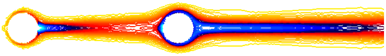
Isogeometric Residual Distribution Scheme

Algiane Froehly
Rémi Abgrall Cécile Dobrzynski

INRIA

Introduction

Problematic



Subsonic flow around 2 cylindres: spurious entropy production

Solution

Isogeometric analysis

Euler Equations

Conservative form

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\vec{\mathbf{u}} \\ \rho E \end{pmatrix}, \begin{cases} \overrightarrow{\nabla}.[\rho\vec{\mathbf{u}}] & = 0 \quad \textbf{Conservation of mass} \\ \overrightarrow{\nabla}.[\rho u_i\vec{\mathbf{u}} + p\delta_i] & = 0 \quad \textbf{Conservation of momentum} \\ \overrightarrow{\nabla}.[(\rho E + p)\vec{\mathbf{u}}] & = 0 \quad \textbf{Conservation of Energy} \end{cases}$$

Euler Equations

Conservative form

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\vec{\mathbf{u}} \\ \rho E \end{pmatrix}, \begin{cases} \vec{\nabla}.[\rho\vec{\mathbf{u}}] & = 0 \quad \textbf{Conservation of mass} \\ \vec{\nabla}.[\rho u_i \vec{\mathbf{u}} + p\delta_i] & = 0 \quad \textbf{Conservation of momentum} \\ \vec{\nabla}.[(\rho E + p)\vec{\mathbf{u}}] & = 0 \quad \textbf{Conservation of Energy} \end{cases}$$

Problem to solve

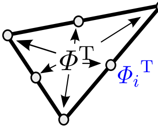
$$\begin{cases} \vec{\nabla}.\vec{\mathcal{F}}(\mathbf{U}) = 0 & \forall x \in \Omega \\ \text{weak boundary conditions} & (\mathcal{BC}) \end{cases}$$

$\mathcal{RDS}$

Definitions

- Φ^{T}: total residual over T

$$\Phi^{\mathrm{T}} = \int_{\partial \mathrm{T}} \overrightarrow{\mathcal{F}}(\mathbf{U}^h)\cdot \vec{\mathbf{n}} d\partial \mathrm{T}$$



- Φ_i^{T}: local nodal residual

$$\sum_{i \in T} \Phi_i^{\mathrm{T}} = \Phi^{\mathrm{T}}$$

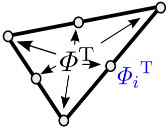
$$\Phi_i^{\mathrm{T}} = \text{ High-order term } + \text{ Stabilization term}$$

$\mathcal{RDS}$

Definitions

- Φ^{T}: total residual over T

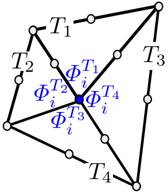
$$\Phi^{\mathrm{T}} = \int_{\partial \mathrm{T}} \overrightarrow{\mathcal{F}}(\mathbf{U}^h) \cdot \vec{\mathbf{n}} d\partial \mathrm{T}$$



- Φ_i^{T}: local nodal residual

$$\sum_{i \in T} \Phi_i^{\mathrm{T}} = \Phi^{\mathrm{T}}$$

$$\Phi_i^{\mathrm{T}} = \text{ High-order term } + \text{ Stabilization term}$$



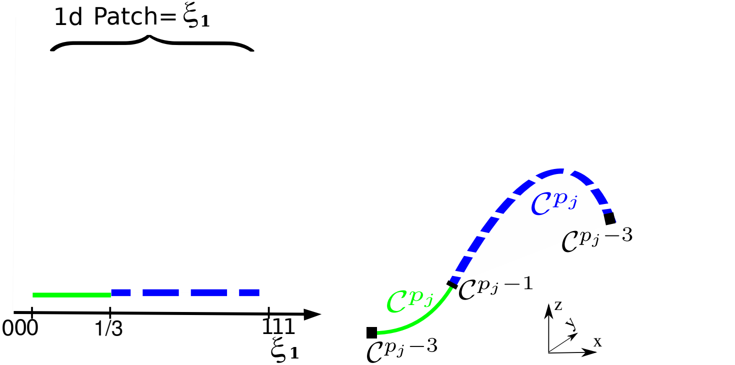
$\mathcal{RDS}$ formulation

$$\forall M_i, \sum_{T \ni M_i} \Phi_i^{\mathrm{T}}(\mathbf{U}^h) = 0$$

$\mathcal{N}$on $\mathcal{U}$niform $\mathcal{R}$ational $\mathcal{B}$-$\mathcal{S}$pline Definitions

B-Splines basis functions in $\mathbb{R}^d$ defined by :

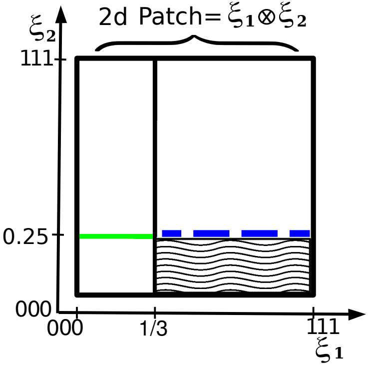
- their polynomial order in the j^{th} dimension: p_j
- d knot vectors $(\xi_j)_{j=1,d}$



$\mathcal{N}$on $\mathcal{U}$niform $\mathcal{R}$ational $\mathcal{B}$-$\mathcal{S}$pline Definitions

B-Splines basis functions in $\mathbb{R}^d$ defined by :

- their polynomial order in the j^{th} dimension: p_j
- d knot vectors $(\xi_j)_{j=1,d}$



$\mathcal{N}$on $\mathcal{U}$niform $\mathcal{R}$ational $\mathcal{B}$-$\mathcal{S}$pline Definitions

B-Splines basis functions in $\mathbb{R}^d$ defined by :

- their polynomial order in the j^{th} dimension: p_j
- d knot vectors $(\xi_j)_{j=1,d}$

$\mathcal{NURBS}$ basis functions in $\mathbb{R}^d$ defined by :

- The n_j B-Splines basis functions in the j^{th} dimension
- A set of n_j weighting factors: W_l

B-Splines and $\mathcal{NURBS}$ basis functions for $d = 1$

i^{th} $\mathcal{NURBS}$ function of order k:

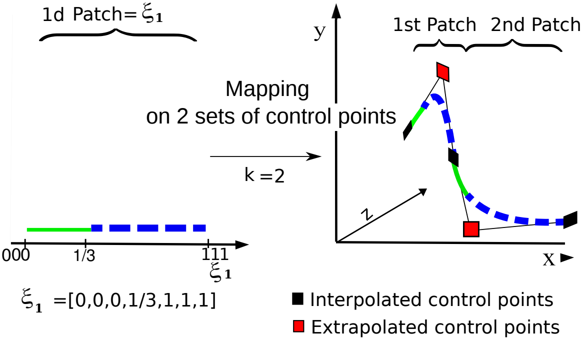
$$M_{i,k}(\xi) = \frac{W_i N_{i,k}(\xi)}{\sum_{l=1,n} W_l N_{l,k}(\xi)}$$

$\mathcal{N}$on $\mathcal{U}$niform $\mathcal{R}$ational $\mathcal{B}$-$\mathcal{S}$pline Definitions

$\mathcal{NURBS}$ curves and surfaces extrapolation

$$C(\xi) = \sum_{i=1,n} \tilde{P}_i M_{i,k}(\xi); \quad S(\xi,\eta) = \sum_{i=1,n\times m} \tilde{P}_i \Psi_{i,k}(\xi,\eta)$$

Mapping $\mathcal{NURBS}$



$\mathcal{NURBS}$ adaptation for $\mathcal{RDS}$

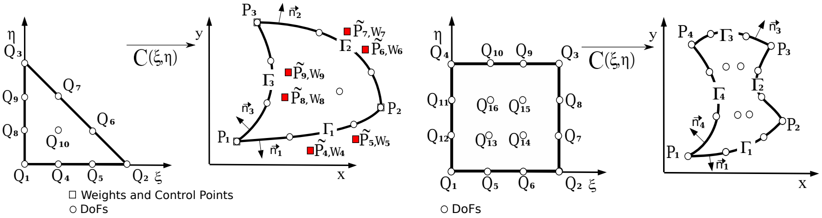
Numerical simplifications and basis functions used

- Same nodal vector for all edges: $\xi = [\overbrace{0, 0, \cdots,}^{k \text{ times}}\overbrace{\cdots, 1, 1}^{k \text{ times}}]$
- One unique weight per point
- Definition of B-Spline-like basis for triangular elements : $\begin{pmatrix} k \\ j1, j2, j3 \end{pmatrix} \xi^{j1}\eta^{j2}\zeta^{j3}, \; j \in [|1, k|], \; j1 + j2 + j3 = k$

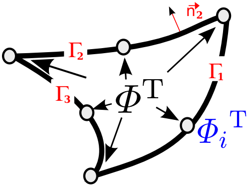
$\mathcal{NURBS}$ adaptation for $\mathcal{RDS}$

Numerical simplifications and basis functions used

- Same nodal vector for all edges: $\xi = [\overbrace{0, 0, \cdots,}^{k \text{ times}} \overbrace{\cdots, 1, 1}^{k \text{ times}}]$
- One unique weight per point
- Definition of B-Spline-like basis for triangular elements



Residuals computation



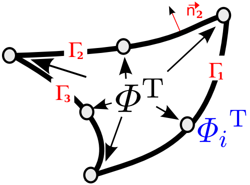
Total residual computation

$$\Phi^T = \int_{\partial T} \vec{\mathcal{F}}_h . \vec{\mathbf{n}} \, d\partial T = \sum_{j=1,n_e} \int_{\Gamma_j} \left(\sum_{l=1}^{n_{\text{DoFs}}} \tilde{\vec{\mathcal{F}}}_l \Psi_l(\xi, \eta) \right) . \vec{\mathbf{n}}_j(x, y) d\Gamma_j$$

3-points Gauss-Legendre quadrature at order 3

6-points Gauss-Legendre quadrature at order 4

Residuals computation

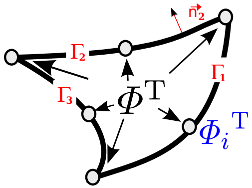


Stabilization term computation

$$D_i^T = \int_T (\vec{\lambda}.\overrightarrow{\nabla\Psi_i})\bar{\tau}(\vec{\lambda}.\overrightarrow{\nabla \mathbf{U}_h})\, dT$$

7-points Gauss-Legendre quadrature at order 3
13-points Gauss-Legendre quadrature at order 4

Slip wall Boundary Condition: $\vec{\mathbf{u}}.\vec{\mathbf{n}} = \vec{\mathbf{0}}$



Weak formulation at the i^{th} boundary node

$$\Phi_{b,i,\Gamma} = \int_\Gamma \Psi_i^T \vec{\mathcal{F}}_{\text{slip}}(\mathbf{U}_h^n, \vec{\mathbf{n}}) d\Gamma$$

$$\Phi_{b,i,\Gamma} = \sum_{q=1}^{n_{quad}} \left(\tilde{\omega}_q \Psi_i^T(\xi_q) \vec{\mathcal{F}}_{\text{slip}}(\mathbf{U}_h^n(\xi_q), \vec{\mathbf{n}}(\xi_q)) \|\Gamma\| \right)$$

$(k-1)^2$-points Newton-Cotes quadrature at order k

$\mathcal{NURBS}$ Mesh generation

Problematic

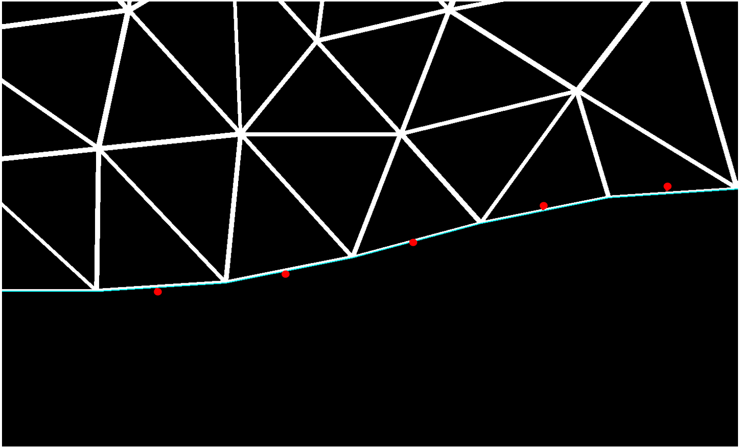
- $\mathcal{NURBS}$ mesh needed to use isogeometric analysis

Solution

- Classical meshing of the geometry (piecewise linear)
- Computing the control points and weights needed to extrapolate a $\mathcal{NURBS}$ curve verifying that :
  - $\mathcal{G}_0$ continuity is preserved (the curve interpolate each $P1$ point)
  - Each boundary edge is approached by a portion of circle

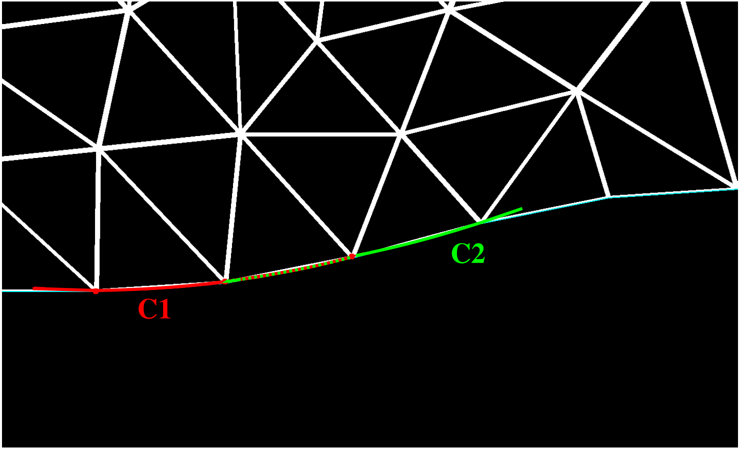
$\mathcal{NURBS}$ Mesh generation

From classical mesh to $\mathcal{NURBS}$ mesh



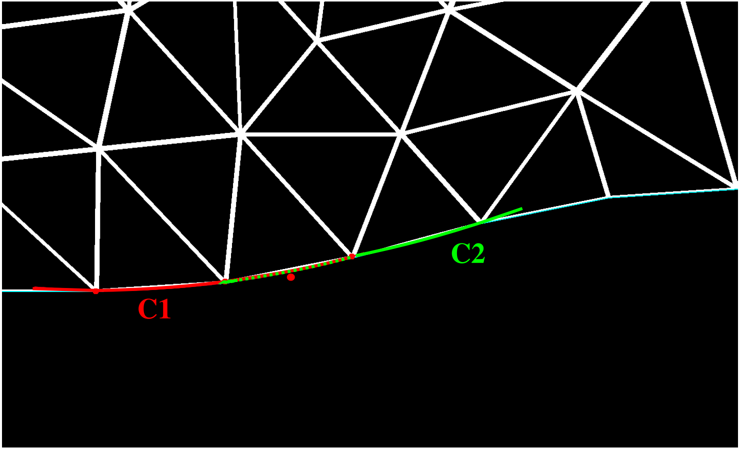
$\mathcal{NURBS}$ Mesh generation

From classical mesh to $\mathcal{NURBS}$ mesh



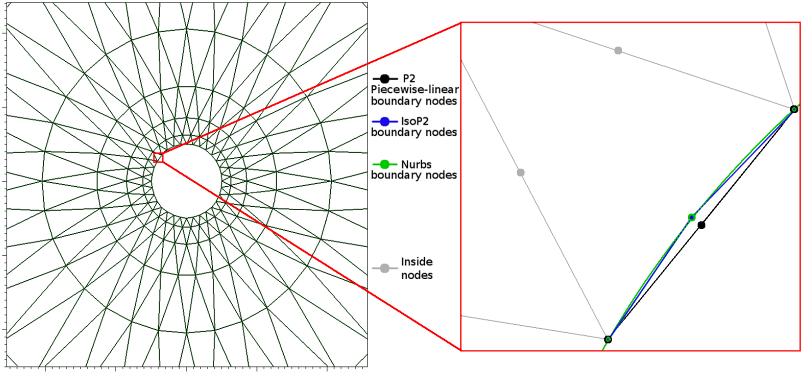
$\mathcal{NURBS}$ Mesh generation

From classical mesh to $\mathcal{NURBS}$ mesh



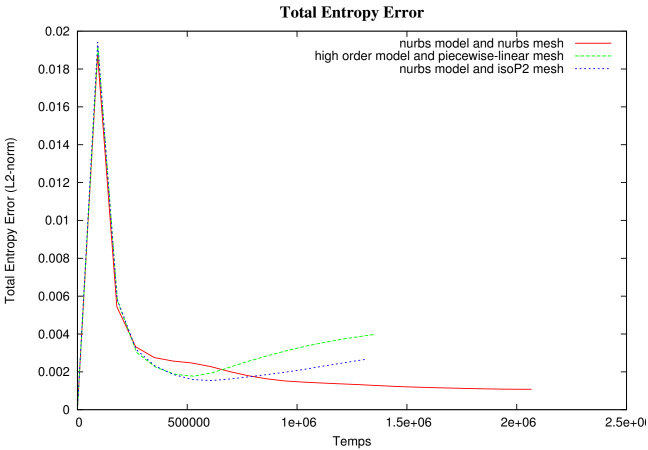
Subsonic flow around a cylindre: $Ma = 0.38$

Boundaries representation



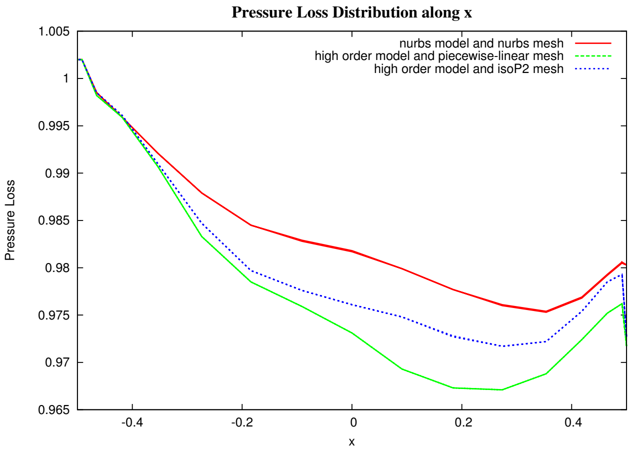
P2 picewise-linear mesh, IsoP2 mesh and Nurbs mesh

Subsonic flow around a cylindre: $Ma = 0.38$



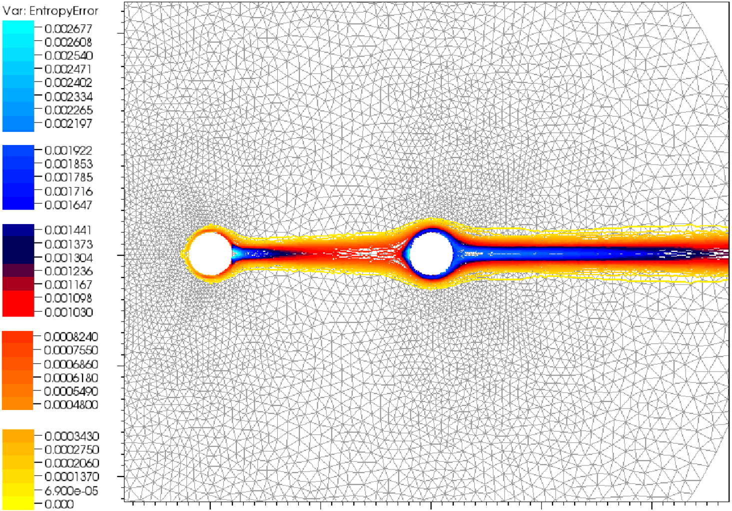
L2-entropy error during iterations for 8×32 nodes meshes

Subsonic flow around a cylindre: $Ma = 0.38$



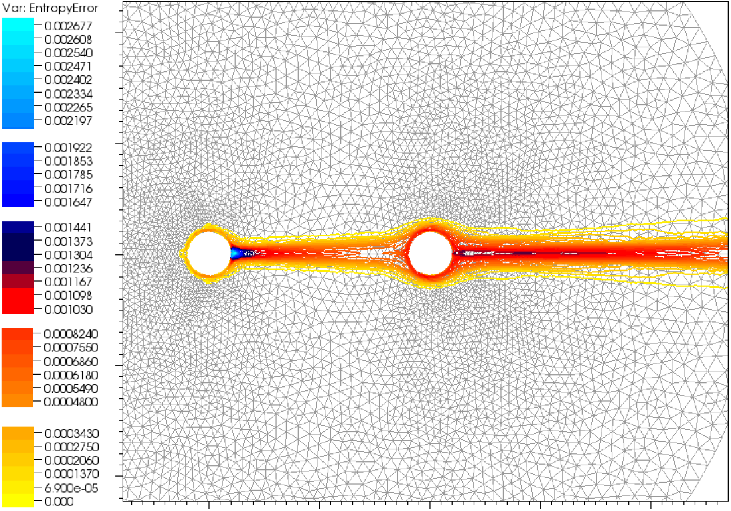
Pressure loss distribution along cylindre for 8×32 nodes meshes

Subsonic flow around 2 cylindres: $Ma = 0.38$



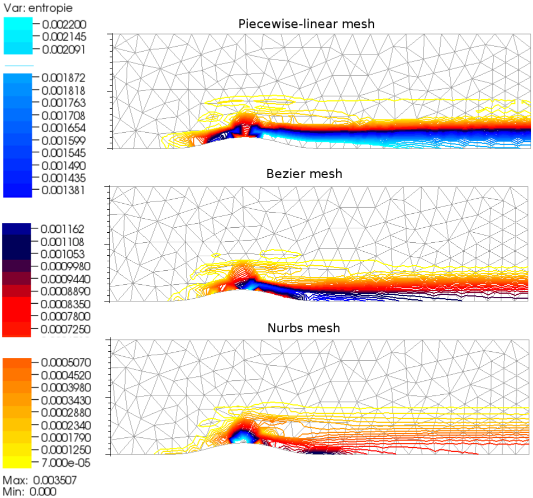
Entropy error

Subsonic flow around 2 cylindres: $Ma = 0.38$



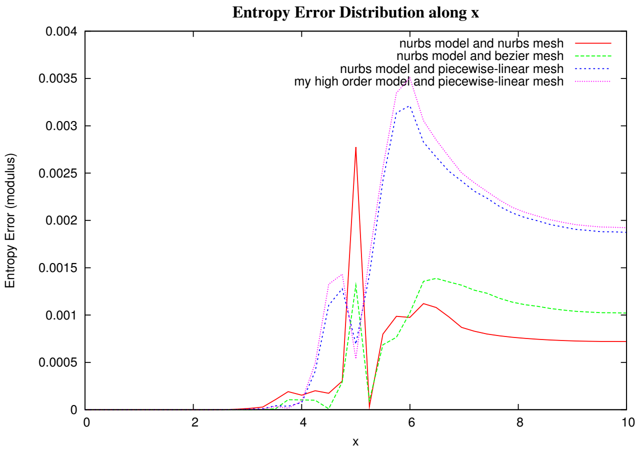
Entropy error

Subsonic flow over a bump: $Ma = 0.63$, slope=26%



Entropy error on the whole bump meshes

Subsonic flow over a bump: $Ma = 0.63$, slope=26%



L2-Entropy error along the bump floor

Conclusion

Done work

- $\mathcal{RDS}$ Isogeometric of order 3 and 4 for triangular, quadrangular and hybrid $2D$-meshes
- $\mathcal{RDS}$ Isogeometric of order 3 for tetrahedral $3D$-mesh
- Automatic generation of $2D$ $\mathcal{NURBS}$ meshes

Perspectives

- To adapt Isogeometric analysis to Navier-Stokes equations
- Automatic generation of $3D$ $\mathcal{NURBS}$ meshes