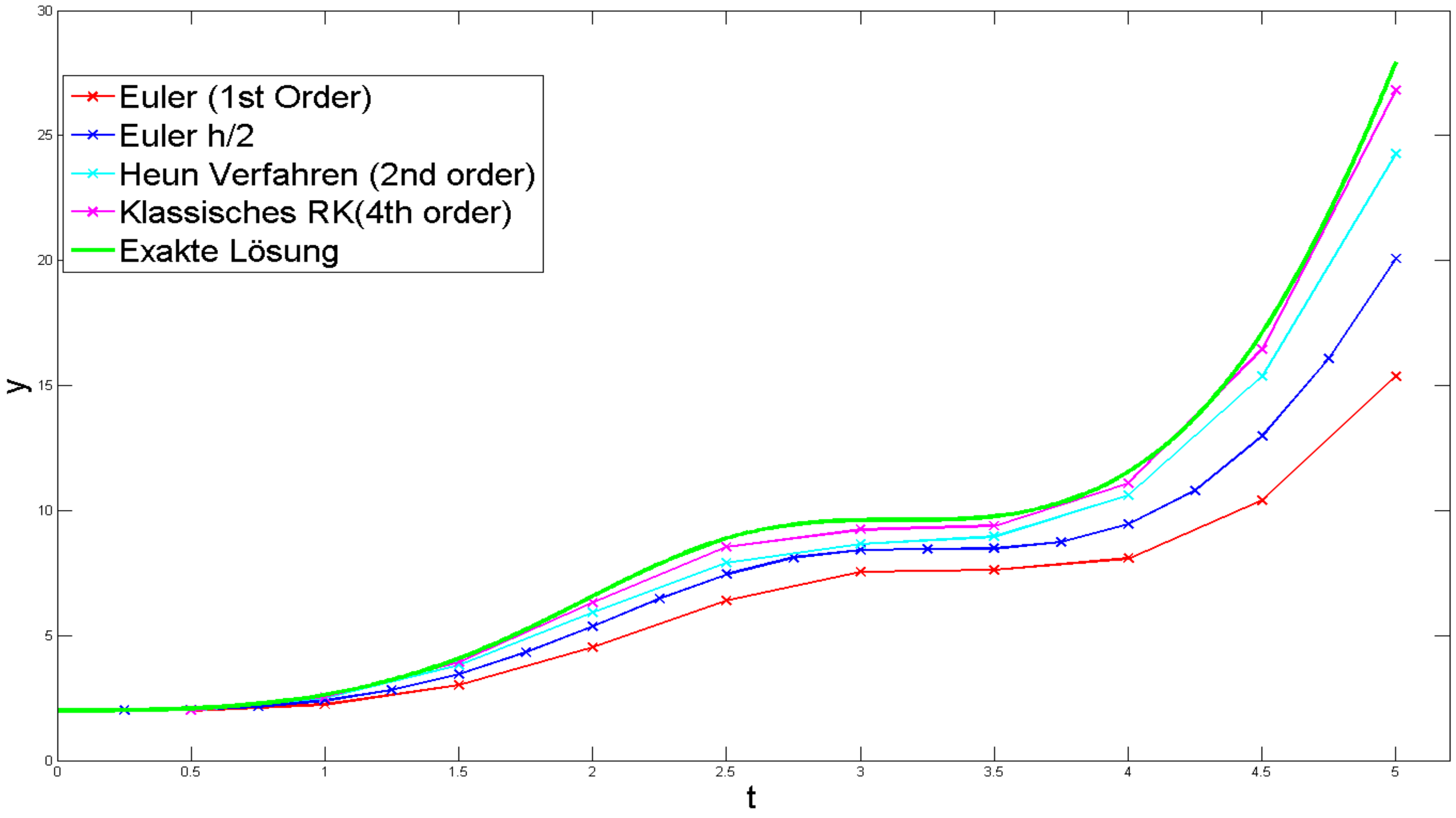
$$k_1 = h \cdot L(t_n, u_h(t_n)) \qquad k_2 = h \cdot L(t_n + \frac{h}{2}, L(t_n) + \frac{k_1}{2})$$

$$k_3 = h \cdot L(t_n + \frac{h}{2}, L(t_n) + \frac{k_2}{2}) \qquad k_4 = h \cdot L(t_n + h, L(t_n) + k_3)$$

$$\boxed{u_h(t_{n+1}) = u_h(t_n) + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)}$$

The DG method: 2. Discretization in time

Solving the first order ODE in time: The Runge-Kutta scheme



Conclusion

Pros:

- complex geometries (tetrahedrals)
- Local operations:
  - Local mass and stiffness matrices
  - Approximation order can change in every element (p-adaptivity)
  - Easy to parallelize
- Energy conservation due the flux concept
- Diagonal mass matrix: explicit time scheme

Cons:

- Large number of degrees of freedom
  - High computational cost

Conclusion

Pros:

- complex geometries (tetrahedrals)
- Local operations:
  - Local mass and stiffness matrices
  - Approximation order can change in every element (p-adaptivity)
  - Easy to parallelize
- Energy conservation due the flux concept
- Diagonal mass matrix: explicit time scheme

Cons:

- Large number of degrees of freedom
  - High computational cost

Essential of every DG scheme: choice of flux and time integration

Outlook/ Next time:

- Nodal vs. modal implementation
- The ADER-DG scheme of SeisSol

Outlook/ Next time:

- Nodal vs. modal implementation
- The ADER-DG scheme of SeisSol

Thank you for your attention!