A NURBS-based Discontinuous Galerkin Framework for Compressible Aerodynamics

Stefano Pezzano, Régis Duvigneau

Université Côte d'Azur, Inria, CNRS, LJAD
e-mail: stefano.pezzano@inria.fr, regis.duvigneau@inria.fr

AIAA Aviation Forum, 15-19 June 2020

Outline

Outline

1. **Introduction to IsoGeometric Analysis**
2. NURBS-Based Discontinuous Galerkin
3. Extension to deformable domains
4. ALE-AMR coupling

The IsoGeometric paradigm

- Industrial drawing is done using CAD software
- Meshes for numerical simulation are generated from CAD data
- Classical finite elements adopt a different geometric representation with respect to CAD

- Conversion between the two formats is time consuming and the process is not completely automatic

- **IsoGeometric Analysis**: CAD basis functions (NURBS) are employed as approximation space for finite elements

Bernstein polynomials

The building blocks of the CAD representation are the Bernstein polynomials:

$$B_i^p(\xi) = \binom{p}{i} \xi^i (1-\xi)^{p-i} \quad \xi \in [0,1]$$

Some important properties:

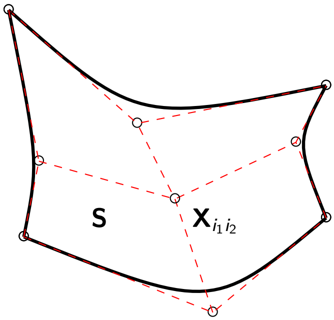
- non-negativity: $B(\xi) \geq 0, \ \forall \xi$
- partition of unity: $\sum_i B_i^p(\xi) = 1, \ \forall \xi$
- $B_0^p(0) = B_p^p(1) = 1$
- they can be computed recursively

Bézier surfaces

Parametric polynomial surfaces defined as:

$$\mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} B_{i_1}^{p}(\xi)\, B_{i_2}^{p}(\eta)\, \mathbf{X}_{i_1 i_2}$$

The coefficients $\mathbf{X}_{i_1 i_2}$ are called control points.



Conics are not exactly represented with polynomials, rational Bézier surfaces are therefore introduced:

$$R_{i_1 i_2}^{p}(\xi,\eta) = \frac{B_{i_1}^{p}(\xi)\, B_{i_2}^{p}(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{p+1}\sum_{j_2=1}^{p+1} B_{j_1}^{p}(\xi)\, B_{j_2}^{p}(\eta)\, \omega_{j_1 j_2}} \qquad \mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} R_{i_1 i_2}^{p}(\xi,\eta)\, \mathbf{X}_{i_1 i_2}$$

The coefficients $\omega_{i_1 i_2}$ are called weights.

B-Splines and NURBS

- complex geometries require high-degree functions when using a single polynomial patch
- B-Spline functions $N_i^p(\xi)$ are the piecewise extension of Bernstein polynomials
- Parametric domain $\hat{\Omega} = [\xi_1, \xi_l]$, discretized by the knot vector $\Xi = (\xi_1, ..., \xi_i, ..., \xi_l)$
- Recursive evaluation:

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi)$$

- NURBS are the rational extension of B-Splines:

$$R_{i_1 i_2}^p(\xi, \eta) = \frac{N_{i_1}^p(\xi)\, N_{i_2}^p(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} N_{j_1}^p(\xi)\, N_{j_2}^p(\eta)\, \omega_{j_1 j_2}}$$

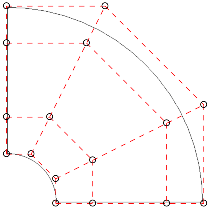
Outline

Bézier Extraction

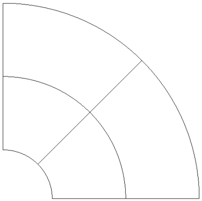
- NURBS suitable for CG (classic IgA)
- Rational Bézier functions are DG-compliant

- Bézier patches can be extracted from NURBS
- Extraction based on multiple knot refinements
- Geometry is unaltered

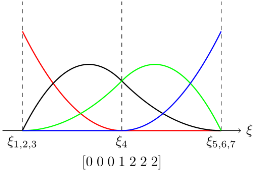
CAD geometry now compatible with DG discretization!

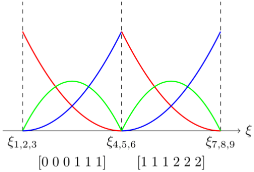


NURBS patch
(4×4, degree 2)



Four disconnected rational Bézier patches





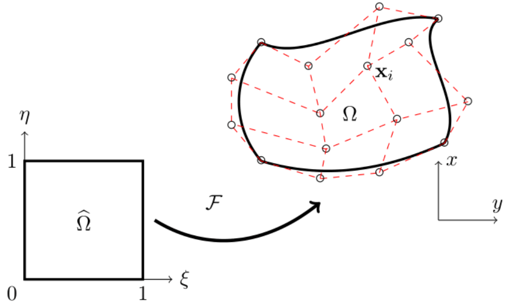
DG formulation

- Navier-Stokes equations in divergence form:

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

- Each element is a rational Bézier patch:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{w}_h \end{pmatrix} = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta) \begin{pmatrix} \mathbf{x}_i \\ \mathbf{w}_i \end{pmatrix}$$



- The weak formulation is:

$$\frac{\mathrm{d}\mathbf{w}_i}{\mathrm{d}t} \int_{\hat{\Omega}_j} R_k R_i \, |J_\Omega| \, \mathrm{d}\hat{\Omega} = \int_{\hat{\Omega}_j} \nabla R_k \cdot \mathbf{F} \, |J_\Omega| \, \mathrm{d}\hat{\Omega} - \oint_{\partial \hat{\Omega}_j} R_k \mathbf{F}^* \, |J_\Gamma| \, \mathrm{d}\hat{\Gamma}$$

- Integrals computed in the parametric domain, J_Ω and J_Γ are metric terms

DG formulation, cont'd

- Elements coupled through numerical flux $\mathbf{F}^*$
- $\mathbf{F}^* = \mathbf{F}^*(\mathbf{w}_h^+, \mathbf{w}_h^-, \mathbf{n})$ is a consistent Riemann solver:

$$\mathbf{F}^*(\mathbf{w}_0, \mathbf{w}_0, \mathbf{n}) = \mathbf{F}(\mathbf{w}_0) \cdot \mathbf{n}$$

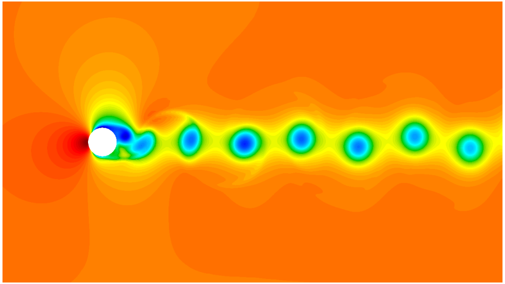
- Space integrals computed through Gauss quadrature
- Time evolution of DOFs described by system of ODEs:

$$\mathcal{M} \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = \mathcal{R}(\mathbf{w}_h)$$

- Explicit Runge-Kutta (RK4 or RK3 SSP) method for time integration

2D Laminar Cylinder

- Exact cylinder representation, using rational functions
- Polynomial degree: 3, 4, 5
- 3 refinement levels: 1065, 2145 and 4455 elements
- $M_\infty = 0.2$, $Re = 500$



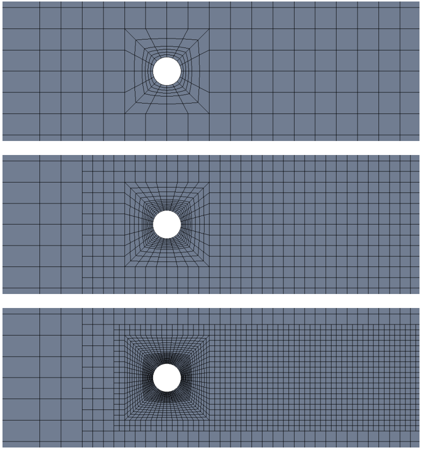
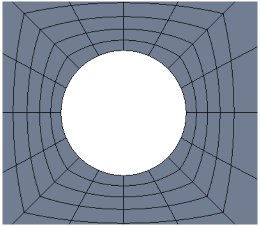
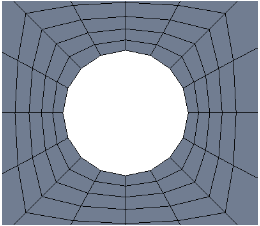


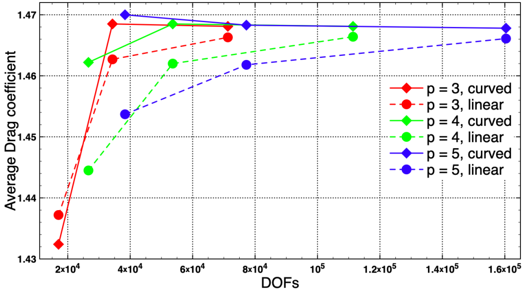
Figure: mesh levels

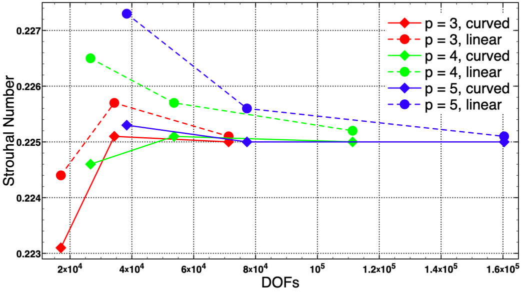
Mesh convergence

- Comparison with linear grid
- Faster convergence with degree 4 and 5, with curved boundary
- Lower convergence rate with linear geometry









Outline

ALE scheme

- The formulation proposed by Nguyen[1] is extended to Bézier elements:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{w}_i \int_{\hat{\Omega}_j} R_k R_i \left|J_\Omega\right| \mathrm{d}\hat{\Omega}\right) = \int_{\hat{\Omega}_j} \nabla R_k \cdot \left(\mathbf{F} - \mathbf{V}_g \mathbf{w}_h\right) \left|J_\Omega\right| \mathrm{d}\hat{\Omega} - \oint_{\partial\hat{\Omega}_j} R_k \mathbf{F}^*_{ale} \left|J_\Gamma\right| \mathrm{d}\hat{\Gamma}$$

- Consistency condition for $\mathbf{F}^*_{ale} = \mathbf{F}^*_{ale}(w_h^+, w_h^-, \mathbf{V}_g, \mathbf{n})$ becomes:

$$\mathbf{F}^*_{ale}\left(\mathbf{w}_0, \mathbf{w}_0, \mathbf{V}_g, \mathbf{n}\right) = \mathbf{F}\left(\mathbf{w}_0\right) \cdot \mathbf{n} - \left(\mathbf{V}_g \cdot \mathbf{n}\right)\mathbf{w}_0$$

- Constant solutions not exactly preserved due to metric terms
- Mass matrix is time dependent:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathcal{M}\mathbf{w}\right) = \mathcal{R}\left(\mathbf{w}_h, \mathbf{V}_g\right)$$

[1] V. T. Nguyen, *An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries*, Journal of Fluids and Structures 26 (2010), 312-329

NURBS-based mesh movement

- Isoparametric paradigm used to define grid velocity field:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{V}_g \\ \mathbf{w}_h \end{pmatrix} = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta) \begin{pmatrix} \mathbf{x}_i \\ \mathbf{v}_{g,i} \\ \mathbf{w}_i \end{pmatrix}$$

- Time evolution of control point net:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_{g,i}$$

- Arbitrarily high-order deformations
- Explicit movement: distribution of $\mathbf{v}_{g,i}$ is imposed at each time step
- Integration with RK4 or RK3 SSP

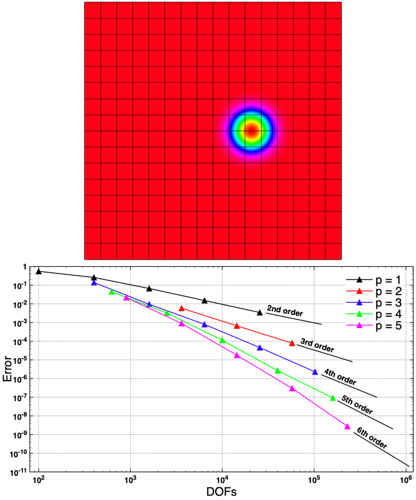
Isentropic vortex test case

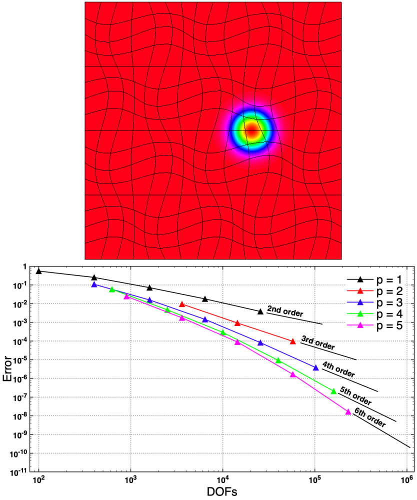
- Euler equations
- Advection of an isentropic vortex:

$$
\begin{cases}
\rho = \left(1 - \frac{\gamma-1}{16\gamma\pi^2}\beta^2 e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}} \\
u = 1 - \beta e^{1-r^2} \frac{y-y_0}{2\pi} \\
v = \beta e^{1-r^2} \frac{x-t-x_0}{2\pi} \\
p = \rho^\gamma
\end{cases}
$$

- Two configurations are compared:
  - fixed mesh
  - deforming mesh: $u_g(\mathbf{x}, t) = v_g(\mathbf{x}, t) = \sin\left(\frac{\pi}{2}x\right)\sin\left(\frac{\pi}{2}y\right)\sin(2\pi t)$

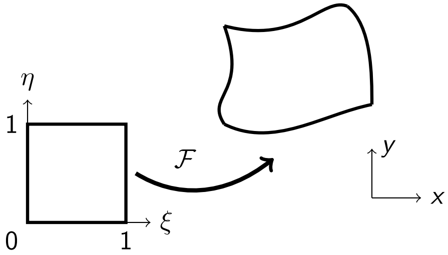
Error analysis

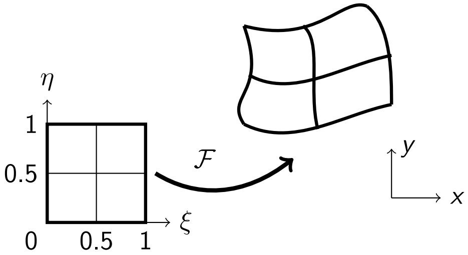




Outline

Adaptive Mesh Refinement



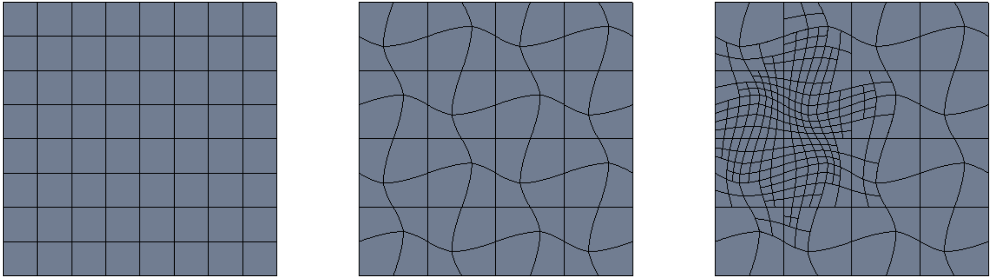


- Isotropic quadtree-like refinement
- Element and solution splitting based on knot insertion
- Very coarse, but CAD-consistent, initial meshes
- Error indicator based on interface jumps:

$$\varepsilon_j = \sum_{k \in \mathcal{N}_j} \int_{\Gamma_{jk}} \| \mathbf{W}|^j - \mathbf{W}|^k \| \, d\Gamma$$

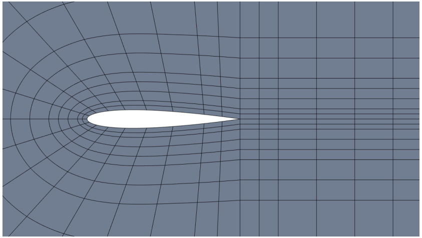
Coupling with mesh movement

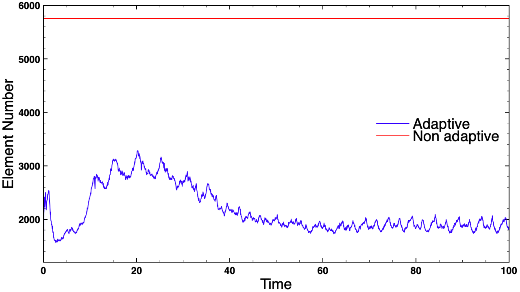
- Moving and deforming bodies = unsteady flows
- Non-linear deformations can make refinement irreversible
- Mesh movement must be AMR-compatible
- Mesh velocity is computed on Level-0 grid and propagated via knot insertion

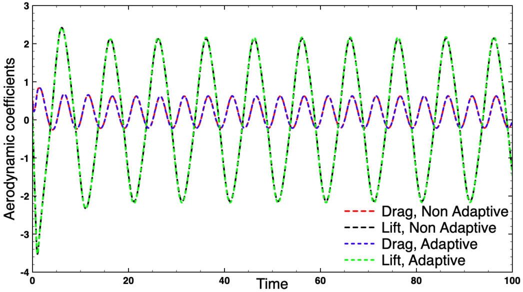


Laminar Pitching Airfoil

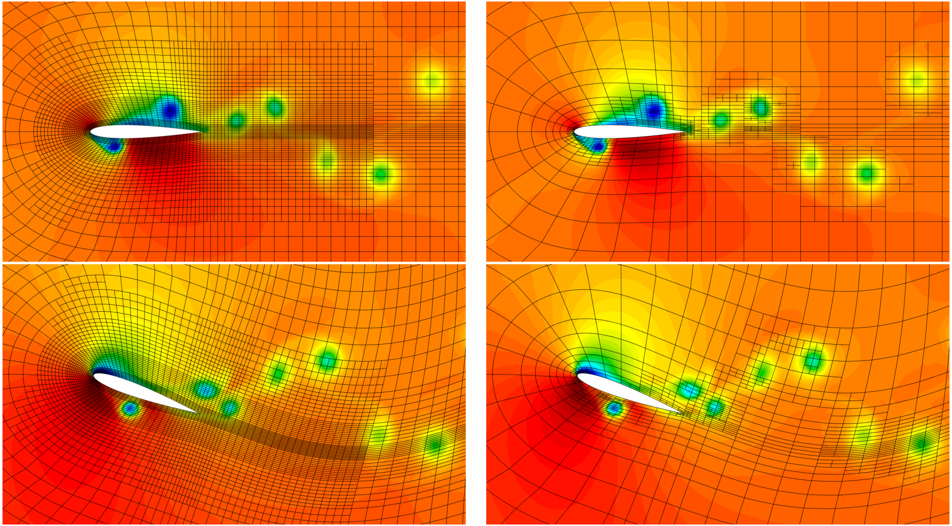
- $M_\infty = 0.2,\ Re = 1000$
- $\Delta\alpha = 20^\circ,\ k = 0.25$
- Separated flow with complex wake pattern
- Level 0 mesh (right): 1248 elements
- 2 level adaptation
- Non-adaptive mesh: 5756 elements





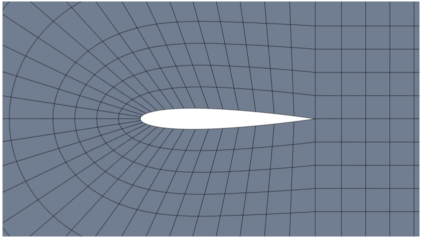


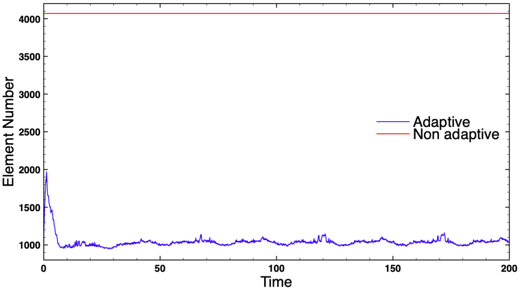
Laminar Pitching Airfoil

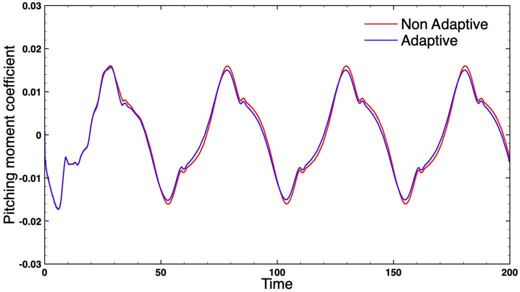


Transonic Pitching Airfoil

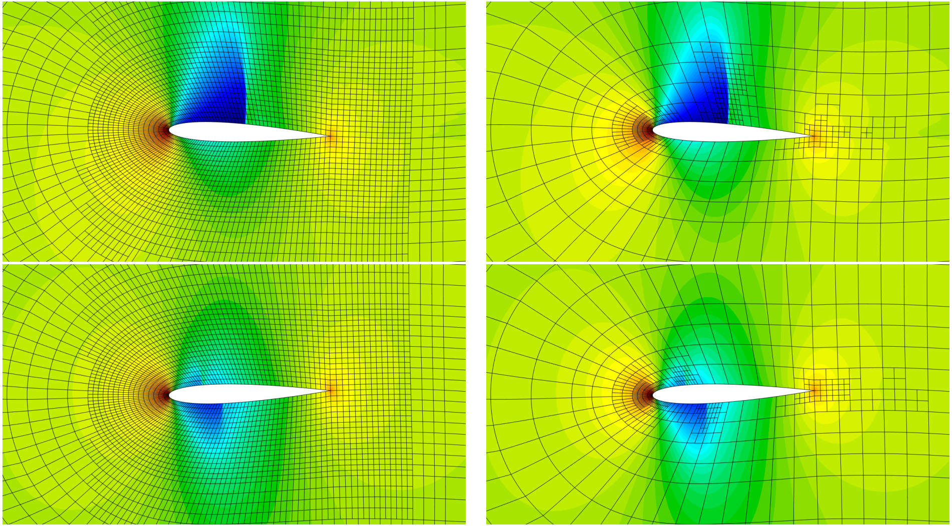
- $M_\infty = 0.755$, inviscid fluid, with artificial viscosity
- $\alpha_0 = 0.016^\circ$, $\Delta\alpha = 2.51^\circ$, $k = 0.0814$
- Variable intensity moving shock
- Level 0 mesh (right): 720 elements
- 2 level adaptation
- Non-adaptive mesh: 4070 elements







Transonic Pitching Airfoil



Conclusions and perspectives

- A DG framework that natively supports CAD geometries
- Curvilinear grids required for a truly high-order scheme
- ALE formulation with NURBS-based grid velocity
- Mesh movement does not impact overall accuracy
- A simple yet effective ALE-AMR coupling algorithm

- Further developments:
  - fully conservative sliding meshes
  - fluid-structure interaction
  - shape optimization

Thanks for your attention!