Isogeometric Analysis for Compressible Flows with Application in Turbomachinery

MSc Thesis Defense

Andrzej Jaeschke

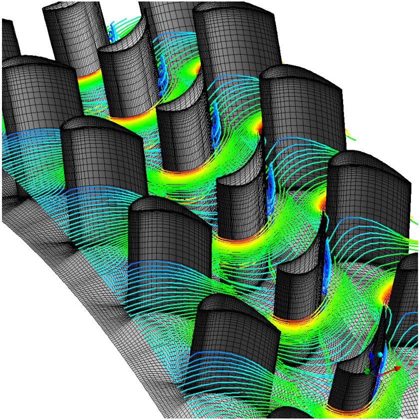
August 31, 2015

TU Delft
Delft University of Technology

Overview

Overview

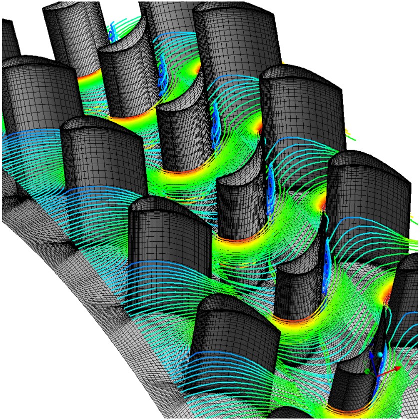
Turbomachinery



Source: http://www.pointwise.com/

- Complex (smooth) geometry
- Structural, heat and flow analysis
- Hard to optimize using engineering experience

Turbomachinery



Source: http://www.pointwise.com/

- Complex (smooth) geometry
- Structural, heat and flow analysis
- Hard to optimize using engineering experience

Used in many applications:

- automobile
- aerospace
- energy production
- heating

Why to optimize?

- Lower emissions
- Lower fuel consumption
- Longer life-cycle

Why to optimize?

- Decrease mass
- Decrease flow losses
- Increase TET by improving cooling
- Evaluate particle impact on the blade

Why IGA?

- No manual re-meshing - automation of optimization possible
- **Idea:** Use the same function space to represent the geometry and to solve the problem
- Represents the geometry exactly
- The same numerical method for fluid and structural simulation
- No gaps between domains

Objectives of the thesis

Main goal:

- Develop a solver for the compressible Euler equations

Side goals:

- Apply the AFC stabilization in the IGA framework
- Develop the constrained L2 projection in the IGA framework

Schedule of the thesis

Literature study:

- Implement the B-spline constructor and evaluator
- Implement the IGA solver for Poisson equation
- Extend it to arbitrary 2D geometry

Schedule of the thesis

Literature study:

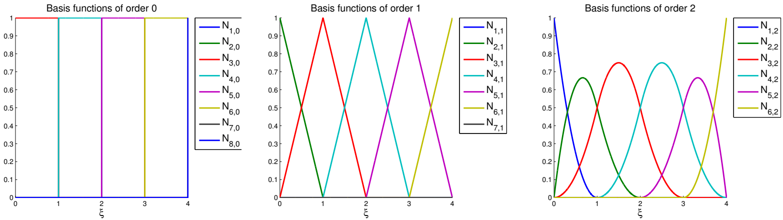
- Implement the B-spline constructor and evaluator
- Implement the IGA solver for Poisson equation
- Extend it to arbitrary 2D geometry

Main part of thesis project:

- Implement the IGA solver for stationary convection-diffusion equation
- Extend it to time-dependent problems
- **Implement the Compressible Euler equations solver**

Overview

B-spline basis functions on uniform knot vector



$$\xi = [0, 0, 0, 1, 2, 3, 4, 4, 4]$$

The higher order B-spline basis functions are computed with **recursive Cox-de Boor formula**.

B-spline basis functions on nonuniform knot vector

Quadratic functions on
$\xi = [0, 0, 0, 1, 2, 2, 3, 4, 5, 5, 6, 6, 6]$

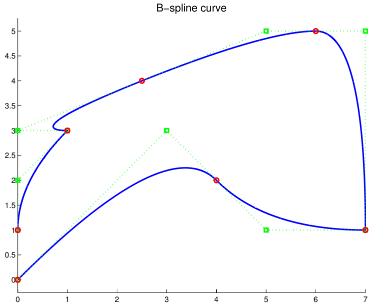
Properties:

- Positivity
- Partition of unity
- Compact support (at most $p + 1$ knot spans)
- C^{p-m} continuity at knots

B-spline curves

B-spline curves

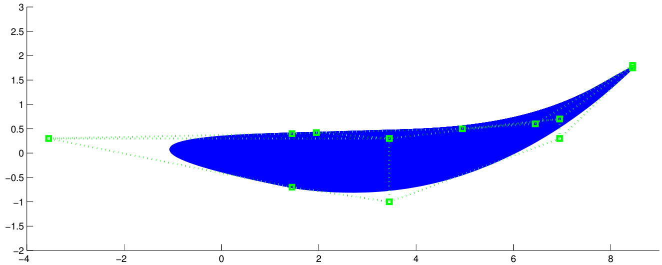
$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B_i}$$



B-spline surfaces

B-spline surfaces

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B_{i,j}}$$



Enriching the basis

Shape preserving operations:

- Inserting new knots (equivalent to **h-refinement**)
- Increasing the order of the basis (equivalent to **p-refinement**) results in $n-1$ new DOFs
- **k-refinement** - firstly increase the order and then add new knot values - No counterpart in standard FEM

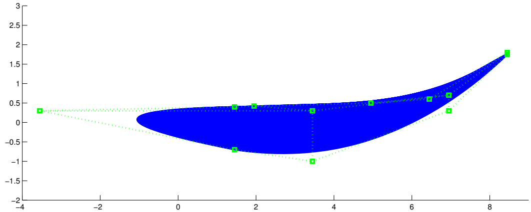
Result: Generalized tensor product basis

Poisson's problem on arbitrary geometry

Problem description

$$-\Delta u(x,y) = f(x,y) \qquad \text{in } \Omega,$$
$$u(x,y) = 0 \qquad \text{on } \partial\Omega$$

with load vector $f(\xi,\eta) = 2\pi^2 \sin(\pi\xi)\sin(\pi\eta)$.



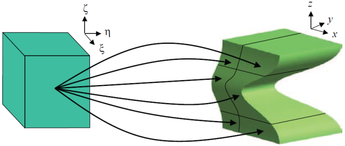
Poisson's problem on arbitrary geometry

Main steps of solving the problem:

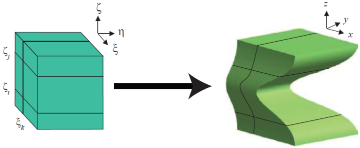
1. Derive the weak formulation taking into account **mapping** of parametric domain onto physical domain
2. Use the B-spline tensor product basis function as the basis functions for analysis (careful indexing)
3. Write the problem in the form of linear system
4. Evaluate the basis functions at points required for integration
5. Evaluate the **mapping matrix function**
6. Integrate numerically and assemble the matrix and RHS
7. Solve the linear system

Parametric and physical domains

FEA:

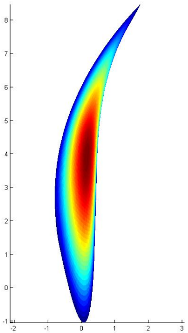


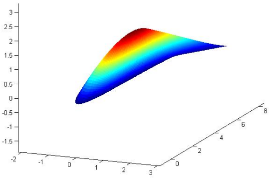
IGA:



Source: J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, "Isogeometric Analysis: Towards Integration of CAD and FEA"

Poisson's problem on arbitrary geometry

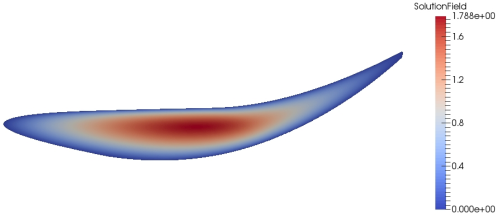




Uniform open knot vectors, $n = 4$, $m = 3$, $p = q = 2$

Poisson's problem on arbitrary geometry

- The Matlab code written during the literature study was not efficient enough.
- It was decided to use the C++ library **G+SMO** for further work



Overview

Framework of AFC

High-order method

- **Accurate** in smooth regions
- Producing **oscillations** in vicinity of discontinuities and steep gradients

Low-order method

- Guaranteeing **no oscillations** in vicinity of discontinuities and steep gradients
- **Overly diffusive**

In framework of AFC low-order method is obtained from high-order method by **algebraic operations** on matrices.

Framework of AFC

Anti-diffusion

Difference between high-order and low-order schemes

Idea: Add limited amount of anti-diffusion to low-order scheme in non-linear fashion

Flux limiting

- Decompose the anti-diffusion into fluxes between the nodes
- Limit them separately using available algorithms
- Compose the anti-diffusion back from the individual fluxes

Challenge: Extend flux limiting to non-nodal DOFs!

Overview

Standard L2 projection

For IGA Nodal assignment of initial and boundary conditions is **NOT** possible.

Standard L2 projection

Find such a projection of analytical f to V^h that the residual

$$R(Pf) = Pf - f$$

is orthogonal to V^h. In other words:

$$(Pf - f, v) = 0, \quad \forall v \in V^h$$

$$(f, g) = \int_\Omega f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$$

Standard L2 projection

$$\text{Find } f^h \in V^h \text{ such that for all } v^h \in V^h, \ (f^h, v^h) = (f, v^h).$$

We can rewrite the problem in the matrix-vector form:

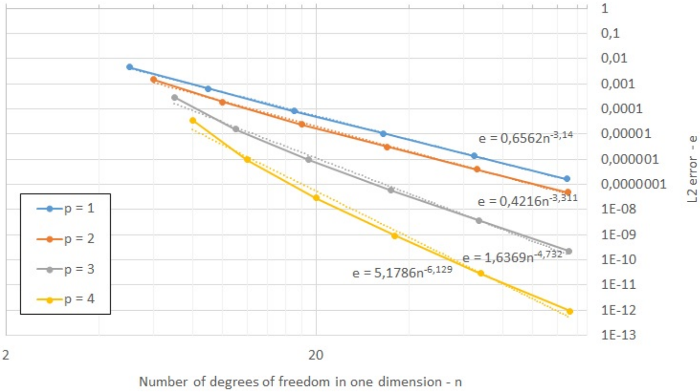
$$M_C \mathbf{x} = \mathbf{b}$$

$M_C = \{m_{ij}\}$ - the **consistent mass matrix**; $m_{ij} = \int_\Omega \varphi_i \varphi_j d\mathbf{x}$

$\mathbf{b} = \{b_i\}$ - the right hand side vector; $b_i = \int_\Omega f \varphi_i d\mathbf{x}$.

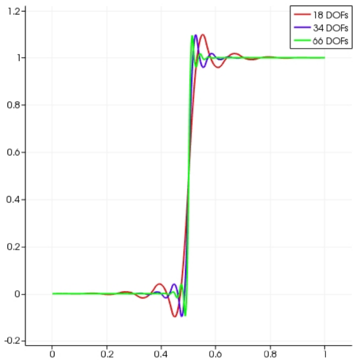
Standard L2 projection

$$f(\mathbf{x}) = \sin(\pi x_1)\sin(\pi x_2) \text{ in } \Omega = [0,1] \times [0,1]$$

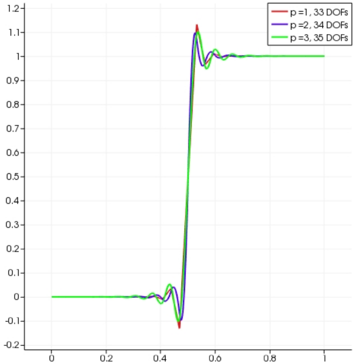


Standard L2 projection

$$f(x) = \begin{cases} 0 & \text{if } x < 0.5 \\ 1 & \text{if } x \geq 0.5 \end{cases} \quad \text{in } \Omega = [0, 1]$$



(a) $p = 2$ and varying number of DOFs



(b) varying p and equivalent number of DOFs

Constrained L2 projection

Idea: Use approach similar to AFC.

High-order method

Standard L2 projection

$$M_C \mathbf{x}^H = \mathbf{b}$$

Low-order method

$$M_L \mathbf{x}^L = \mathbf{b}$$

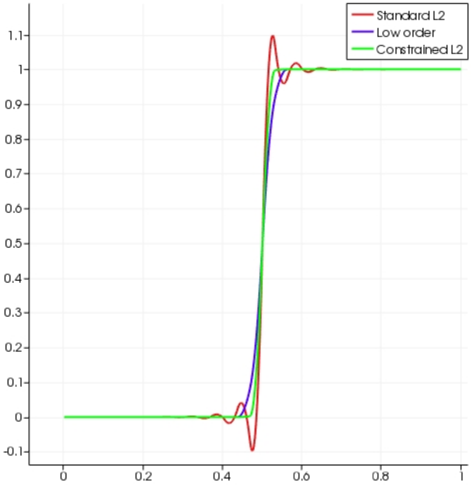
M_L - result of row-sum mass lumping of M_C

Resulting method

$$M_L \mathbf{x} = M_L \mathbf{x}^L + \bar{\mathbf{f}}(\mathbf{x}^H)$$

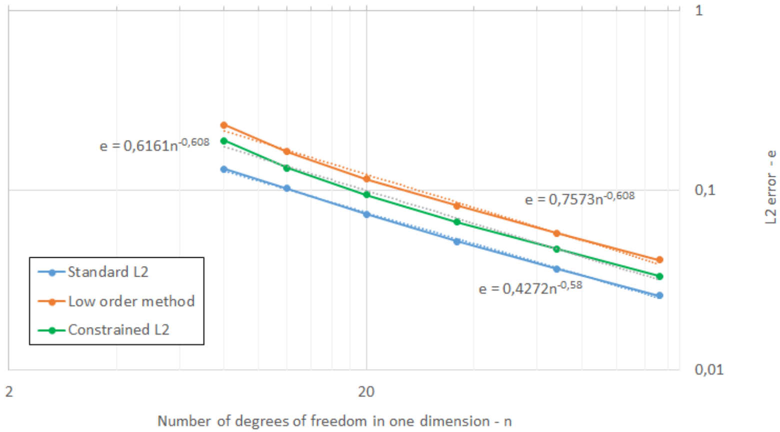
$\bar{\mathbf{f}}(\mathbf{x}^H)$ - limited anti-diffusion

Constrained L2 projection



$p = 2, \ n = 34$

Constrained L2 projection



Overview

Stationary convection-diffusion equation

Definition of problem

$$-\nabla \cdot (D \nabla u(\mathbf{x})) + \nabla \cdot (\mathbf{v}(\mathbf{x}) u(\mathbf{x})) = R(\mathbf{x}) \quad \text{in } \Omega$$

$$u(\mathbf{x}) = \gamma(\mathbf{x}) \;\; \text{on } \mathit{\Gamma}_D$$
$$\frac{du}{d\mathbf{n}}(\mathbf{x}) = \beta(\mathbf{x}) \;\; \text{on } \mathit{\Gamma}_N$$

$u(\mathbf{x})$ - the variable of interest
D - the diffusion tensor (or scalar coefficient d)
$\mathbf{v}(\mathbf{x})$ - the average velocity of quantity
$R(\mathbf{x})$ - the source term.

Stationary convection-diffusion equation

Dicrete problem

$$(S - K)\mathbf{u} = \mathbf{r}$$

$S = \{s_{ij}\}$ - **discrete diffusion operator**

$$s_{ij} = \int_\Omega (D\nabla\varphi_j \cdot \nabla\varphi_i) d\mathbf{x}$$

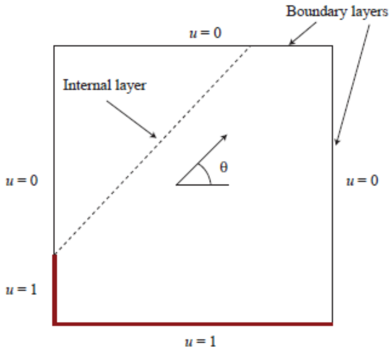
$K = \{k_{ij}\}$ - **discrete convection operator**

$$k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \quad \mathbf{c}_{ij} = \int_\Omega \nabla\varphi_j \varphi_i d\mathbf{x}$$

$\mathbf{r}$ - right-hand side vector

$$r_i = \int_\Omega R\varphi_i d\mathbf{x} + \int_{\Gamma_N} D\beta\varphi_i ds$$

Benchmark problem



$|\mathbf{v}| = 1$

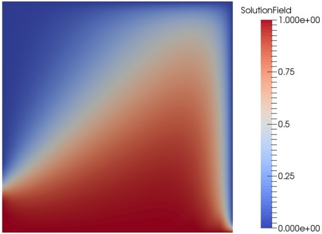
Element Peclet number

$$Pe_h = \frac{|\mathbf{v}|h}{2d}$$

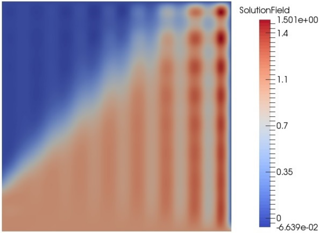
$Pe_h > 1$ convection-dominated
$Pe_h < 1$ diffusion-dominated
(on the length scale of mesh)

Source: J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, "Isogeometric Analysis: Towards Integration of CAD and FEA"

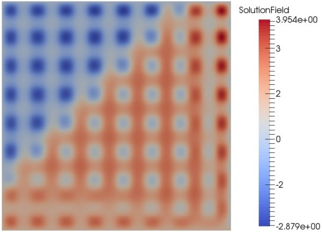
Benchmark problem on $p = 2$, 18×18 B-spline basis



(a) $d = 0.1,\ Pe_h = 0.56$



(b) $d = 0.01,\ Pe_h = 5.56$



(c) $d = 0.001,\ Pe_h = 55.56$

Algebraic flux correction

Remedy: Use AFC stabilization.

High-order method

$$(S-K)\mathbf{u} = \mathbf{r}$$

Low-order method

$$(S-L)\mathbf{u} = \mathbf{r}$$

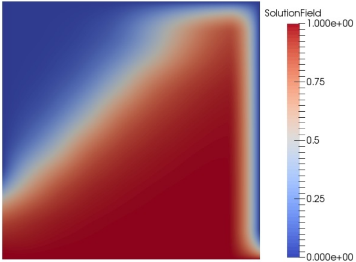
$L = K + D$ - result of adding the artificial diffusion to K

Resulting method

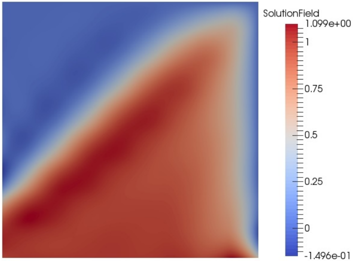
$$(S-L)\mathbf{u} = \mathbf{r} + \bar{\mathbf{f}}(\mathbf{u})$$

$\bar{\mathbf{f}}(\mathbf{u})$ - limited anti-diffusion

AFC vs. SUPG



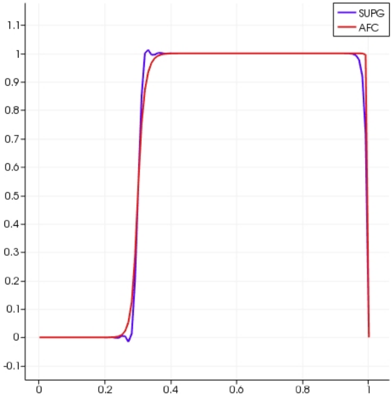
(a) AFC



(b) SUPG

$d = 0.0001$, uniform $p = 2$, 10×10 B-spline basis

AFC vs. SUPG



$d = 0.0001$, uniform $p = 2$, 130×130 B-spline basis

Estimated L2 convergence rates:

p	AFC	SUPG
1	1.71	0.69
2	1.03	0.40
3	0.71	0.47
4	0.48	0.50

Time-dependent convection-diffusion equation

Definition of problem

$$\frac{\partial u(\mathbf{x},t)}{\partial t} - \nabla \cdot (D \nabla u(\mathbf{x},t)) + \nabla \cdot (\mathbf{v}(\mathbf{x},t) u(\mathbf{x},t)) = R(\mathbf{x},t) \quad \text{in } \Omega$$

$$u(\mathbf{x},t) = \gamma(\mathbf{x},t) \;\; \text{on } \mathit{\Gamma}_D$$
$$\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x},t) = \beta(\mathbf{x},t) \;\; \text{on } \mathit{\Gamma}_N$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega$$

$u(\mathbf{x},t)$ - the variable of interest
D - the diffusion tensor (or scalar coefficient d)
$\mathbf{v}(\mathbf{x},t)$ - average velocity field
$R(\mathbf{x},t)$ - source term

Time-dependent convection-diffusion equation

Semi-dicrete problem

$$M_C \frac{d\mathbf{u}}{dt} = (K - S)\mathbf{u} + \mathbf{r}$$

$M_C = \{m_{ij}\}$ - **consistent mass matrix**

$$m_{ij} = \int_\Omega \varphi_j \varphi_i d\mathbf{x}$$

Time discretization using:

- Forward Euler method
- SSP-eRK-2 method
- SSP-eRK-3 method

Algebraic flux correction

High-order method

$$M_C \frac{d\mathbf{u}}{dt} = (K - S)\mathbf{u} + \mathbf{r}$$

Low-order method

$$M_L \frac{d\mathbf{u}}{dt} = (L - S)\mathbf{u} + \mathbf{r}$$

M_L - result of row-sum mass lumping of M_C

$L = K + D$ - result of adding the artificial diffusion to K

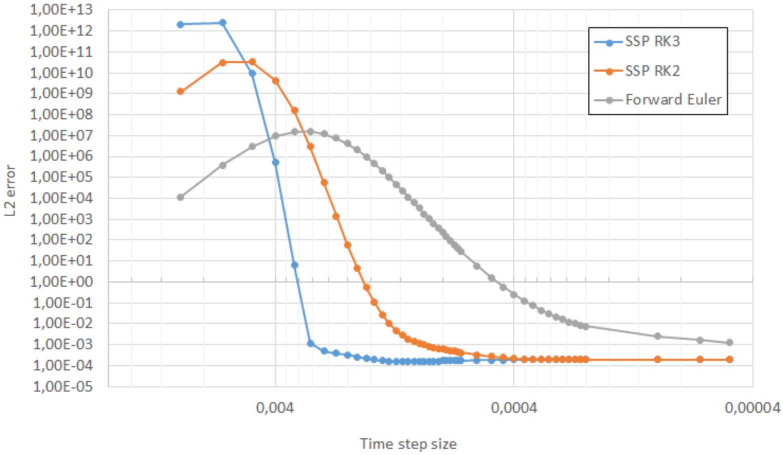
Resulting method

$$M_L \frac{d\mathbf{u}}{dt} = (L - S)\mathbf{u} + \mathbf{r} + \bar{\mathbf{f}}(\mathbf{u})$$

$\bar{\mathbf{f}}(\mathbf{u})$ - limited anti-diffusion

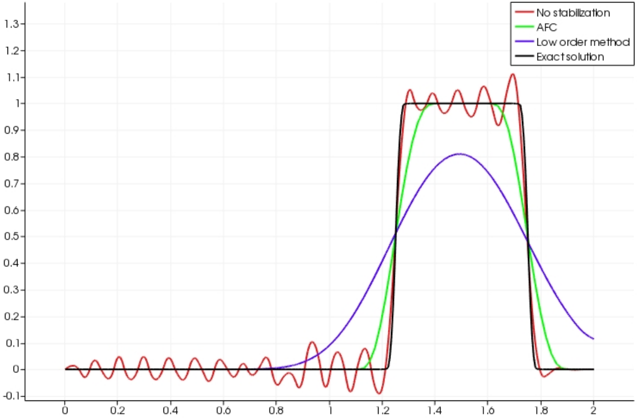
Convection of smooth hump - AFC disabled

Convection of smooth hump - AFC disabled



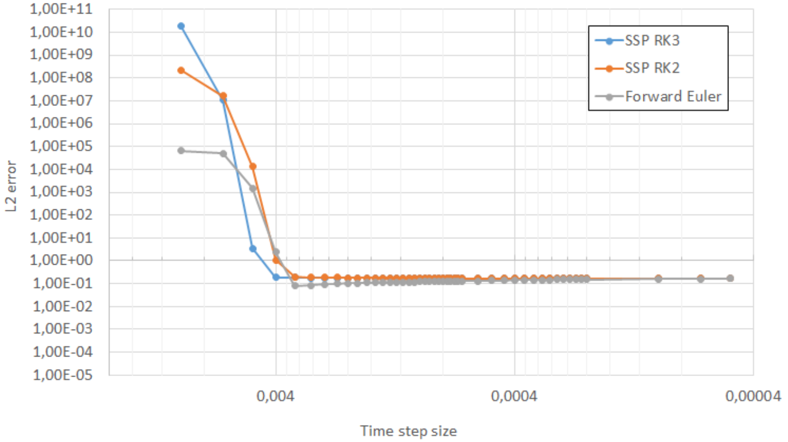
uniform $p = 2$, 66×66 B-spline basis

Convection of rectangular wave



uniform $p = 2$, 66×66 B-spline basis
SSP-eRK-3, $\Delta t = 0.0001$

Convection of rectangular wave - AFC enabled



uniform $p = 2$, 66×66 B-spline basis

Compressible Euler equations

Problem in divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0$$

vector of **conservative variables**:

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}$$

vector of **inviscid fluxes**:

$$\mathbf{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p\mathscr{I} \\ \rho E \mathbf{v} + p \mathbf{v} \end{bmatrix}$$

Compressible Euler equations

Semi-discrete problem

$$M_C \frac{dU}{dt} = KU + S(U)$$

$$U = \begin{bmatrix} \rho_1 & (\rho \mathbf{v})_1 & (\rho E)_1 & \cdots & \rho_N & (\rho \mathbf{v})_N & (\rho E)_N \end{bmatrix}^T$$

M - the **block consistent mass matrix**

$$M_{ij} = m_{ij} I, \qquad m_{ij} = \int_\Omega \varphi_i \varphi_j d\mathbf{x}$$

K - the **discrete Jacobian operator**

$$K_{ij} = \mathbf{c}_{ji} \cdot \mathbf{A}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j d\mathbf{x}$$

S - the **boundary load vector**

$$S_i = -\int_{\Gamma_N} \varphi_i F_n ds$$

Algebraic flux correction

High-order method

$$M_C \frac{dU}{dt} = KU + S(U)$$

Low-order method

$$M_L \frac{dU}{dt} = LU + S(U)$$

M_L - result of row-sum mass lumping of M_C

$L = K + D$ - result of adding the artificial diffusion to K

Resulting method

$$M_L \frac{dU}{dt} = LU + S(U) + \bar{F}(U)$$

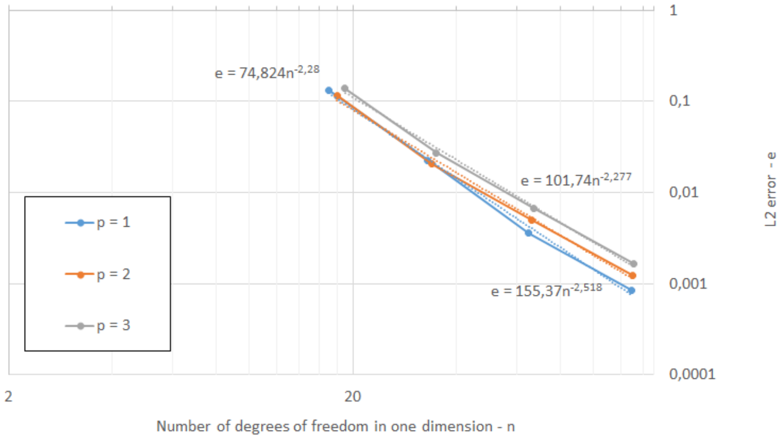
$\bar{F}(U)$ - limited anti-diffusion

Limiting in terms of **primitive variables**

Convection of isentropic vortex - AFC disabled

Density ρ

Convection of isentropic vortex - AFC disabled



SSP-eRK-3, $\Delta t = 0.005$

Sod's shock tube

Density

Velocity

Pressure

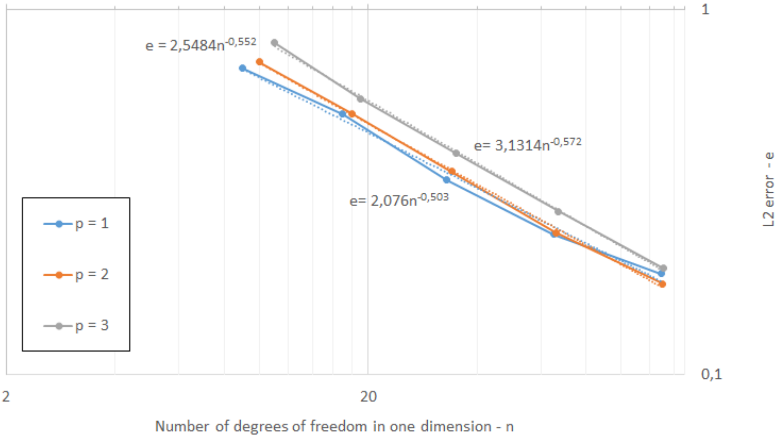
Sod's shock tube

No stabilization

AFC

ρ, uniform $p = 1$, 129×129 B-spline basis, SSP-eRK-3 $\Delta t = 0.001$

Sod's shock tube - AFC enabled



ρ at $t = 0.231$, SSP-eRK-3 $\Delta t = 0.001$

Overview

Outcome from this thesis project

- Pilot implementation of IGA-based compressible Euler solver
- Implementation of AFC generalized to non-nodal DOFs
- Implementation of constrained L2 projection generalized to non-nodal DOFs
- Directions of further development of IGA based approach to compressible flow problems

Outcome from this thesis project

- Pilot implementation of IGA-based compressible Euler solver
- Implementation of AFC generalized to non-nodal DOFs
- Implementation of constrained L2 projection generalized to non-nodal DOFs
- Directions of further development of IGA based approach to compressible flow problems

For author:

- Understanding of B-splines and IGA
- Experience with G+SMO and templated C++
- Knowledge in field of compressible inviscid flows
- Understanding of FCT, TVD and AFC frameworks

Future developments

- Full Navier-Stokes solver
- Optimization of code
- Multi-patch problems
- 3D problems
- NURBS

Future developments

- Full Navier-Stokes solver
- Optimization of code
- Multi-patch problems
- 3D problems
- NURBS
- Non-linear FCT or better linearisation
- Alternative time-discretization schemes
- Investigation of limitation of convergence rates for Euler equation (by 0.5 and 2)

Thank you for your attention!

Full text of the thesis: