HYBRIDISABLE DISCONTINUOS GALERKIN METHODS

PART 2: RECENT DEVELOPMENTS OF HDG

Matteo Giacomini

XXII Spanish-French School Jacques-Louis Lions
January 12-14, 2026 - Paris

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
CIMNE INTERNATIONAL CENTRE FOR NUMERICAL METHODS IN ENGINEERING
IMTech

Thanks to…

Antonio Huerta
(UPC-CIMNE)

Rubén Sevilla
(Swansea University)

My co-authors:

Former PhD students:

Alexandros Karkoulias (2020)
Jordi Vila-Pérez (2021)
Davide Cortellessa (2024)
Luan M. Vieira (2024)

Outline

1. Review of DG methods: IPM & LDG
2. HDG formulation
3. Accuracy and computational cost

4. HDGlab: implementation aspects
5. HDG for mechanical problems
6. HDG with convection stabilisation via Riemann solvers
7. HDG with exact geometry and degree adaptivity
8. From HDG (high-order) to FCFV (low-order)
9. Some ongoing research lines

LABORATORI DE CÀLCUL NUMÈRIC UNIVERSITAT POLITÈCNICA DE CATALUNYA

Outline

1. Review of DG methods: IPM & LDG
2. HDG formulation
3. Accuracy and computational cost

4. HDGlab: implementation aspects
5. HDG for mechanical problems
6. HDG with convection stabilisation via Riemann solvers
7. HDG with exact geometry and degree adaptivity
8. From HDG (high-order) to FCFV (low-order)
9. Some ongoing research lines

Implementation aspects of HDG

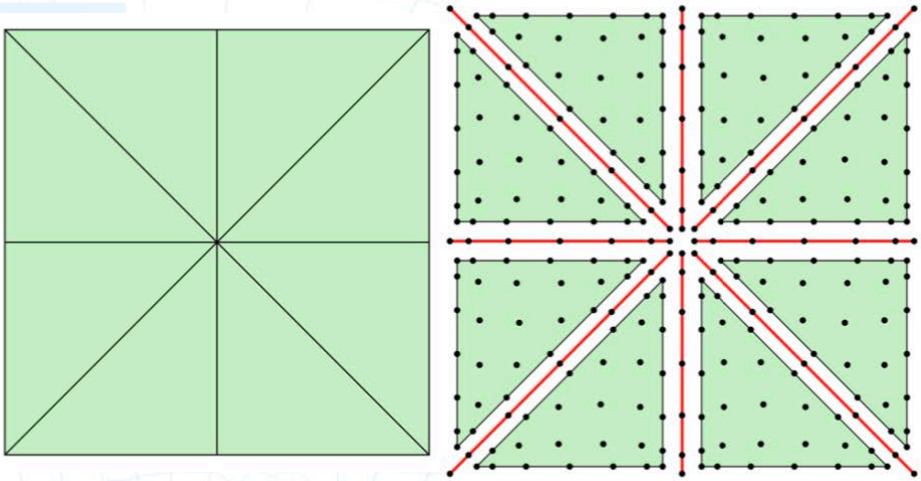
HDGlab

- Open-source **educational Matlab implementation** of the hybridisable discontinuous Galerkin (**HDG**) method.
- Repository: https://git.lacan.upc.edu/hybridLab/HDGlab

Features [Giacomini, Sevilla, Huerta. HDGlab. ACME, 2021]

1. **Scalar** (Poisson) and **vectorial** (Stokes) elliptic problems.
2. **High-order** polynomial approximations up to degree 9 (equally-spaced and Fekete nodal distributions).
3. Support of **curved isoparametric** simplicial elements (seamless in 2D and 3D).
4. Support of **non-uniform degree** polynomial approximations.
5. **Interface with Gmsh** mesh generator.

Degrees of freedom and functional approximation



Primal variable (solution):

$$u^h = \sum_{i=1}^{\mathtt{n_{en}}} N_i \mathtt{u}_i,$$

Mixed variable (gradient):

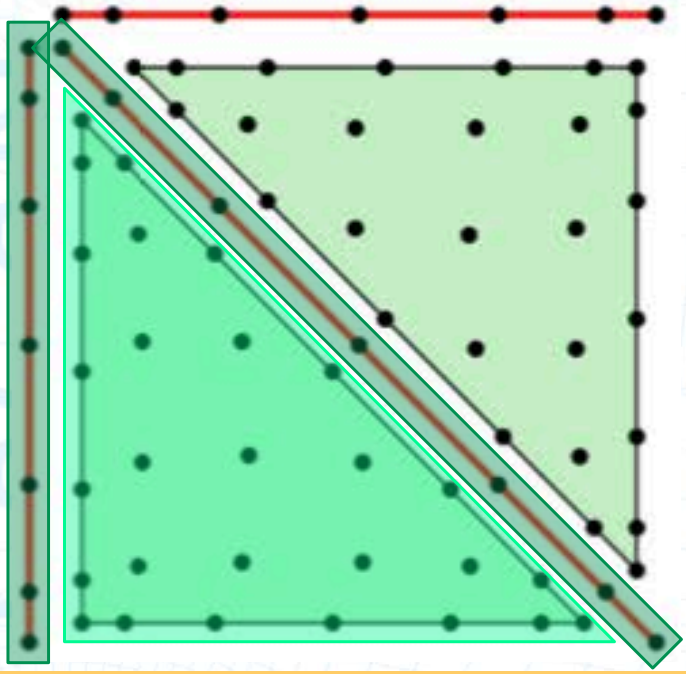
$$\boldsymbol{q}^h = \sum_{i=1}^{\mathtt{n_{en}}} N_i \mathbf{q}_i,$$

Hybrid variable (trace):

$$\hat{u}^h = \sum_{i=1}^{\mathtt{n_{fn}}} \hat{N}_i \hat{\mathtt{u}}_i,$$

$\mathtt{n_{en}}$: Number of nodes per element $\mathtt{n_{fn}}$: Number of nodes per face

Degrees of freedom and functional approximation



Local degrees of freedom

$$u^h = \sum_{i=1}^{\mathtt{n_{en}}} N_i \mathtt{u}_i, \quad \text{(solution)}$$

$$\boldsymbol{q}^h = \sum_{i=1}^{\mathtt{n_{en}}} N_i \mathbf{q}_i, \quad \text{(gradient of the solution)}$$

Global degrees of freedom

$$\hat{u}^h = \sum_{i=1}^{\mathtt{n_{fn}}} \hat{N}_i \hat{\mathtt{u}}_i, \quad \text{(trace of the solution)}$$

The HDG discrete system: local problem

Linear system from the local problem:

$$\begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uq} \\ \mathbf{A}_{uq}^T & \mathbf{A}_{qq} \end{bmatrix}_e \begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e = \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_q \end{Bmatrix}_e + \begin{bmatrix} \mathbf{A}_{u\hat{u}} \\ \mathbf{A}_{q\hat{u}} \end{bmatrix}_e \hat{\mathbf{u}}_e, \quad \rightarrow \quad \begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e = \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e + \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e \hat{\mathbf{u}}_e,$$

$$\begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e := \begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uq} \\ \mathbf{A}_{uq}^T & \mathbf{A}_{qq} \end{bmatrix}_e^{-1} \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_q \end{Bmatrix}_e.$$

$$\begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e := \begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uq} \\ \mathbf{A}_{uq}^T & \mathbf{A}_{qq} \end{bmatrix}_e^{-1} \begin{bmatrix} \mathbf{A}_{u\hat{u}} \\ \mathbf{A}_{q\hat{u}} \end{bmatrix}_e$$

The HDG discrete system: global problem

Linear system from the global problem:

$$\sum_{e=1}^{n_{el}} \left\{ \begin{bmatrix} \mathbf{A}_{u\hat{u}}^T & \mathbf{A}_{q\hat{u}}^T \end{bmatrix}_e \begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e + [\mathbf{A}_{\hat{u}\hat{u}}]_e \hat{\mathbf{u}}_e \right\} = \sum_{e=1}^{n_{el}} [\mathbf{f}_{\hat{u}}]_e.$$

$$\hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}},$$

The matrix and the vector are assembled starting from the elemental contributions $\begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e = \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e + \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e \hat{\mathbf{u}}_e,$

$$\hat{\mathbf{K}}^e := \begin{bmatrix} \mathbf{A}_{u\hat{u}}^T & \mathbf{A}_{q\hat{u}}^T \end{bmatrix}_e \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e + [\mathbf{A}_{\hat{u}\hat{u}}]_e, \qquad \hat{\mathbf{f}}^e := [\mathbf{f}_{\hat{u}}]_e - \begin{bmatrix} \mathbf{A}_{u\hat{u}}^T & \mathbf{A}_{q\hat{u}}^T \end{bmatrix}_e \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e.$$

Computed from the local problems

The HDG discrete system: postprocessing

Local element-by-element postprocess

$$\begin{bmatrix} \mathbf{A}_{\star\star} & \mathbf{a}_{\star\lambda} \\ \mathbf{a}_{\star\lambda}^T & 0 \end{bmatrix}_e \begin{Bmatrix} \mathbf{u}_\star \\ \lambda \end{Bmatrix}_e = \begin{bmatrix} \mathbf{0} & \mathbf{A}_{\star q} \\ \mathbf{a}_{\star\lambda}^T & \mathbf{0} \end{bmatrix}_e \begin{Bmatrix} \mathcal{I}^\star \mathbf{u} \\ \mathcal{I}^\star_{\mathtt{n_{sd}}} \mathbf{q} \end{Bmatrix}_e,$$

Structure of an HDG code

HDGpoisson.m:

```
disp('HDG solution...')
[uHat,local] = hdg_Poisson_GlobalSystem(mesh,refElem,refFace,hdg,ctt,problemParams);
[u,q] = hdg_Poisson_LocalProblem(mesh,refElem,refFace,hdg,uHat,local,ctt.nOfComponents);
uStar = hdg_Poisson_LocalPostprocess(mesh,refElem,u,q,problemParams,ctt.nOfComponents);
```

Steps:

1. **Compute and assembly the block matrices** for the local and global problems.
2. Solve the **global problem**.
3. Solve **element-by-element** the **local problems**.
4. Compute the **postprocessed solution**.

Step 1a: Compute elemental matrices

HDG_Poisson_GlobalSystem.m:

```
% Elemental matrices (from faces to elements)
[Zqle,Zule,zqfe,zufe,Ke,fe] = hdg_Poisson_ElementalMatrices(refElem,refFace,Xe,pElem,...
                                                            matElem,hdg.tau,faceInfo,ctt,problemParams);
```

For each element in the mesh, compute the blocks related to the local problems:

$$\begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e = \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e + \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e \hat{\mathbf{u}}_e,$$

$$\begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e := \begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uq} \\ \mathbf{A}_{uq}^T & \mathbf{A}_{qq} \end{bmatrix}_e^{-1} \begin{Bmatrix} \mathbf{f}_u \\ \mathbf{f}_q \end{Bmatrix}_e.$$

$$\begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e := \begin{bmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uq} \\ \mathbf{A}_{uq}^T & \mathbf{A}_{qq} \end{bmatrix}_e^{-1} \begin{bmatrix} \mathbf{A}_{u\hat{u}} \\ \mathbf{A}_{q\hat{u}} \end{bmatrix}_e$$

Step 1b: Compute matrices for global problem

HDG_Poisson_GlobalSystem.m:

```
% Elemental matrices (from faces to elements)
[Zqle,Zule,zqfe,zufe,Ke,fe] = hdg_Poisson_ElementalMatrices(refElem,refFace,Xe,pElem,...
                                                            matElem,hdg.tau,faceInfo,ctt,problemParams);
```

For each element in the mesh, compute its contribution to the global problem:

$$\hat{\mathbf{K}}^e := \begin{bmatrix} \mathbf{A}_{u\hat{u}}^T & \mathbf{A}_{q\hat{u}}^T \end{bmatrix}_e \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e + [\mathbf{A}_{\hat{u}\hat{u}}]_e,$$

$$\hat{\mathbf{f}}^e := [\mathbf{f}_{\hat{u}}]_e - \begin{bmatrix} \mathbf{A}_{u\hat{u}}^T & \mathbf{A}_{q\hat{u}}^T \end{bmatrix}_e \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e.$$

Contributions of an element to the matrix and vector of the global problem

Step 1c: Assembly the global problem

HDG_Poisson_GlobalSystem.m:

```
% Assembly
[indexGlobalFace, nOfFaceDOF] = hdgElemToFaceIndex(mesh.indexTf,refFace,hdg.faceInfo(iElem),...
                                refElem(pElem).nOfFaces,ctt.nOfComponents);

nOfFaceDOF2 = nOfFaceDOF^2;
currentIndex = indexIni + (1:nOfFaceDOF);
currentIndex2 = indexIni + (1:nOfFaceDOF2);
for i=1:nOfFaceDOF
    mat.i(currentIndex) = indexGlobalFace(i);
    mat.j(currentIndex) = indexGlobalFace;
    currentIndex = currentIndex + nOfFaceDOF;
end
mat.Kij(currentIndex2) = reshape(Ke',1,nOfFaceDOF2);

% RHS
f(indexGlobalFace) = f(indexGlobalFace) + fe;
```

Key aspect: exploit the relationship between **local** (element-based) and **global** (face-based) **degrees of freedom**

Step 2: Solve the global problem

HDG_Poisson_GlobalSystem.m:

```
% Solution
uhat = zeros(hdg.nDOFglobal*ctt.nOfComponents,1);
uhat(hdg.vDOFtoSolve) = K(hdg.vDOFtoSolve,hdg.vDOFtoSolve)\f(hdg.vDOFtoSolve);
```

Solve the system of the globally-coupled degrees of freedom (variables on the **internal + Neumann faces**)

$$\hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}},$$

Step 3: Solve the local problems

HDG_Poisson_LocalProblem.m:

```
u(indexUIni:indexUEnd) = local(iElem).Zul*uHat(indexGlobalFace) + local(iElem).zuf;
q(indexQIni:indexQEnd) = local(iElem).Zql*uHat(indexGlobalFace) + local(iElem).zqf;
```

$$\begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix}_e = \begin{Bmatrix} \mathbf{z}_u^f \\ \mathbf{z}_q^f \end{Bmatrix}_e + \begin{bmatrix} \mathbf{Z}_{u\hat{u}} \\ \mathbf{Z}_{q\hat{u}} \end{bmatrix}_e \hat{\mathbf{u}}_e,$$

For each mesh element, compute the primal and mixed variable inside the element, using:

1. the previously assembled matrices;
2. the solution of the global problem;
3. the previously assembled vectors.

Step 4: Compute the postprocessed solution

HDG_Poisson_LocalPostprocess.m:

```
[Ke,Bqe,intUStar,intU] = hdg_Poisson_LocalPostprocessElemMat(refElem(pElemStar),...
                                XeStar,ug,qg,kappa);

% Constraint with Lagrange multipliers
K = [Ke intUStar; intUStar' 0];
f = [Bqe; intU];
```

For each element in the mesh, compute the postprocessed primal variable solving

$$\begin{bmatrix} \mathbf{A}_{\star\star} & \mathbf{a}_{\star\lambda} \\ \mathbf{a}_{\star\lambda}^T & 0 \end{bmatrix}_e \begin{Bmatrix} \mathbf{u}_\star \\ \lambda \end{Bmatrix}_e = \begin{bmatrix} \mathbf{0} & \mathbf{A}_{\star q} \\ \mathbf{a}_{\star\lambda}^T & \mathbf{0} \end{bmatrix}_e \begin{Bmatrix} \mathcal{I}^\star \mathbf{u} \\ \mathcal{I}_{\mathrm{n_{sd}}}^\star \mathbf{q} \end{Bmatrix}_e,$$

HDG formulation for mechanical problems

Cauchy stress vs. velocity-pressure formulation

- Cauchy stress tensor formulation:

$$
\begin{cases}
-\nabla\cdot\boldsymbol{\sigma} = \boldsymbol{s} & \text{in } \Omega,\\
\nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega,\\
\boldsymbol{\sigma} = -p\mathbf{I}_{\mathrm{n_{sd}}} + 2\nu\nabla^{\mathrm{s}}\boldsymbol{u} & \text{in } \Omega,\\
\boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D,\\
\boldsymbol{n}\cdot\boldsymbol{\sigma} = \boldsymbol{t} & \text{on } \Gamma_N,
\end{cases}
\qquad \text{with } \nabla^{\mathrm{s}} := \tfrac{1}{2}\left(\nabla + \nabla^T\right)
$$

- Velocity-pressure formulation:

$$
\begin{cases}
-\nabla\cdot\left(\nu\nabla\boldsymbol{u} - p\mathbf{I}_{\mathrm{n_{sd}}}\right) = \boldsymbol{s} & \text{in } \Omega,\\
\nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega,\\
\boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D,\\
\boldsymbol{n}\cdot\left(\nu\nabla\boldsymbol{u} - p\mathbf{I}_{\mathrm{n_{sd}}}\right) = \boldsymbol{t} & \text{on } \Gamma_N,
\end{cases}
$$

not the same

of course

$$\nabla\cdot\left(\nu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)\right) = \nabla\cdot\left(\nu\nabla\boldsymbol{u}\right) + \nabla\nu(\nabla\cdot\boldsymbol{u}) = \nabla\cdot\left(\nu\nabla\boldsymbol{u}\right)$$

BUT $\boldsymbol{n}\cdot\boldsymbol{\sigma} = \boldsymbol{n}\cdot\left(\nu\nabla\boldsymbol{u} - p\boldsymbol{I}_{\mathrm{n_{sd}}}\right) + \boldsymbol{n}\cdot\left(\nu\nabla\boldsymbol{u}^T\right)$

A simple alternative

[Cockburn, Nguyen, Peraire, J. Sci. Comput. (2010)]

$$\boldsymbol{L} - \boldsymbol{\nabla} \boldsymbol{u} = \boldsymbol{0}$$

then

$$(2\nu \boldsymbol{\nabla}^{\mathrm{s}} \boldsymbol{u} - p \boldsymbol{I}_{\mathtt{n_{sd}}}) = \left(\nu(\boldsymbol{L} + \boldsymbol{L}^T) - p \boldsymbol{I}_{\mathtt{n_{sd}}}\right)$$

ISSUES: [Cockburn, Nguyen, Peraire. JSC 2010]

For low-order polynomials (k=1 & 2), HDG experiences:

- **Loss of optimal convergence** of *L*.
- **Loss of superconvergence** of u^{sc}.

Another mixed variable

- Define the **mixed** variable as $\boldsymbol{L} \in \mathcal{H}(\mathrm{div}, \mathbb{S})$

$$\begin{cases} \boldsymbol{L} + \sqrt{2\nu} \boldsymbol{\nabla}^{\mathrm{s}} \boldsymbol{u} = \boldsymbol{0} & \text{in } \Omega_e, \text{ and for } e = 1, \dots, \mathtt{n_{el}}, \\ \boldsymbol{\nabla} \cdot (\sqrt{2\nu} \boldsymbol{L} + p \boldsymbol{I}_{\mathtt{n_{sd}}}) + \boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = \boldsymbol{s} & \text{in } \Omega_e, \text{ and for } e = 1, \dots, \mathtt{n_{el}}, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 & \text{in } \Omega_e, \text{ and for } e = 1, \dots, \mathtt{n_{el}}, \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n} \cdot (\sqrt{2\nu} \boldsymbol{L} + p \boldsymbol{I}_{\mathtt{n_{sd}}}) + (\boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n} = -\boldsymbol{t} & \text{on } \Gamma_N, \\ [\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} & \text{on } \Gamma, \\ [\![\boldsymbol{n} \cdot (\sqrt{2\nu} \boldsymbol{L} + p \boldsymbol{I}_{\mathtt{n_{sd}}}) + (\boldsymbol{u} \otimes \boldsymbol{u}) \boldsymbol{n}]\!] = \boldsymbol{0} & \text{on } \Gamma, \end{cases}$$

where $\mathcal{H}(\mathrm{div}, \mathbb{S})$ is the space of square integrable symmetric tensors $\mathbb{S}$ of order $\mathtt{n_{sd}}$ with square integrable row-wise divergence.

HDG with strongly-imposed symmetry of L

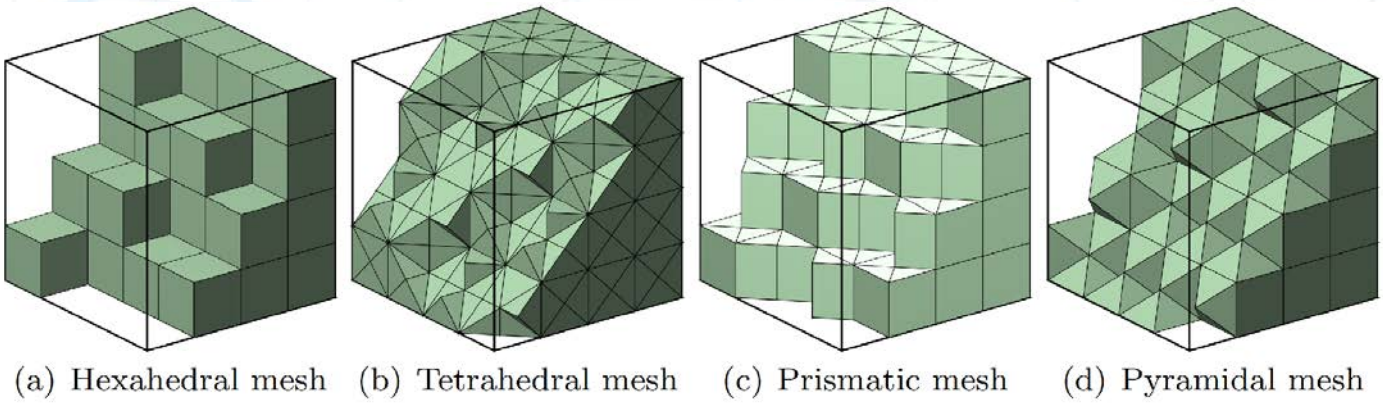
$$
\begin{aligned}
\boldsymbol{L} + \mathbf{D}^{1/2}\nabla_{\mathrm{S}}\boldsymbol{u} &= \mathbf{0} && \text{in } \Omega_e,\\
\nabla_{\mathrm{S}}^{T}\left(\mathbf{D}^{1/2}\boldsymbol{L} + \mathbf{E}\,p\right) &= \boldsymbol{s} && \text{in } \Omega_e,\\
\mathbf{E}^{T}\nabla_{\mathrm{S}}\boldsymbol{u} &= 0 && \text{in } \Omega_e,\\
\boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \Gamma_D,\\
\mathbf{N}^{T}\left(\mathbf{D}^{1/2}\boldsymbol{L} + \mathbf{E}\,p\right) &= -\boldsymbol{t} && \text{on } \Gamma_N,\\
[\![\boldsymbol{u}\otimes\boldsymbol{n}]\!] &= \mathbf{0} && \text{on } \Gamma,\\
[\![\mathbf{N}^{T}\left(\mathbf{D}^{1/2}\boldsymbol{L} + \mathbf{E}\,p\right)]\!] &= \mathbf{0} && \text{on } \Gamma,
\end{aligned}
$$

New definition of mixed variable:

1. Strong and pointwise imposition of symmetry
2. Physical tractions
3. Lower ndof for $\boldsymbol{L}$
4. Optimal convergence
5. **Superconvergent velocity**

[Giacomini, Karkoulias, Sevilla, Huerta. J. Sci. Comput. (2018)]

Robustness to element type

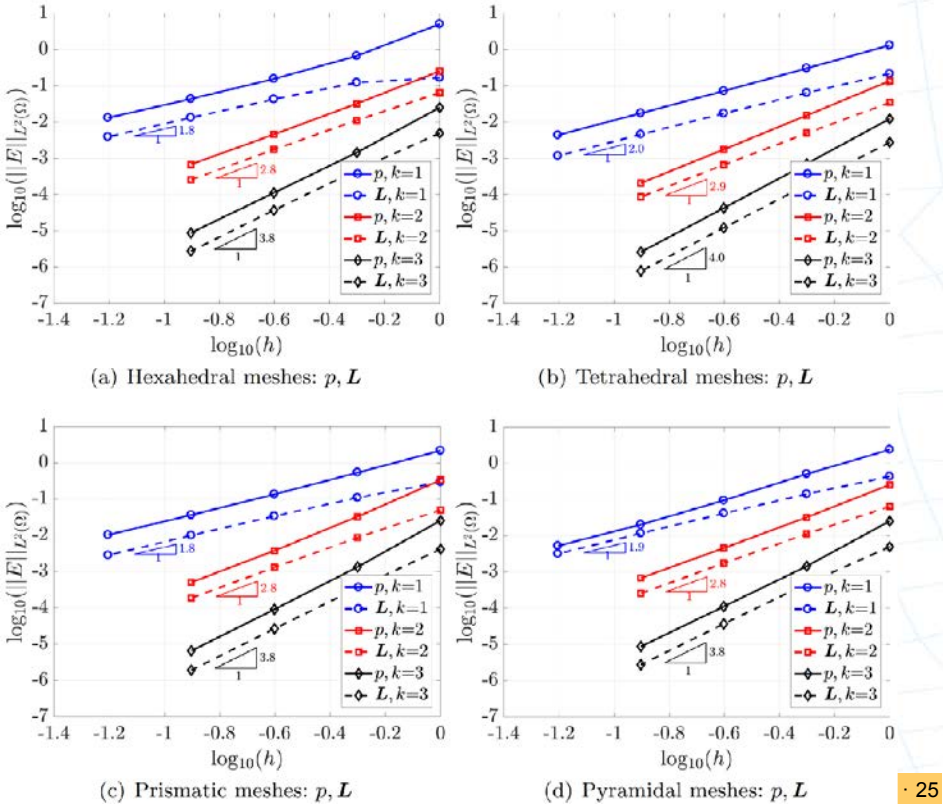


(a) Hexahedral mesh (b) Tetrahedral mesh (c) Prismatic mesh (d) Pyramidal mesh

[Giacomini, Karkoulias, Sevilla, Huerta, J. Sci. Comput. (2018)] - Stokes
[Sevilla, Giacomini, Karkoulias, Huerta, IJNME (2018)] - Elasticity

[Giacomini, Karkoulias, Sevilla, Huerta, J. Sci. Comput. (2018)] - Stokes

[Sevilla, Giacomini, Karkoulias, Huerta, IJNME (2018)] - Elasticity

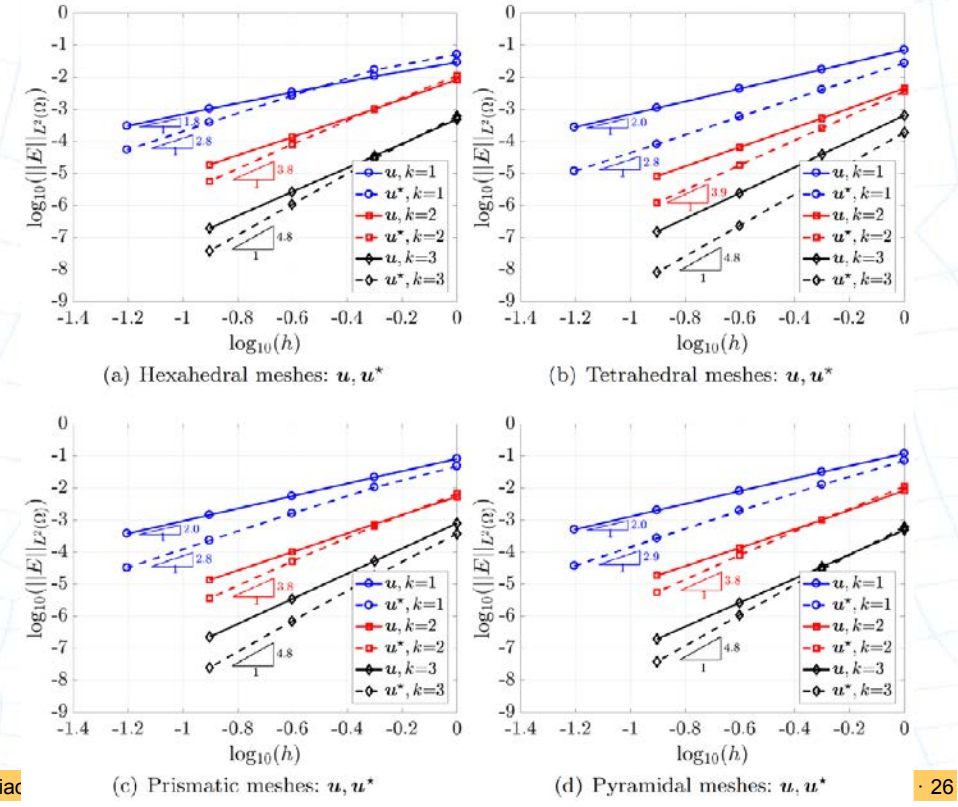


(a) Hexahedral meshes: $p, \boldsymbol{L}$ (b) Tetrahedral meshes: $p, \boldsymbol{L}$

(c) Prismatic meshes: $p, \boldsymbol{L}$ (d) Pyramidal meshes: $p, \boldsymbol{L}$

[Giacomini, Karkoulias, Sevilla, Huerta, J. Sci. Comput. (2018)] - Stokes

[Sevilla, Giacomini, Karkoulias, Huerta, IJNME (2018)] - Elasticity



(a) Hexahedral meshes: $\boldsymbol{u}, \boldsymbol{u}^\star$ (b) Tetrahedral meshes: $\boldsymbol{u}, \boldsymbol{u}^\star$

(c) Prismatic meshes: $\boldsymbol{u}, \boldsymbol{u}^\star$ (d) Pyramidal meshes: $\boldsymbol{u}, \boldsymbol{u}^\star$

Weakly compressible fluid-structure interaction

- Weakly compressible formulation alleviates the constraints of the instability condition of the artificial added mass effect, e.g. [van Brummelen. JAM (2009)] [La Spina, Förster, Kronbichler, Wall. IJNMF (2020)]
- HDG nicely suited for this problem: LBB, high-order, coupling…

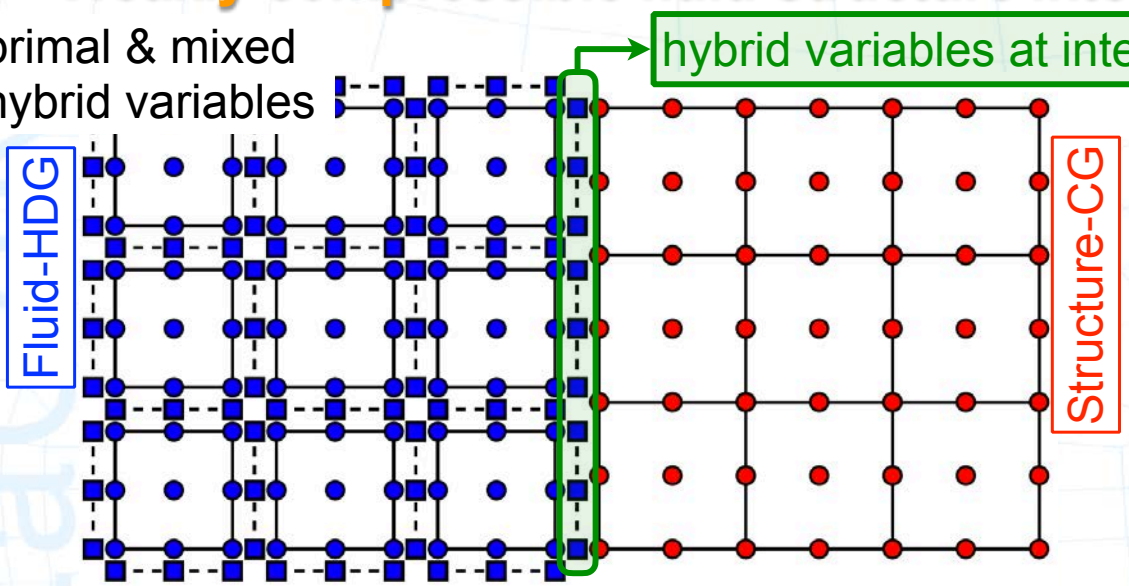
$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{v}) = 0 \\ \dfrac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{v}\otimes\boldsymbol{v} + p\boldsymbol{I}_{\mathrm{n_{sd}}}) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}^d = \rho \boldsymbol{b} \\ p(\rho) = 0 \end{cases}$$

with $\boldsymbol{\sigma}^d = \dfrac{\mu}{R_e}\boldsymbol{\varepsilon}^d$ and $\boldsymbol{\varepsilon}^d = 2\boldsymbol{\nabla}^{\mathrm{s}}\boldsymbol{v} - \dfrac{2}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v})\boldsymbol{I}_{\mathrm{n_{sd}}}$

[La Spina, Giacomini, Huerta, Comput. Mech. (2020)]
[La Spina, Kronbichler, Giacomini, Wall, Huerta, CMAME (2020)]

Weakly compressible fluid-structure interaction

- primal & mixed
- hybrid variables



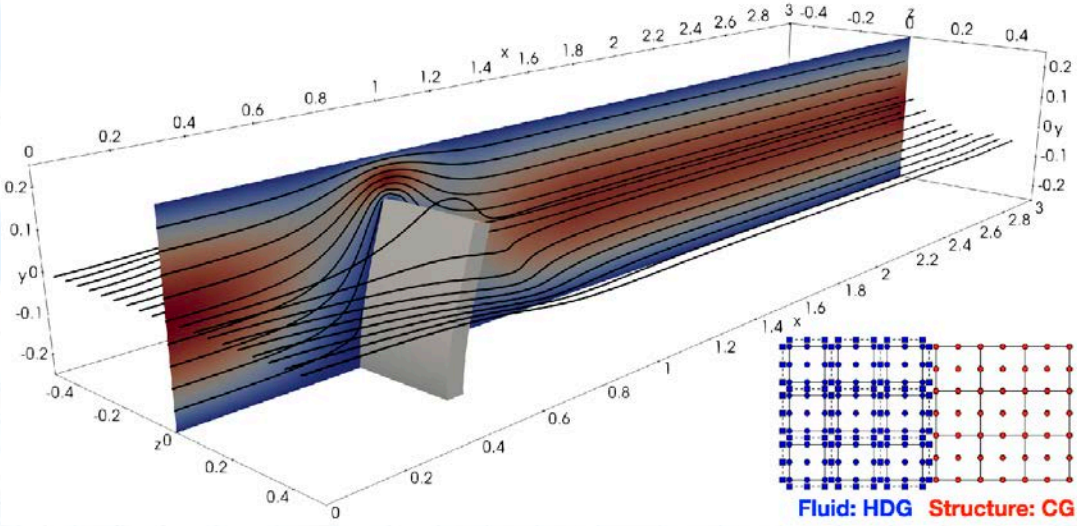
Idea: use the hybrid variable from the fluid domain to impose the Dirichlet condition for the structure at the interface using Nitsche's method

Weakly compressible fluid-structure interaction

[La Spina, Kronbichler, Giacomini, Wall, Huerta. CMAME 2020]

- Weakly compressible fluid at Re=5 (**HDG**)
- Wall of Neo-Hookean material (**CG with Nitsche's method**)

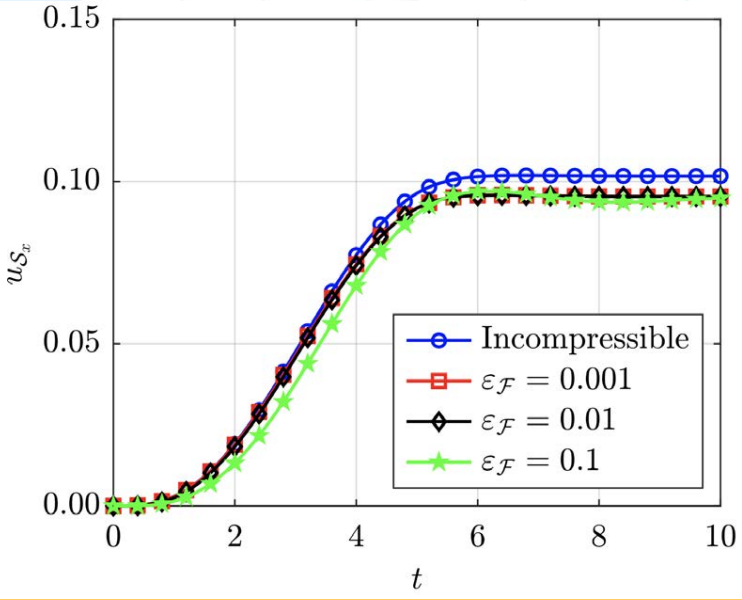
The **displacement of the wall** (Dirichlet BC for the structure) is **obtained from the motion of the fluid**, which is computed on a different nodal distribution.



Weakly compressible fluid-structure interaction

[La Spina, Kronbichler, Giacomini, Wall, Huerta. CMAME 2020]

- Displacement of the tip of the wall

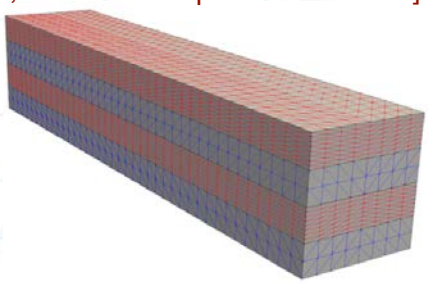


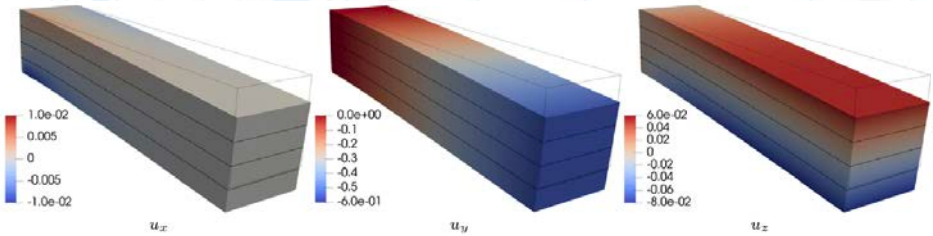
- **Robust** for **weak compressibility** effects.
- **Stable** in the **incompressible** limit.
- **No** need for **pressure stabilisation**.

Laminated composite beam

[La Spina, Giacomini, Huerta. Comput. Mech. 2020]

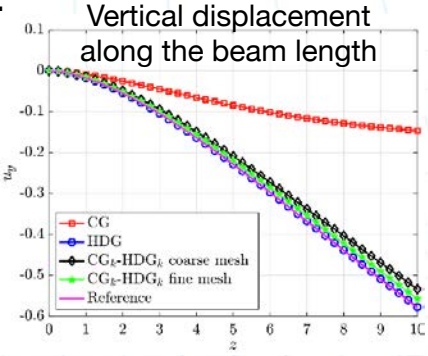
- Laminated beam featuring a compressible material (red) and a nearly incompressible one (blue).
- **HDG** discretisation in the nearly incompressible material and **CG with Nitsche's method** in the compressible one.





Components of the displacement field

Vertical displacement along the beam length



HDG with convection stabilisation inspired by Riemann solvers

Compressible Navier-Stokes equations

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{F}-\boldsymbol{G}) = \boldsymbol{0} \qquad \boldsymbol{U} = \begin{pmatrix} \rho \\ \rho \boldsymbol{u} \\ \rho E \end{pmatrix}$$

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} \rho \boldsymbol{u}^T \\ \rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I} \\ (\rho E + p)\boldsymbol{u}^T \end{pmatrix}$$

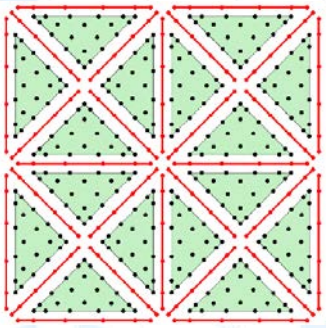
$$\boldsymbol{G}(\boldsymbol{U}, \boldsymbol{\varepsilon}, \boldsymbol{\phi}) = \frac{1}{\mathrm{Re}_\infty} \begin{pmatrix} \boldsymbol{0}^T \\ 2\boldsymbol{\varepsilon} \\ (2\boldsymbol{\varepsilon}\boldsymbol{u} + \boldsymbol{\phi}/\mathrm{Pr}_\infty)^T \end{pmatrix}$$

mixed formulation using mixed variables

$$\boldsymbol{\varepsilon} := \tfrac{1}{2}(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T) - \tfrac{1}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{u})\boldsymbol{I}$$

$$\boldsymbol{\phi} := \boldsymbol{\nabla}\theta$$

HDG



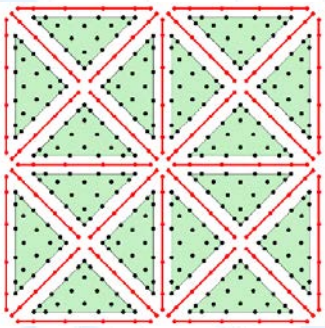
HDG for compressible Navier-Stokes

1.- The local problem: at each element solve

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{F}-\boldsymbol{G}) \right) \cdot \boldsymbol{V}\, d\Omega = 0$$

+ Dirichlet boundary conditions: hybrid variables $\widehat{\boldsymbol{U}}$

HDG



HDG for compressible Navier-Stokes

1.- The **local** problem: at each element solve

$$\int_{\Omega_e} \boldsymbol{U}_t \cdot \boldsymbol{V} - (\boldsymbol{F} - \boldsymbol{G}) : \boldsymbol{\nabla} \boldsymbol{V} \, d\Omega + \int_{\partial\Omega_e} (\widehat{\boldsymbol{F}} - \widehat{\boldsymbol{G}}) \boldsymbol{n} \cdot \boldsymbol{V} \, d\Gamma$$

+ Dirichlet boundary conditions: hybrid variables $\widehat{\boldsymbol{U}}$

2.- The **global** problem solves for hybrid variables $\widehat{\boldsymbol{U}}$ enforcing:

2.1.- continuity of **numerical fluxes** along the interior skeleton

2.2.- **boundary conditions**

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

HDG numerical fluxes

$$\int_{\Omega_e} \boldsymbol{U}_t \cdot \boldsymbol{V} - (\boldsymbol{F} - \boldsymbol{G}) : \boldsymbol{\nabla} \boldsymbol{V} \, d\Omega + \int_{\partial\Omega_e} (\widehat{\boldsymbol{F}} - \widehat{\boldsymbol{G}}) \boldsymbol{n} \cdot \boldsymbol{V} \, d\Gamma$$

➡ Exploit the hybrid variable to evaluate the physical fluxes

➡ Encapsulate the Riemann solver in the stabilisation parameter

$$\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} := \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \boldsymbol{\tau}^a(\widehat{\boldsymbol{U}})(\boldsymbol{U}_e - \widehat{\boldsymbol{U}}) \quad \text{and}$$

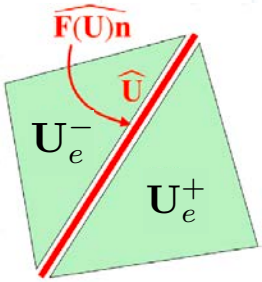
$$\widehat{\boldsymbol{G}(\boldsymbol{U}_e, \boldsymbol{\varepsilon}_e^d, \boldsymbol{\phi}_e)\boldsymbol{n}} := \boldsymbol{G}(\widehat{\boldsymbol{U}}, \boldsymbol{\varepsilon}_e^d, \boldsymbol{\phi}_e)\boldsymbol{n} - \boldsymbol{\tau}^d(\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$$

$$\boldsymbol{\tau}^d = Re^{-1} \operatorname{diag}\left(0, \mathbf{1}_{\mathrm{n_{sd}}}, \left[(\gamma - 1) M_\infty^2 Pr\right]^{-1}\right)$$

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Robust treatment of convection

$$\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} := \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \boldsymbol{\tau}^a(\widehat{\boldsymbol{U}})(\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$$



Riemann solvers-inspired stabilisation

Lax-Friedrichs: $\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \widehat{\lambda}_{\max}(\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$

Roe: $\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + |\boldsymbol{A}_n(\widehat{\boldsymbol{U}})|(\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$

Intermediate state: $\widehat{\boldsymbol{U}} = \dfrac{\boldsymbol{U}_e^+ + \boldsymbol{U}_e^-}{2}$

Jacobian of the convective flux in the normal direction

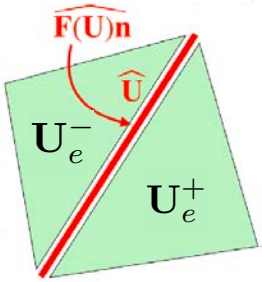
$\boldsymbol{A}_n(\widehat{\boldsymbol{U}}) := [\partial \boldsymbol{F}(\widehat{\boldsymbol{U}})/\partial \widehat{\boldsymbol{U}}]$

$\widehat{\lambda}_{\max}$: max eigenvalue of $\mathbf{A}_n$

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Robust treatment of convection

$$\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} := \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \boldsymbol{\tau}^a(\widehat{\boldsymbol{U}})(\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$$



Riemann solvers-inspired stabilisation

HLL: $\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \left[s^+ \mathbf{I}_{\mathrm{n_{sd}+2}}\right](\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$

HLLEM: $\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \left[s^+ \boldsymbol{\theta}(\widehat{\boldsymbol{U}})\right](\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$

Intermediate state: $\widehat{\boldsymbol{U}} = \dfrac{s^+ \boldsymbol{U}_e^+ + s^- \boldsymbol{U}_e^-}{s^+ + s^-}$

s^+: largest wave speed

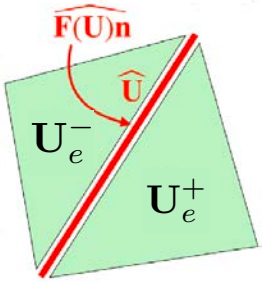
[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Robust treatment of convection

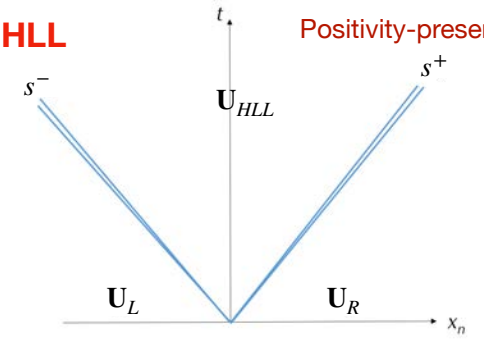
Riemann solvers-inspired stabilisation

HLL: $$\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \left[s^{+}\mathbf{I}_{\mathrm{n_{sd}}+2}\right](\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$$

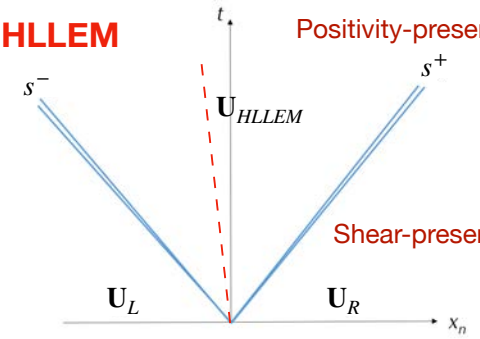
HLLEM: $$\widehat{\boldsymbol{F}(\boldsymbol{U}_e)\boldsymbol{n}} = \boldsymbol{F}(\widehat{\boldsymbol{U}})\boldsymbol{n} + \left[s^{+}\boldsymbol{\theta}(\widehat{\boldsymbol{U}})\right](\boldsymbol{U}_e - \widehat{\boldsymbol{U}})$$



HLL Positivity-preserving



HLLEM Positivity-preserving



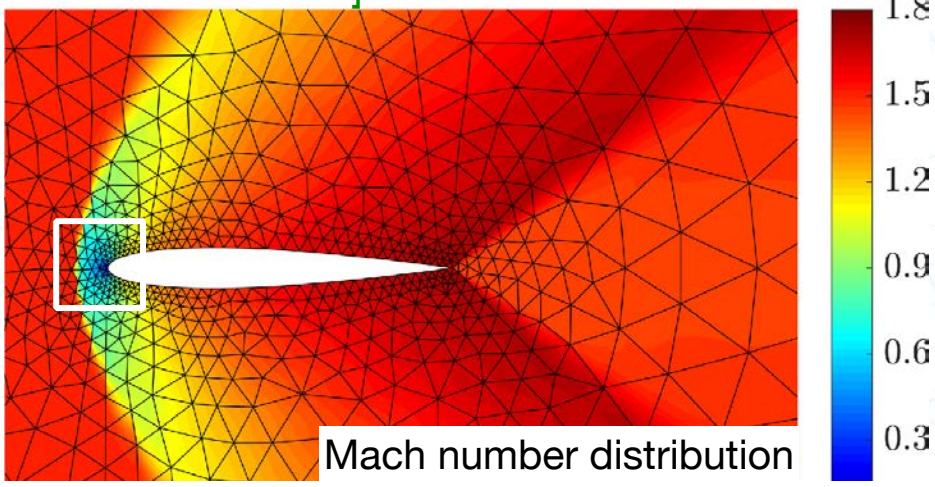
Shear-preserving

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Supersonic inviscid flow over a NACA 0012 aerofoil

Setup: $M_\infty = 1.5$, HLL Riemann solver, polynomial degree $k = 4$
also for geometry (high-order curved mesh)

Shock capturing [Persson & Peraire AIAA'06] :
discontinuity
shock sensor with
Laplacian-based
artificial viscosity



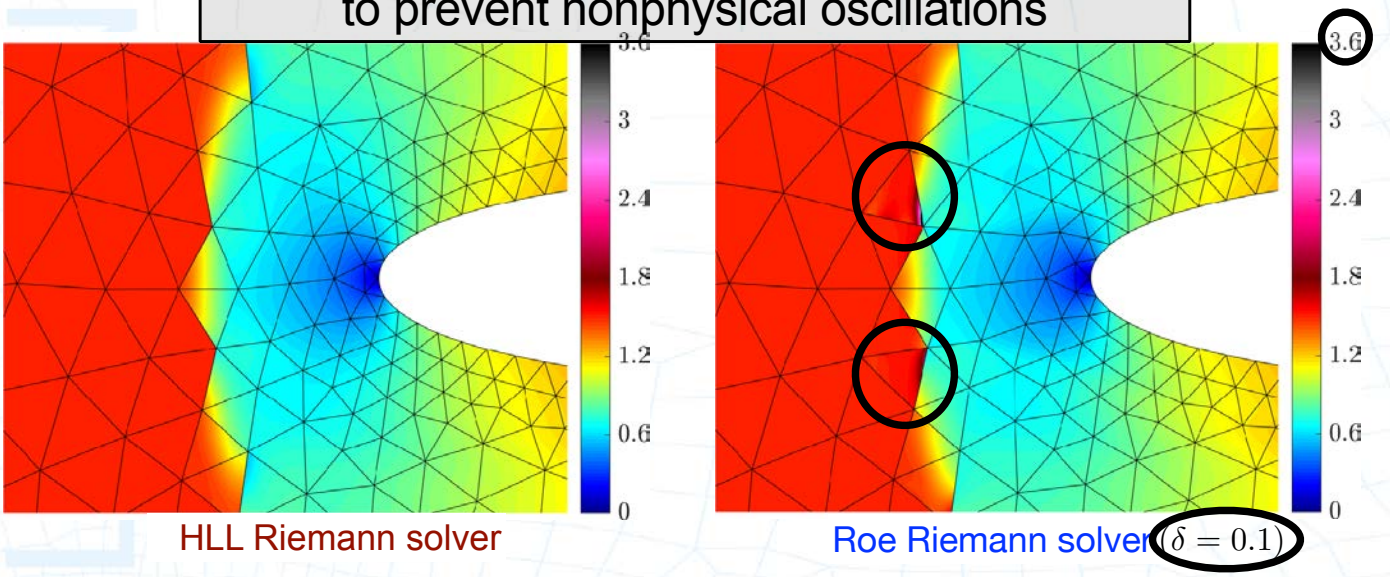
Mach number distribution

Positivity preservation in presence of shocks

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Supersonic inviscid flow over a NACA 0012 aerofoil

Roe Riemann solver requires entropy-fix tuning to prevent nonphysical oscillations



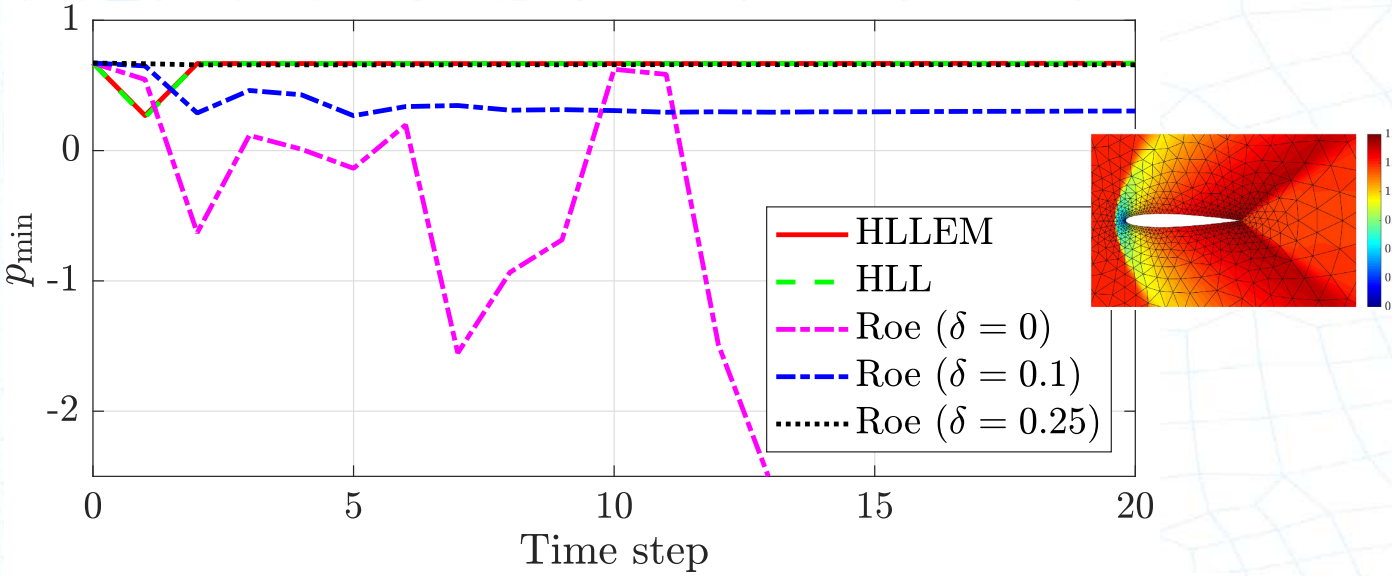
HLL Riemann solver

Roe Riemann solver ($\delta = 0.1$)

[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

Supersonic inviscid flow over a NACA 0012 aerofoil

HDG with **HLL** Riemann solver is **robust** and avoids non-physical solutions **without entropy fix**



[Vila-Pérez, Giacomini, Sevilla, Huerta, Arch. Comput. Meth. Eng.(2021)]

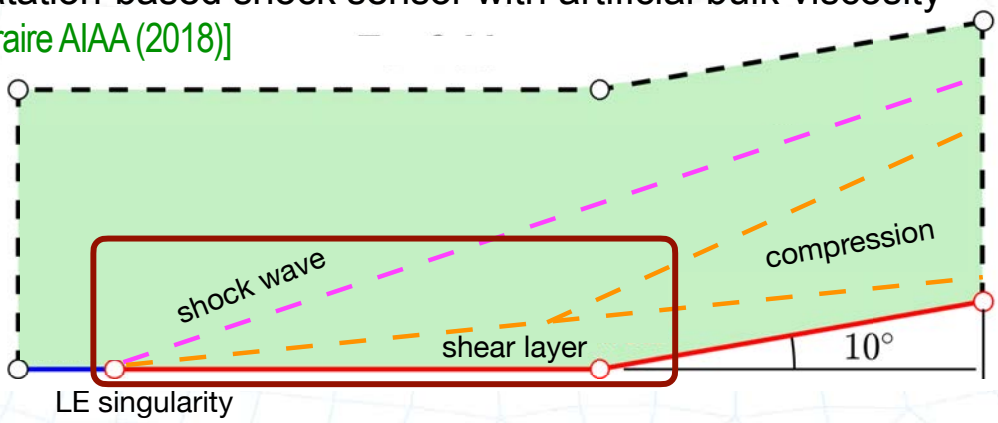
Supersonic viscous flow over a compression corner

Supersonic viscous flow over a compression corner

Setup: $M_\infty = 3$, $Re = 16\,800$, HLLEM Riemann solver, polynomial degree $k = 3$

Shock capturing: dilatation-based shock sensor with artificial bulk viscosity
[Fernández, Nguyen & Peraire AIAA (2018)]

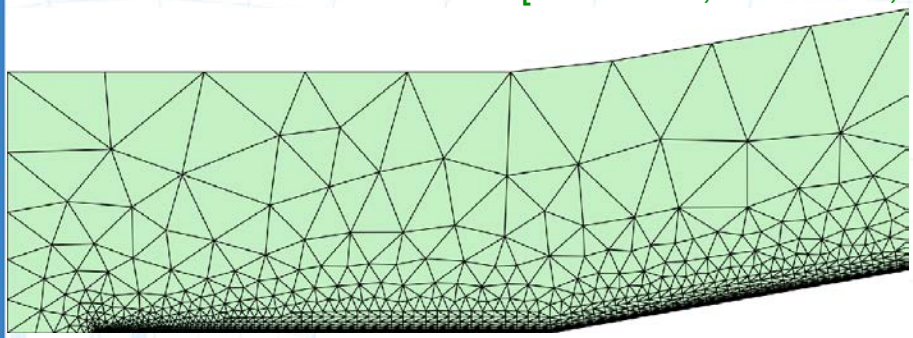
Many interacting phenomena!



Near the wall: shock wave/boundary layer **interaction**
[Vila-Pérez, Giacomini, Sevilla, Huerta. ACME, 2021]

Supersonic viscous flow over a compression corner

[Vila-Pérez, Giacomini, Sevilla, Huerta. ACME, 2021]



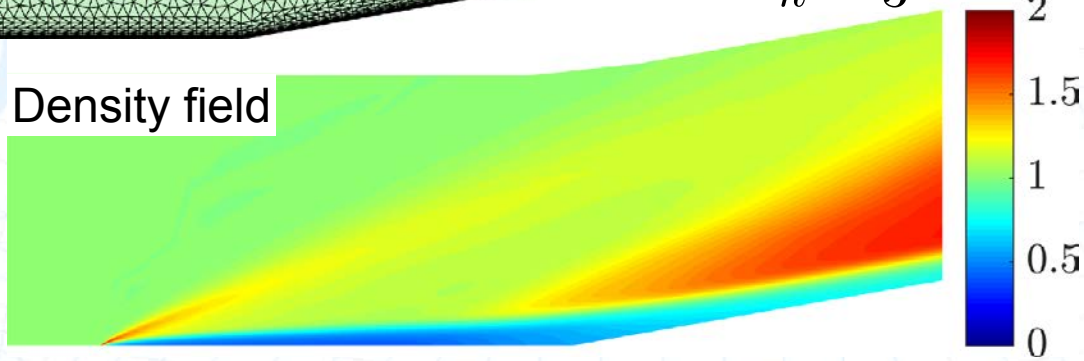
Coarse isotropic mesh

$\Longrightarrow$

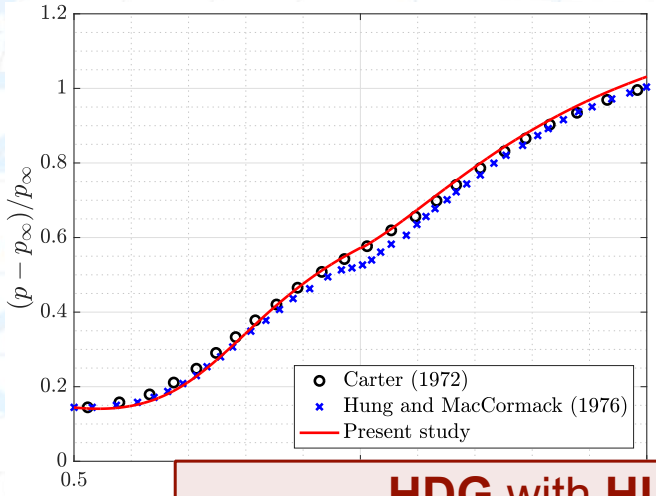
not adapted to capture shocks!

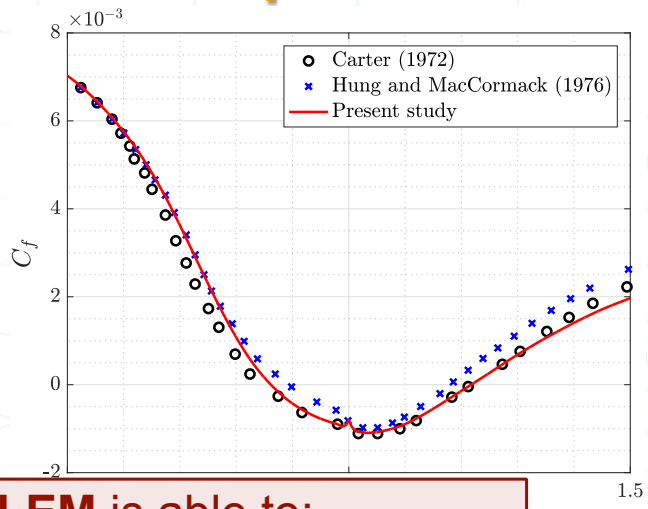
$k = 3$

Density field



Supersonic viscous flow over a compression corner



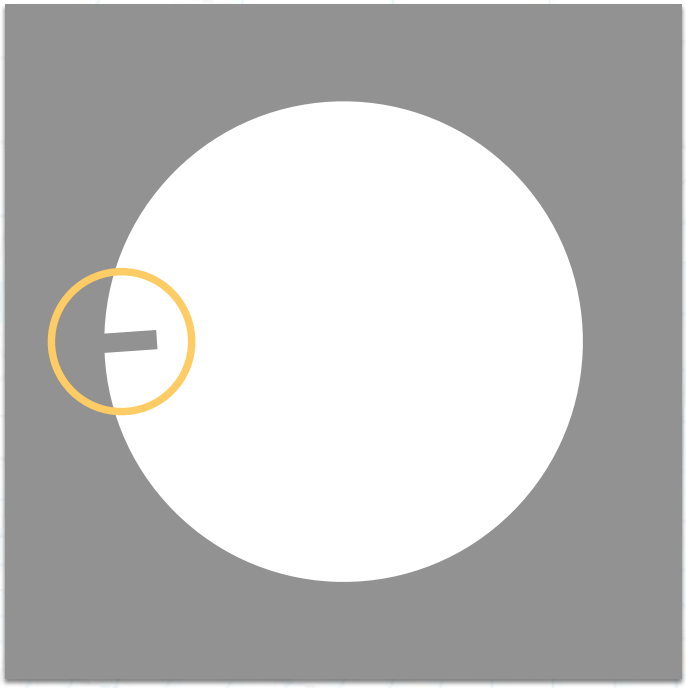


HDG with **HLLEM** is able to:
- ✓ accurately represent boundary layers
- ✓ preserve positivity in abrupt shocks
- ✓ handle shock wave/boundary layer interaction

[Vila-Pérez, Giacomini, Sevilla, Huerta. ACME, 2021]

High-fidelity HDG with exact geometry and degree adaptivity

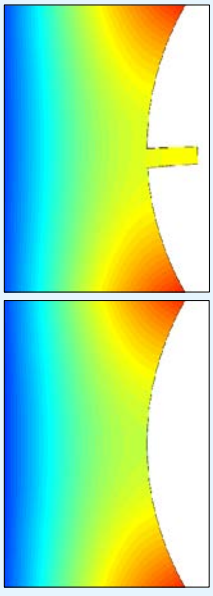
The importance of exact geometry representation



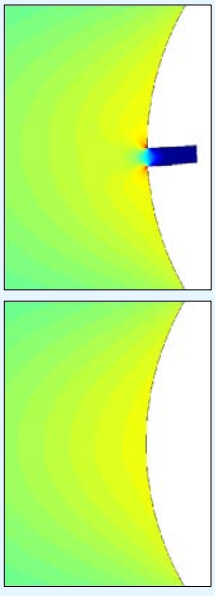
Is this feature relevant for the simulation?

The importance of exact geometry representation

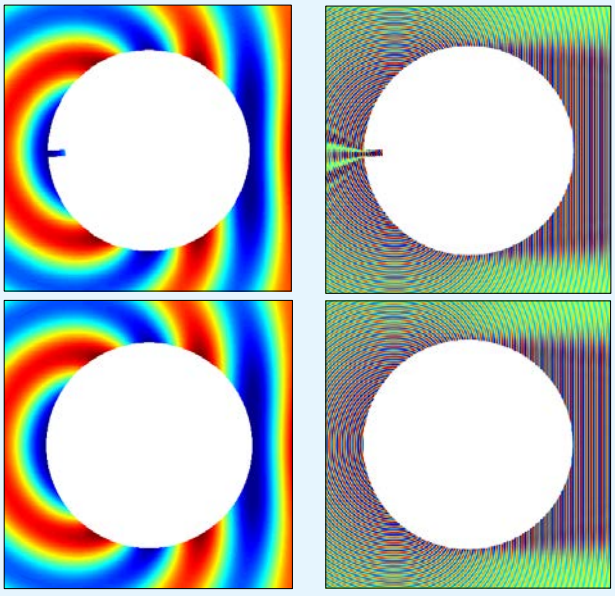
Heat transfer



Solid mechanics



Electromagnetics / Acoustics



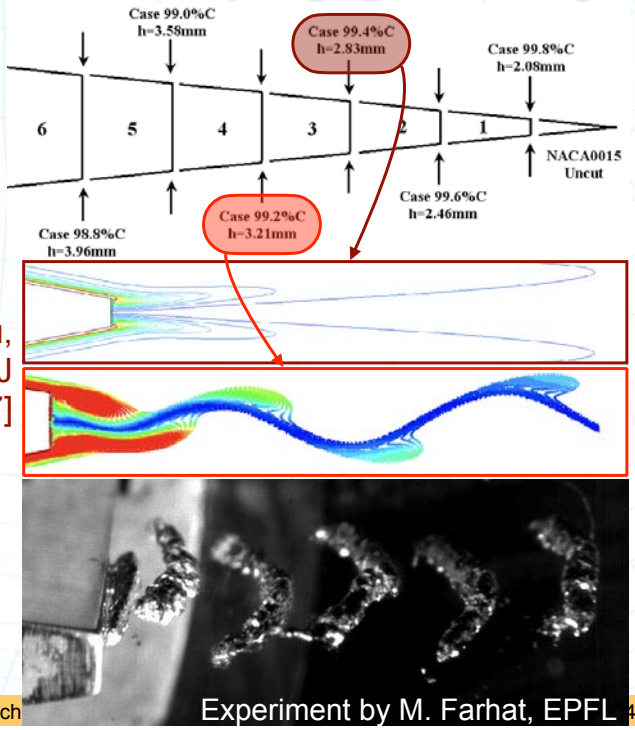
Is it possible to simplify geometry?

- Not easy to determine which features can be removed
- Manual de-featuring can take **up to 90% of the simulation time**
- Dependent on the **problem** and **parameters**

Case	Vortex Shedding
99.8%C	No
99.6%C	No
99.4%C	No
99.2%C	Yes
99.0%C	Yes
98.8%C	Yes

[Do, Chen, Tu, ANZIAM J 2007]

Need for ***exact*** **geometry**



Experiment by M. Farhat, EPFL

Simulations accounting for exact geometry

- **Isogeometric Analysis (IGA)**: Use the same approximation (NURBS functions) for the geometry and the solution.
- IGA is suited for problems in which the domain basically coincides with its boundary (e.g., thin shells).

[Hughes, Cottrell, Bazilevs. Wiley 2009]

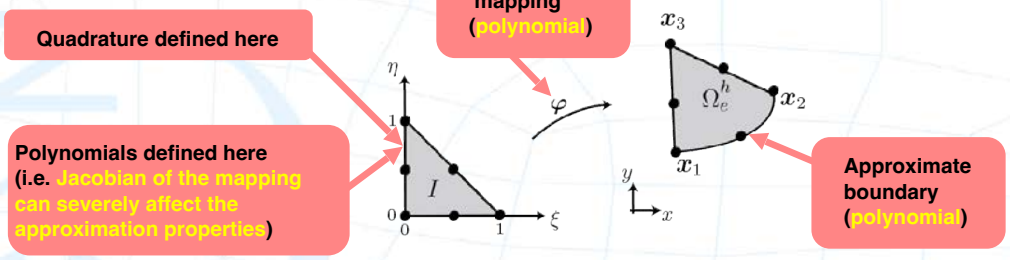
- **NURBS-enhanced FEM (NEFEM)**: Use standard finite element basis functions for the solution, while geometry is described using NURBS.
- NEFEM maintains the classical structure of FEM in terms of flowchart of the code and resulting matrices.

[Sevilla, Fernández-Méndez, Huerta. ACME 2011]

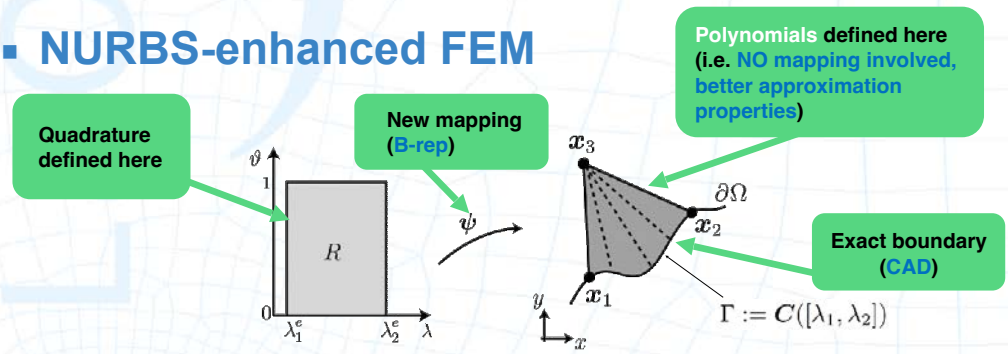
NURBS-enhanced finite element method

[Sevilla, Fernández-Méndez, Huerta, IJNME 2008]
[Sevilla, Fernández-Méndez, Huerta, ACME 2011]

- **Isoparametric FEM**



- **NURBS-enhanced FEM**



- Introduce a new **reference element**
- Introduce a new **mapping**
- Integrate a new **integration strategy**

NEFEM: NURBS-enhanced finite element method

- Only the boundary of the computational domain is described by a CAD model.
- Basis functions for approximation: polynomials, as in FEM.
- Easy treatment of trimmed and degenerate NURBS surfaces that are common in realistic geometric models.
- Element size purely dictated by the user (the size required to capture properly the physics) and not by the geometric complexity.
- Formulation of your choice: continuous or discontinuous Galerkin…
- Minimum changes in an existing solver: as “easy” (or not) as to define a new element!
- Exact geometry is feasible, without mesh refinement and/or de-featuring

Degree adaptivity procedure

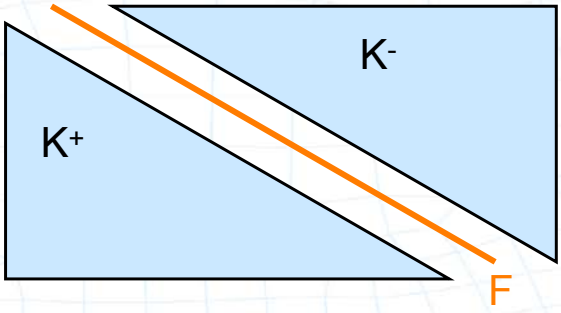
- CHEAP and RELIABLE error estimate

$$E_K^2 = \frac{1}{A_K} \int_K (u_\star - \boldsymbol{u})^2 \, d\Omega$$

HDG superconvergent solution

- Degree update in each element K
  (inspired in [Remacle, Flaherty, Shepard, SIAM Rev. 2003])
  Goal: uniform error distribution

$$\Delta p_K = \left\lceil \log_b(E_K/\epsilon_K) \right\rceil$$

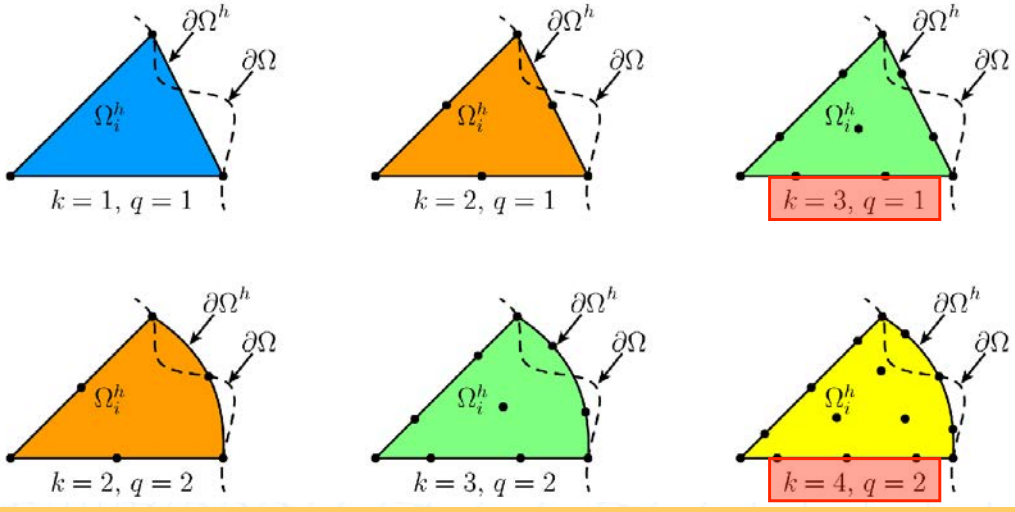


- Degree update for faces
  [Chen, Cockburn, IMANUM 2012]

$$p_F = \max\{p_{K^+}, p_{K^-}\}$$

Boundary representation and degree adaptivity

- To avoid communication with the CAD during the adaptivity procedure, the accuracy of the geometric description is restricted by an initial choice

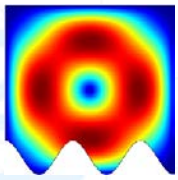


Results in **subparametric formulations** (Geometry approximated with lower order polynomials than the solution)

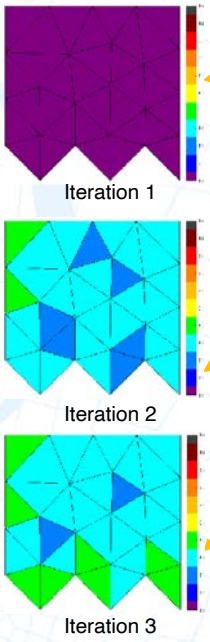
[Sevilla, Huerta. J Sci Comput 2018]

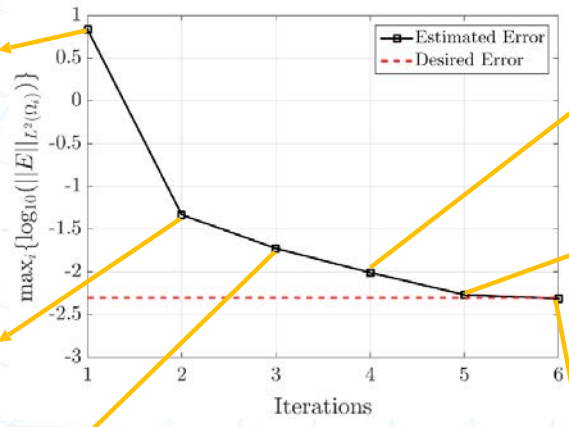
Adaptivity - FEM & no communication with CAD

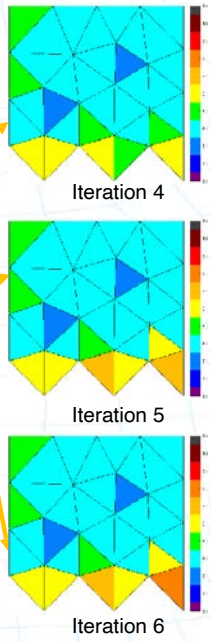
- Initial linear geometry



Stokes flow



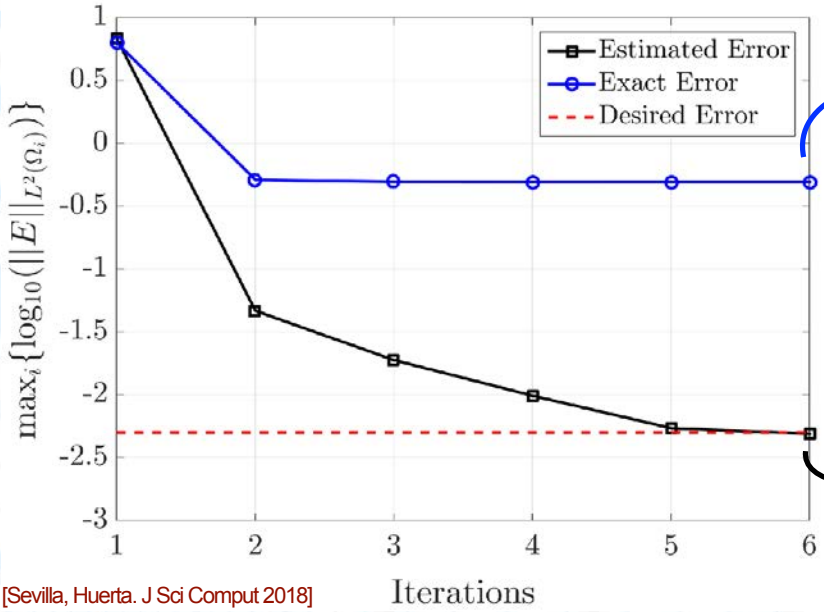


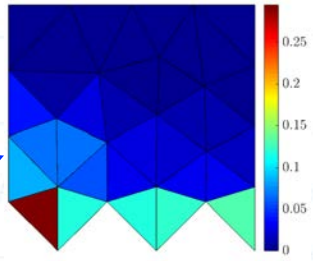


[Sevilla, Huerta. J Sci Comput 2018]

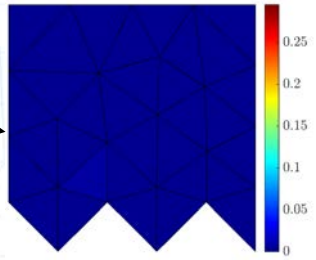
Adaptivity - FEM & no communication with CAD

- Initial linear geometry





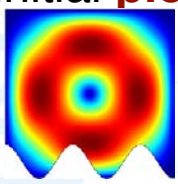
A **linear approximation** of the geometry is **not good enough** for high-order methods!



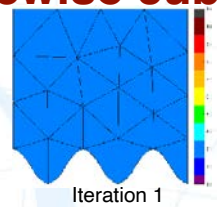
[Sevilla, Huerta. J Sci Comput 2018]

Adaptivity - FEM & no communication with CAD

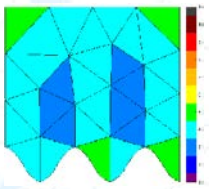
- Initial **piecewise cubic** geometry (“overkill”)



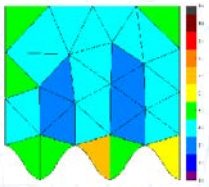
Stokes flow



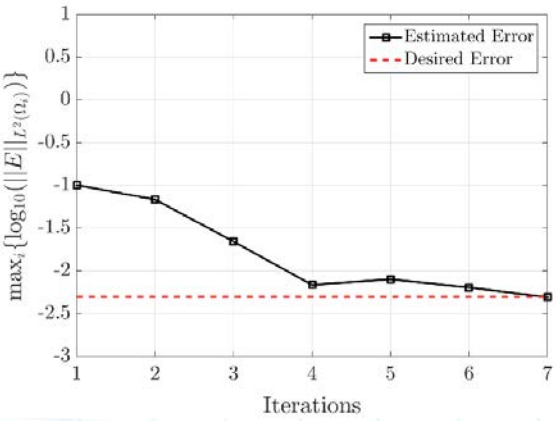
Iteration 1

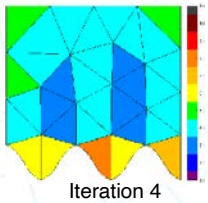


Iteration 2

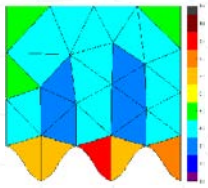


Iteration 3

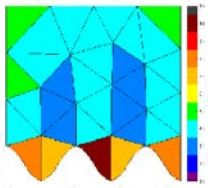




Iteration 4



Iteration 5

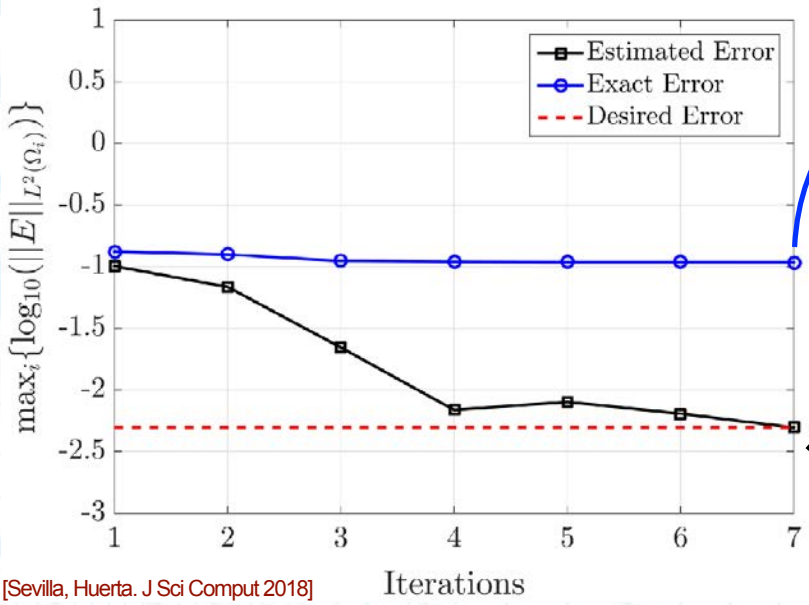


Iteration 6

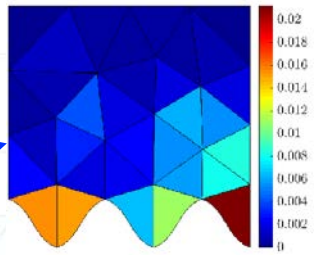
[Sevilla, Huerta. J Sci Comput 2018]

Adaptivity - FEM & no communication with CAD

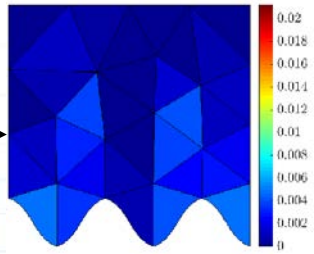
- Initial **piecewise cubic** geometry (“overkill”)



[Sevilla, Huerta. J Sci Comput 2018]

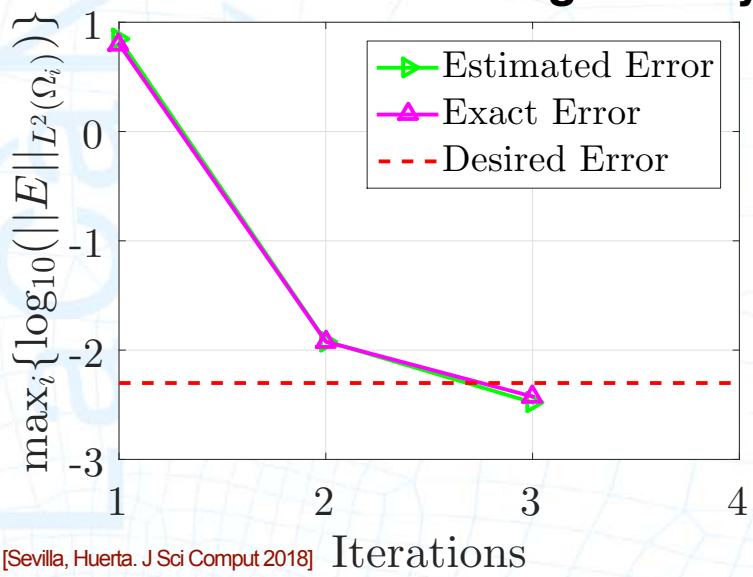


Even **overkill** approximation of the geometry is **not accurate enough**!

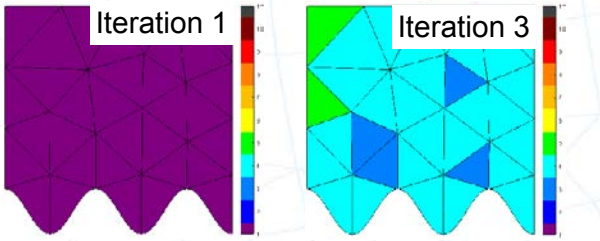


Adaptivity - NEFEM & no communication with CAD

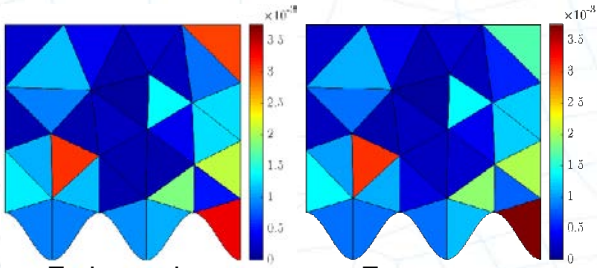
- The solution incorporates the information of the **exact geometry**



[Sevilla, Huerta. J Sci Comput 2018]

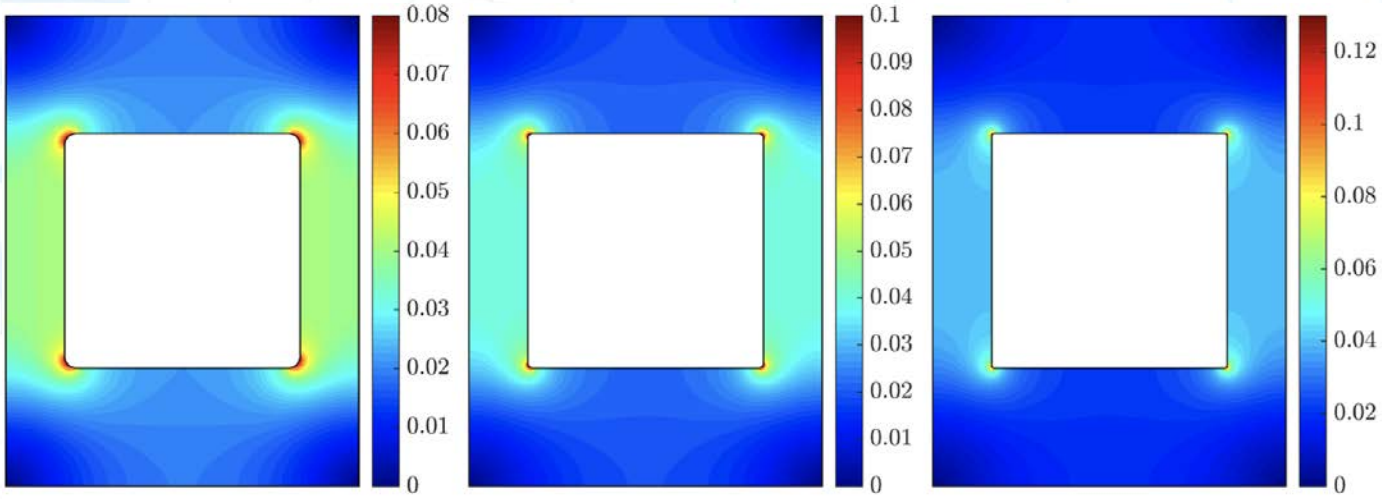


Reliable error estimator with **exact geometry** provided by **NEFEM**



NEFEM and small geometry features

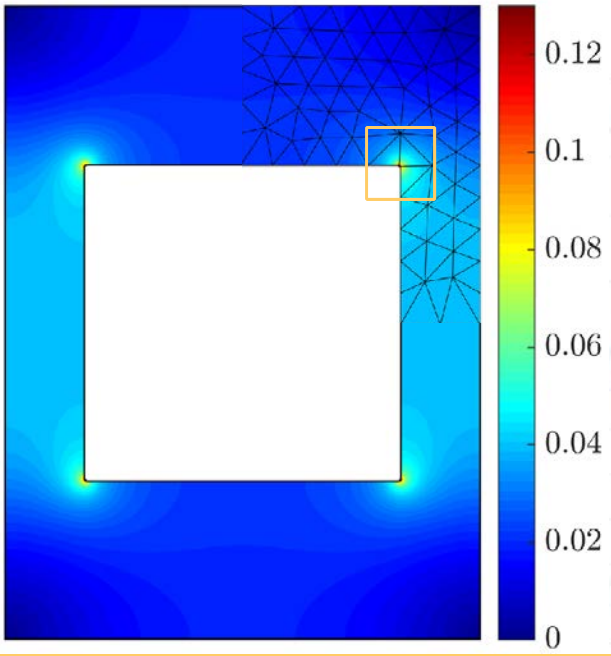
- Rounded corners control the strength of the electric field



[Giacomini, Sevilla. SNAS 2019]

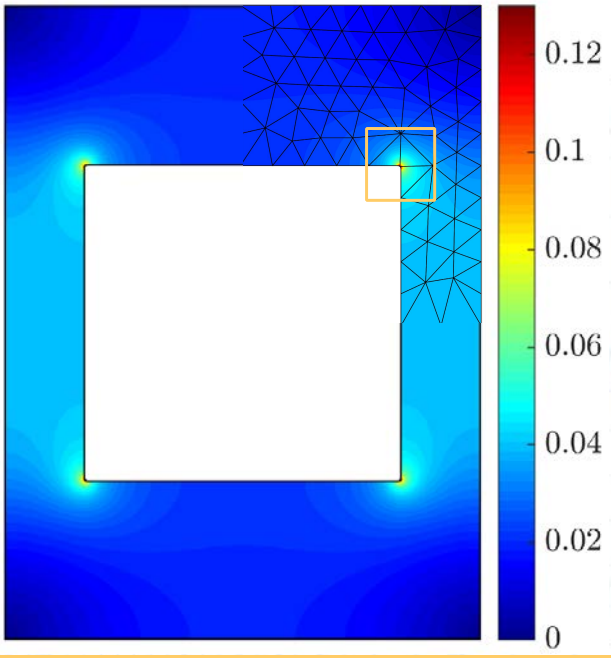
[Sevilla, Huerta, J. Sci. Comput. 2018]
[Sevilla, Comput. Struct. 2019]
[Giacomini, Sevilla, SN Appl. Sci. 2019]

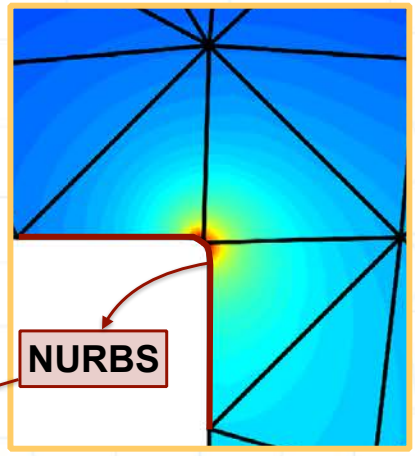
HDG + NURBS-enhanced FEM



[Sevilla, Huerta, J. Sci. Comput. 2018]
[Sevilla, Comput. Struct. 2019]
[Giacomini, Sevilla, SN Appl. Sci. 2019]

HDG + NURBS-enhanced FEM





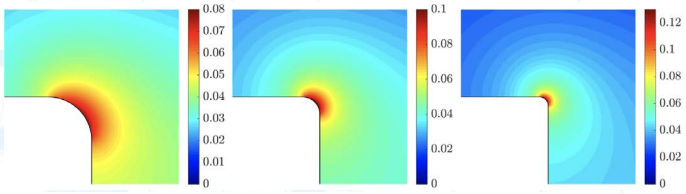
[Sevilla, Fernández-Méndez, Huerta, ACME 2011]

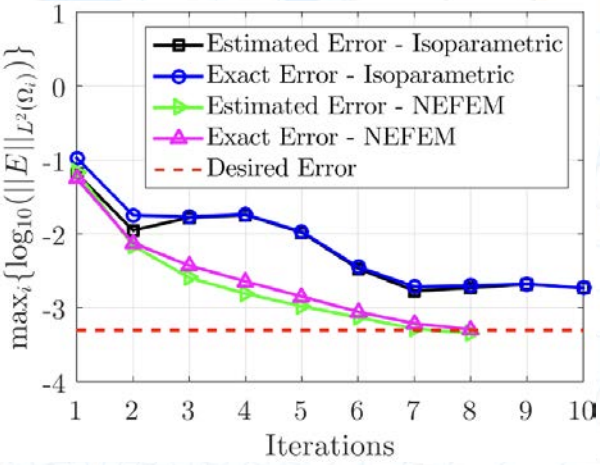
Mesh **size independent**
of geometric features

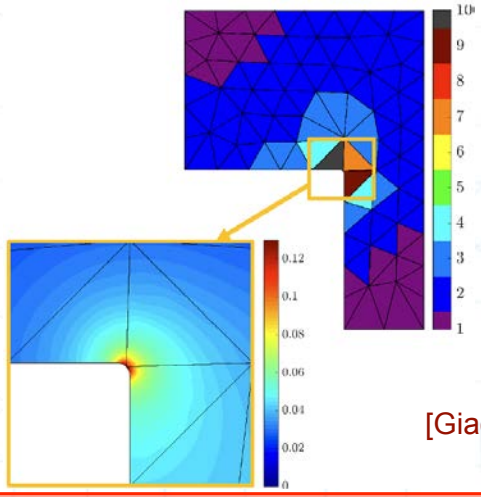
Adaptive degree to control accuracy

NEFEM and small geometry features

Zoom of the corners:

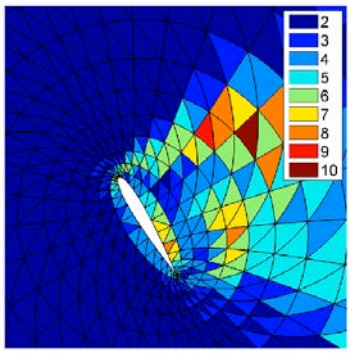






[Giacomini, Sevilla. SNAS 2019]

NEFEM outperforms isoparametric elements (with CAD communication in each iteration) and allows meshes with **element size independent of geometric features**



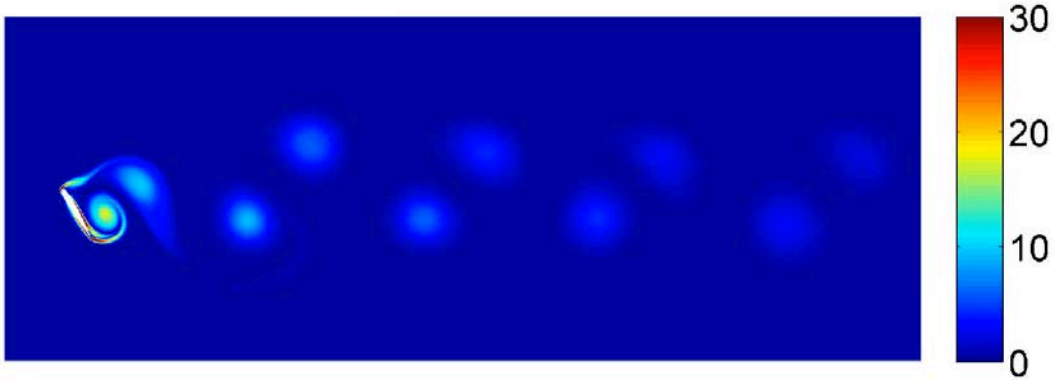
Viscous incompressible flow

Incompressible Navier-Stokes

$Re = 10{,}000$

$\alpha = 2^{o}$

[Giorgiani. PhD thesis (2013)]



High-order: HDG contributions

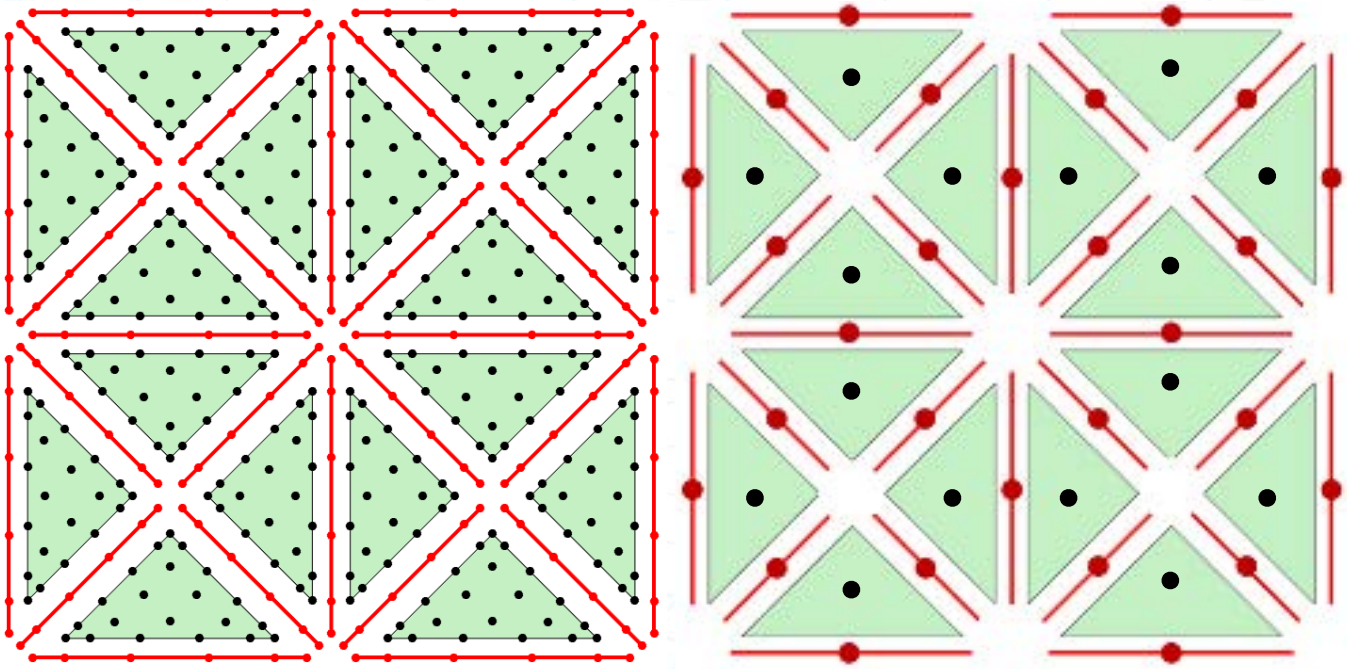
- ➜ High-fidelity **(convergence of order $k+1$ for all variables)**
- ➜ Accurate treatment of convection **(Riemann solvers)**
- ➜ Exact treatment of geometry **(NEFEM)**
- ➜ Reduced number of DOFs with respect to DG **(hybridisation)**
- ➜ Robust in the incompressible limit (**no pressure correction**)

[Giacomini, Karkoulias, Sevilla, Huerta, J Sci Comput 2018] - Stokes
[Sevilla, Giacomini, Karkoulias, Huerta, IJNME 2018] - Elasticity
[Giacomini, Sevilla, Huerta, CISM 2020] - Tutorial NS
[La Spina, Giacomini, Huerta, Comput Mech 2020] - Coupling HDG/CG
[La Spina, Kronbichler, Giacomini, Wall, Huerta, CMAME 2020] - FSI
[Giacomini, Sevilla, Huerta, ACME 2021] - HDGlab
[Vila-Pérez, Giacomini, Sevilla, Huerta, ACME 2021] - Compressible flow
[Piccardo, Giacomini, Huerta, JCP 2024] - Unfitted HDG with NURBS
[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026] - Characteristic artificial BC

From high to low order: face-centred finite volume (FCFV) method

What if
the **solution** and its **gradient**
are approximated
in each element using
piecewise constant functions?

from HDG p=5 to Face-Centered Finite Volume p=0



Low-order (HDG p=0): FCFV contributions

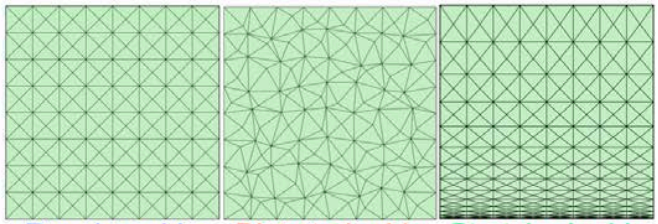
- ➜ First-order convergence of stress/flux (**without reconstruction**)
- ➜ Accurate approximation of aerodynamic forces (**not sensitive to mesh distortion and stretching**)
- ➜ Robust in the incompressible limit (**no pressure correction**)
- ➜ Non-oscillatory discontinuous approximations (**no shock capturing**)

[Sevilla, Giacomini, Huerta, IJNME 2018] - Face Centered FV (Poisson & Stokes)
[Sevilla, Giacomini, Huerta, Comput. Struct. 2019] - Elasticity
[Vieira, Giacomini, Sevilla, Huerta, CMAME 2020] - 2nd order for primal
[Giacomini, Sevilla, IJNME 2020] - 2nd order on hybrid meshes
[Vila-Pérez, Giacomini, Sevilla, Huerta, Comput. Fluids 2022] - Compressible flow
[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023] - Benchmarking wrt Fluent
[Vieira, Giacomini, Sevilla, Huerta, Comput. Fluids 2024] - Laminar & turbulent incompressible
[Giacomini, Cortellessa, Vieira, Sevilla, Huerta, IJNME 2025] - Hybrid pressure FCFV
[Cortellessa, Giacomini, Huerta, LNCSE 2025] - FCFV in OpenFOAM

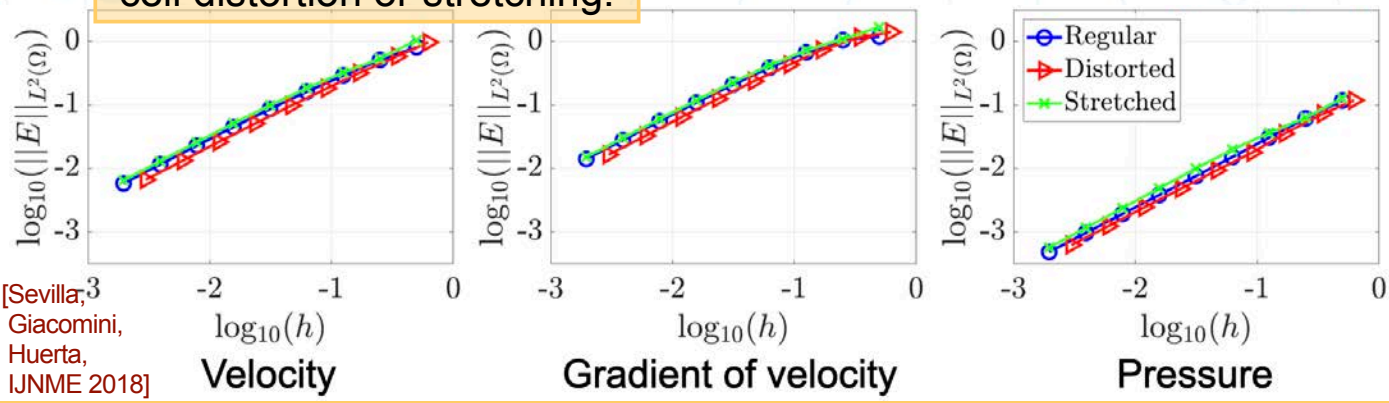
FCFV convergence properties

Stokes flow (analytical solution)

- **First-order convergence** of velocity, gradient of velocity and pressure
- **No** need for **flux reconstruction**

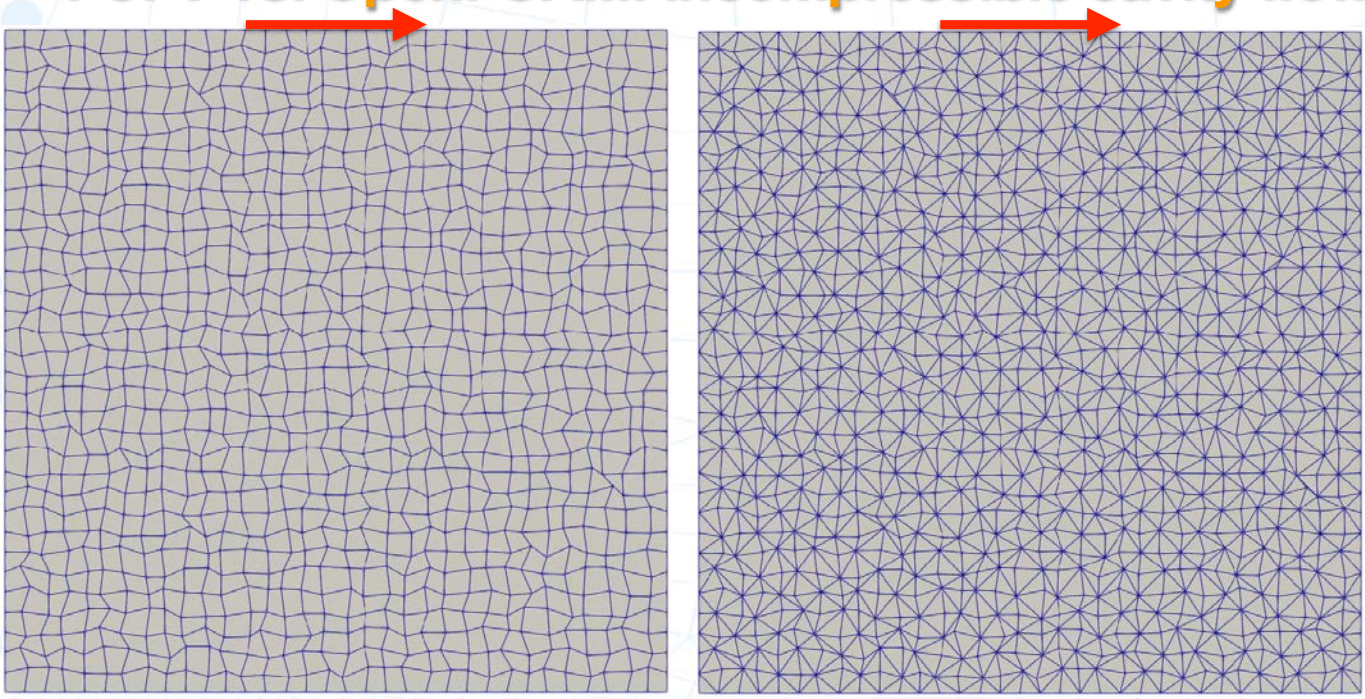


No loss of accuracy due to cell distortion or stretching!



[Sevilla, Giacomini, Huerta, IJNME 2018]

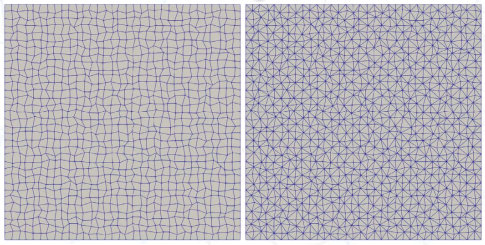
FCFV vs. OpenFOAM: incompressible cavity flow

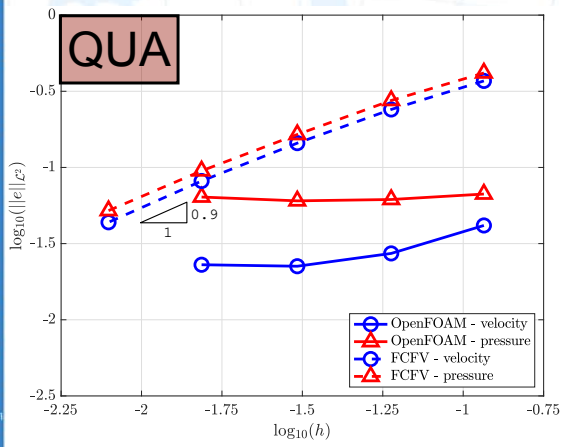


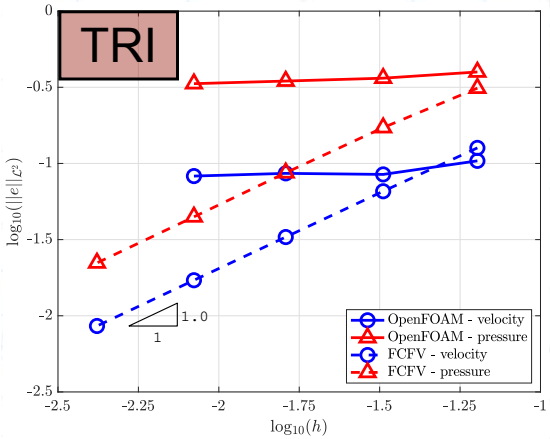
Benchmarking with SimpleFoam

Incompressible Navier-Stokes flow

- Cavity flow
- **Re=100**







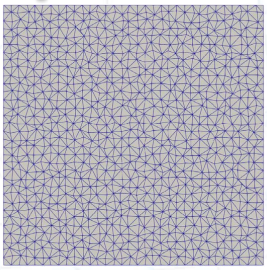
FCFV in **OpenFOAM!**

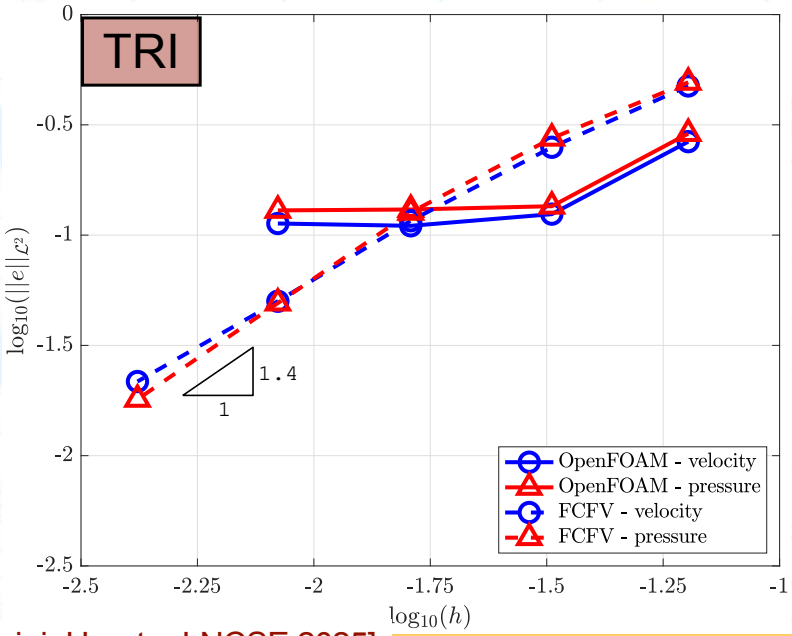
[Cortellessa, Giacomini, Huerta, LNCSE 2025]

Benchmarking with SimpleFoam

Incompressible Navier-Stokes flow

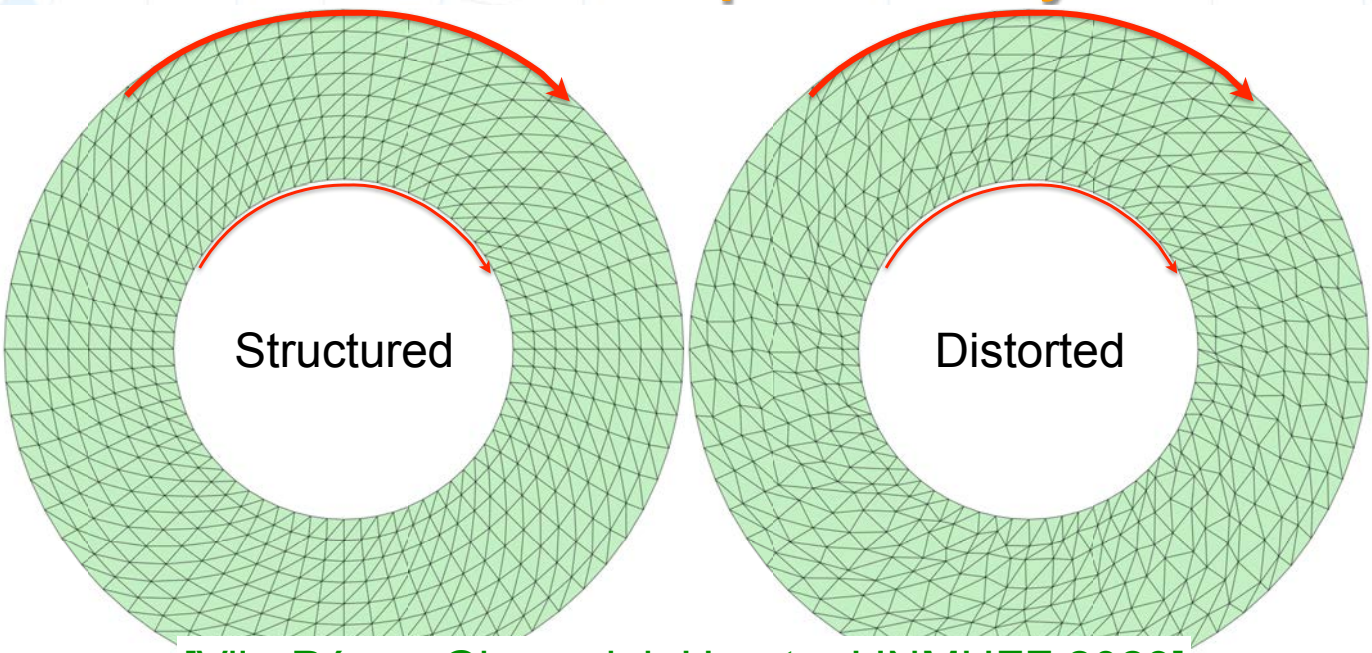
- Cavity flow
- **Re=1000**





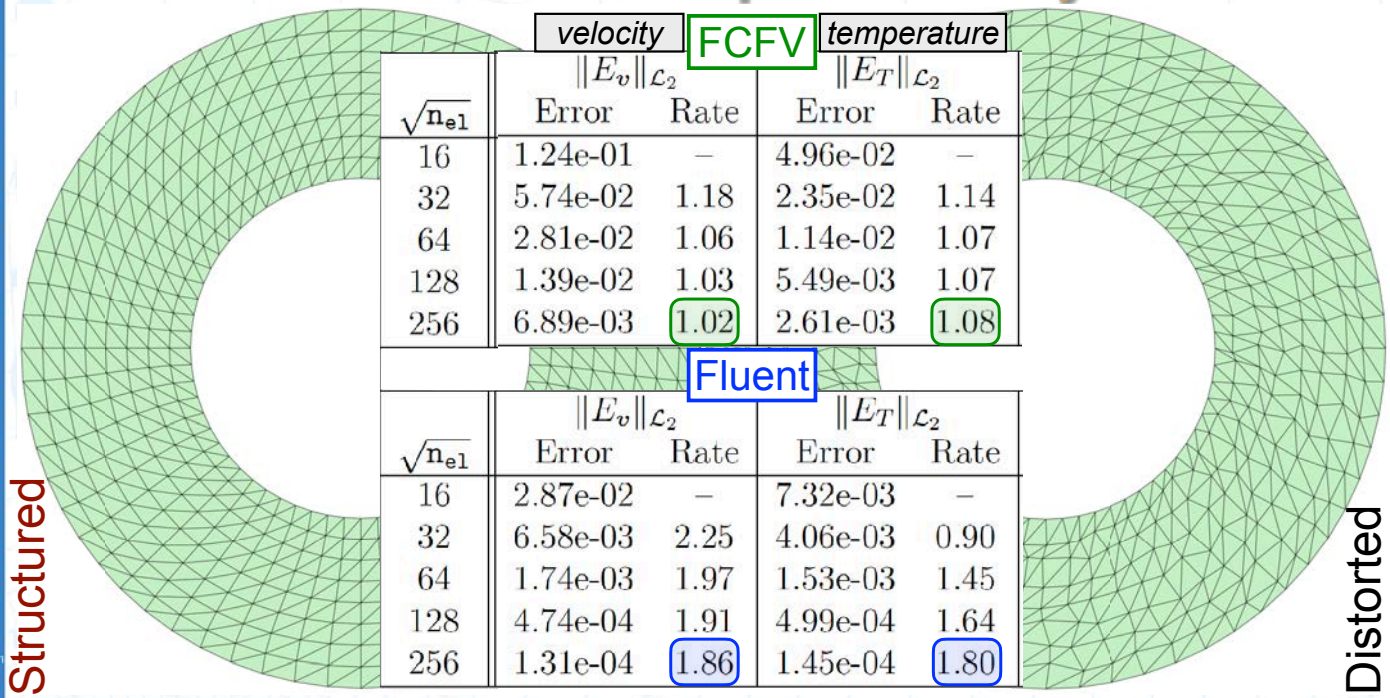
[Cortellessa, Giacomini, Huerta, LNCSE 2025]

FCFV vs FLUENT: compressible Taylor-Couette



[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023]

FCFV vs FLUENT: compressible Taylor-Couette



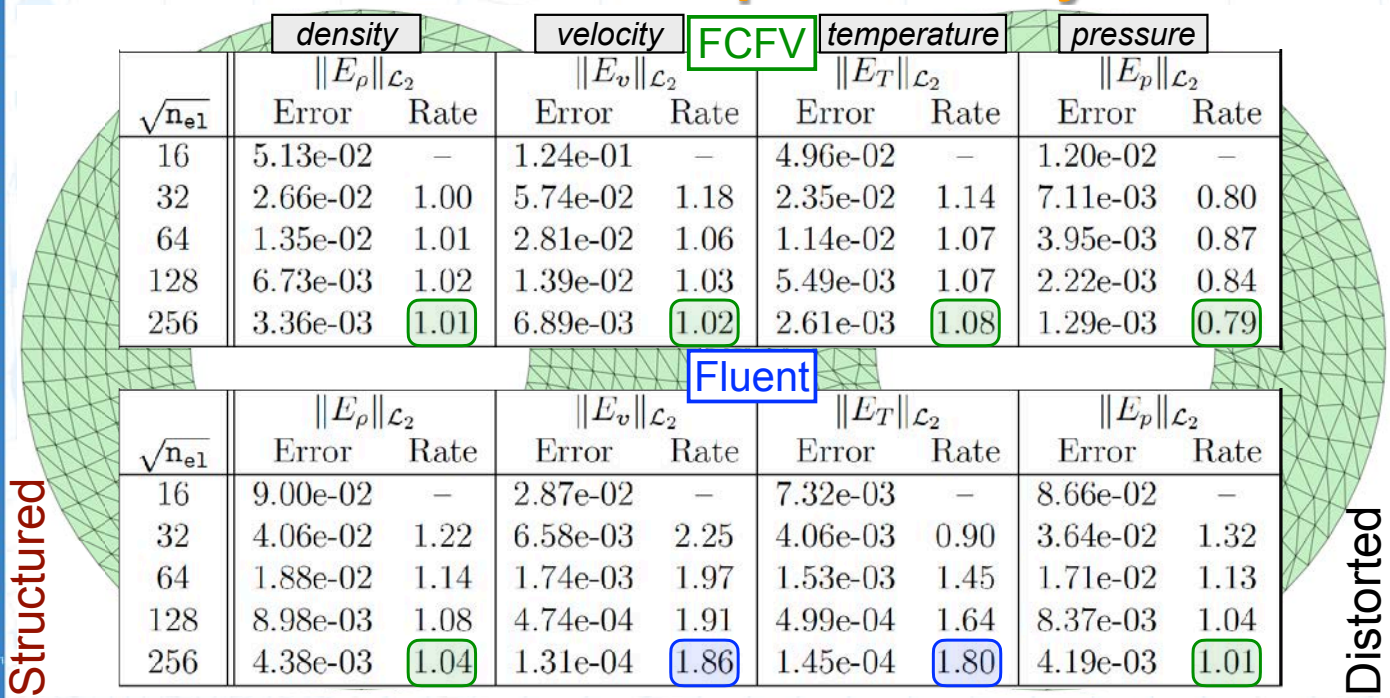
velocity | FCFV | *temperature*

$\sqrt{n_{el}}$	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate
16	1.24e-01	–	4.96e-02	–
32	5.74e-02	1.18	2.35e-02	1.14
64	2.81e-02	1.06	1.14e-02	1.07
128	1.39e-02	1.03	5.49e-03	1.07
256	6.89e-03	1.02	2.61e-03	1.08

Fluent

$\sqrt{n_{el}}$	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate
16	2.87e-02	–	7.32e-03	–
32	6.58e-03	2.25	4.06e-03	0.90
64	1.74e-03	1.97	1.53e-03	1.45
128	4.74e-04	1.91	4.99e-04	1.64
256	1.31e-04	1.86	1.45e-04	1.80

FCFV vs FLUENT: compressible Taylor-Couette



density | *velocity* | FCFV | *temperature* | *pressure*

$\sqrt{n_{el}}$	$\|E_\rho\|_{\mathcal{L}_2}$ Error	Rate	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate	$\|E_p\|_{\mathcal{L}_2}$ Error	Rate
16	5.13e-02	–	1.24e-01	–	4.96e-02	–	1.20e-02	–
32	2.66e-02	1.00	5.74e-02	1.18	2.35e-02	1.14	7.11e-03	0.80
64	1.35e-02	1.01	2.81e-02	1.06	1.14e-02	1.07	3.95e-03	0.87
128	6.73e-03	1.02	1.39e-02	1.03	5.49e-03	1.07	2.22e-03	0.84
256	3.36e-03	1.01	6.89e-03	1.02	2.61e-03	1.08	1.29e-03	0.79

Fluent

$\sqrt{n_{el}}$	$\|E_\rho\|_{\mathcal{L}_2}$ Error	Rate	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate	$\|E_p\|_{\mathcal{L}_2}$ Error	Rate
16	9.00e-02	–	2.87e-02	–	7.32e-03	–	8.66e-02	–
32	4.06e-02	1.22	6.58e-03	2.25	4.06e-03	0.90	3.64e-02	1.32
64	1.88e-02	1.14	1.74e-03	1.97	1.53e-03	1.45	1.71e-02	1.13
128	8.98e-03	1.08	4.74e-04	1.91	4.99e-04	1.64	8.37e-03	1.04
256	4.38e-03	1.04	1.31e-04	1.86	1.45e-04	1.80	4.19e-03	1.01

FCFV vs FLUENT: compressible Taylor-Couette

density | *velocity* | FCFV | *temperature* | *pressure*

$\sqrt{n_{el}}$	$\|E_\rho\|_{\mathcal{L}_2}$ Error	Rate	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate	$\|E_p\|_{\mathcal{L}_2}$ Error	Rate
16	6.13e-02	–	1.68e-01	–	6.35e-02	–	1.34e-02	–
32	3.43e-02	0.98	7.65e-02	1.32	3.24e-02	1.13	7.53e-03	0.97
64	1.88e-02	0.86	3.84e-02	0.99	1.71e-02	0.92	3.94e-03	0.93
128	9.41e-03	1.10	1.92e-02	1.10	8.35e-03	1.14	2.23e-03	0.91
256	4.93e-03	0.94	9.54e-03	1.01	4.16e-03	1.01	1.32e-03	0.76

Fluent

$\sqrt{n_{el}}$	$\|E_\rho\|_{\mathcal{L}_2}$ Error	Rate	$\|E_v\|_{\mathcal{L}_2}$ Error	Rate	$\|E_T\|_{\mathcal{L}_2}$ Error	Rate	$\|E_p\|_{\mathcal{L}_2}$ Error	Rate
16	7.72e-02	–	3.35e-02	–	9.21e-03	–	7.36e-02	–
32	3.24e-02	1.46	1.14e-02	1.81	5.55e-03	0.85	2.77e-02	1.64
64	1.47e-02	1.13	4.83e-03	1.23	2.61e-03	1.08	1.25e-02	1.15
128	6.93e-03	1.20	2.35e-03	1.14	1.11e-03	1.35	6.15e-03	1.12
256	3.25e-03	1.10	1.12e-03	1.07	5.02e-04	1.16	2.96e-03	1.06

Structured

Distorted

FCFV vs FLUENT: compressible Taylor-Couette

viscous stress tensor | *heat flux*

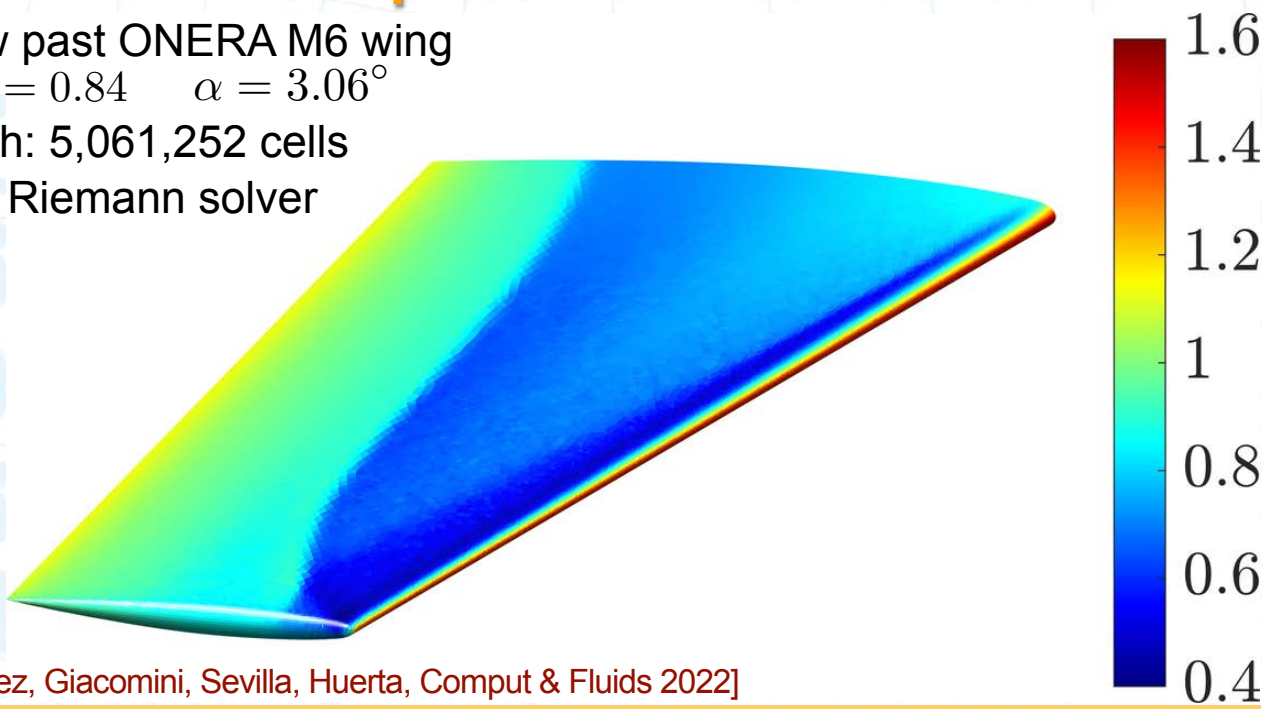
$\sqrt{n_{el}}$	$\|E_{\boldsymbol{\sigma}^d}\|_{\mathcal{L}_2}$ FCFV Error	Rate	$\|E_{\boldsymbol{\sigma}^d}\|_{\mathcal{L}_2}$ Fluent Error	Rate	$\|E_{\boldsymbol{q}}\|_{\mathcal{L}_2}$ FCFV Error	Rate	$\|E_{\boldsymbol{q}}\|_{\mathcal{L}_2}$ Fluent Error	Rate
16	7.56e-1	–	1.12e-1	–	3.67e-1	–	6.47e-2	–
32	4.30e-1	0.95	4.52e-2	1.52	1.86e-1	1.14	3.65e-2	0.96
64	2.52e-1	0.77	2.61e-2	0.79	9.86e-2	0.92	1.74e-2	1.07
128	1.49e-1	0.83	1.84e-2	0.55	5.10e-2	1.05	9.42e-3	0.97
256	8.69e-2	0.78	1.42e-2	0.38	2.63e-2	0.96	6.99e-3	0.43

Structured

Distorted

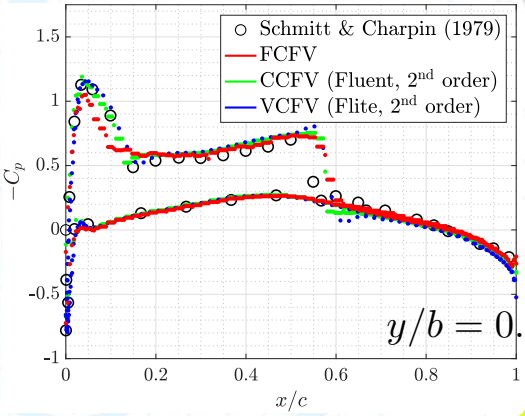
Compressible inviscid transonic flow

- Flow past ONERA M6 wing
  $M_\infty = 0.84 \quad \alpha = 3.06^\circ$
- Mesh: 5,061,252 cells
- HLL Riemann solver

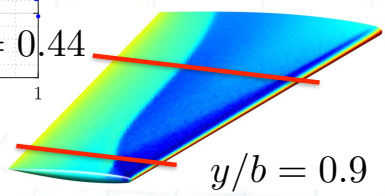


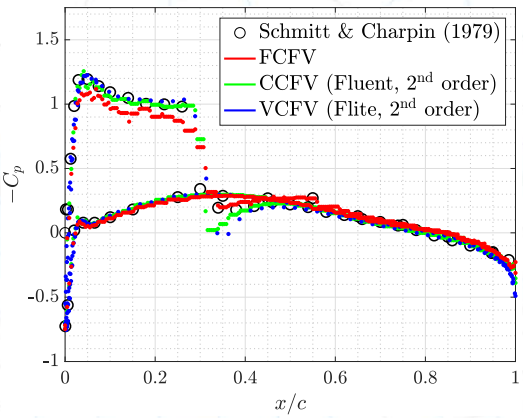
[Vila-Pérez, Giacomini, Sevilla, Huerta, Comput & Fluids 2022]

Compressible inviscid transonic flow



Accuracy **comparable with second-order** finite volume solvers





Non-oscillatory approximation of the **shock wave**

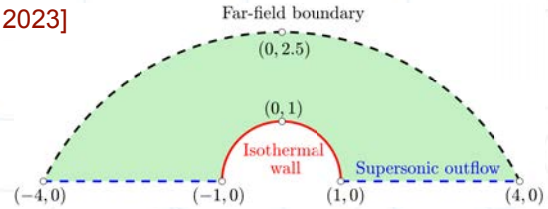
[Vila-Pérez, Giacomini, Sevilla, Huerta, Comput & Fluids 2022]

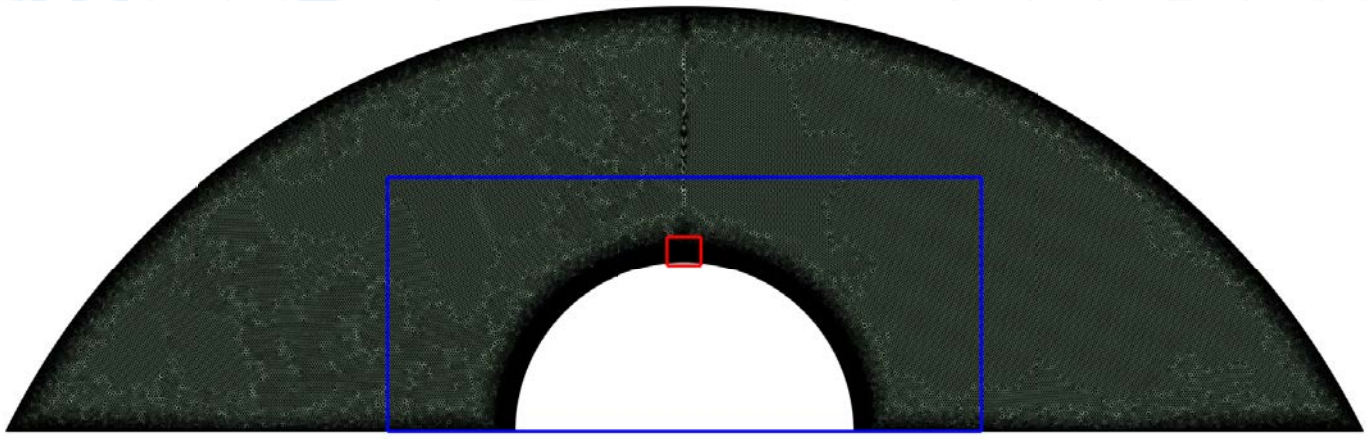
Supersonic laminar flow over a cylinder

[Vila-Pérez, Giacomini, Huerta. IJNMHFF, 2023]

Flow conditions

$$M_\infty = 4$$
$$Re = 10^4$$



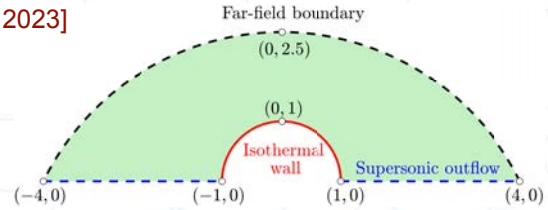


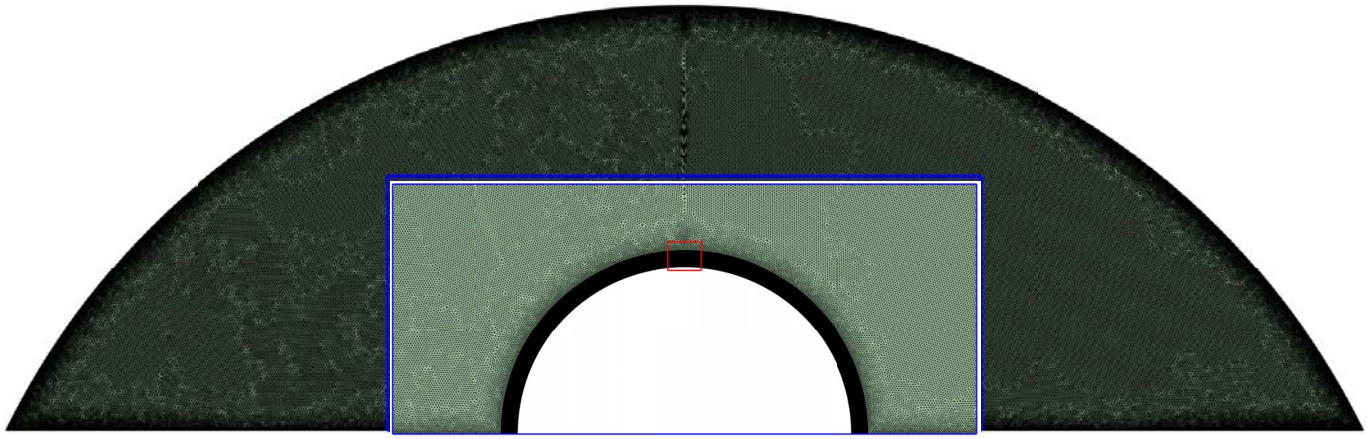
Supersonic laminar flow over a cylinder

[Vila-Pérez, Giacomini, Huerta. IJNMHFF, 2023]

Flow conditions

$$M_\infty = 4$$
$$Re = 10^4$$





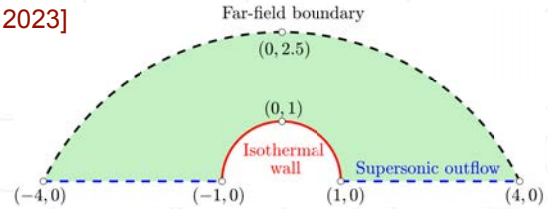
Supersonic laminar flow over a cylinder

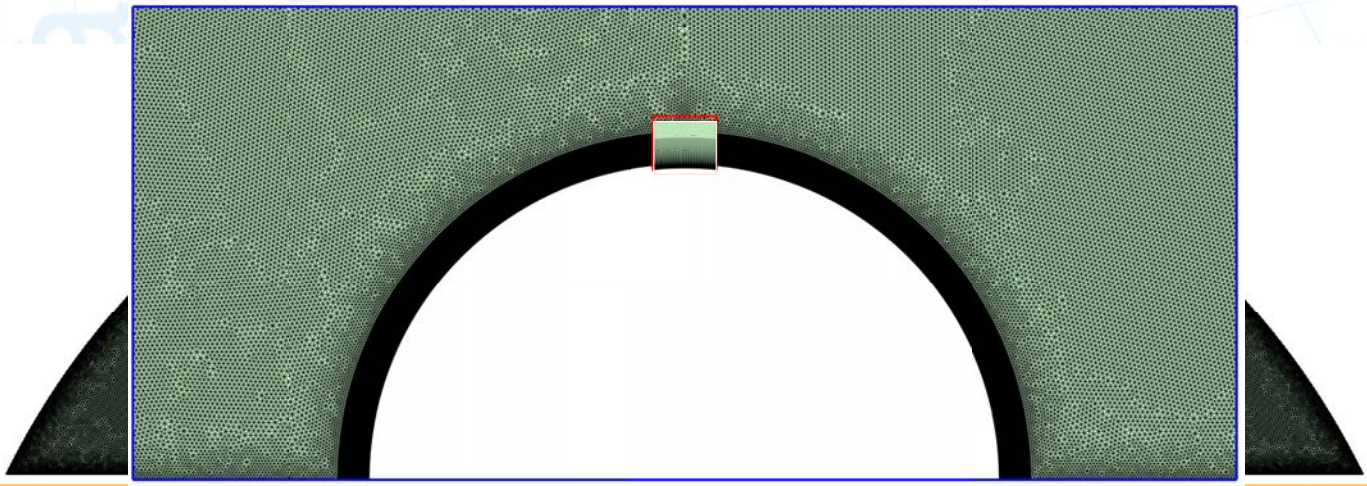
[Vila-Pérez, Giacomini, Huerta. IJNMHFF, 2023]

Flow conditions

$M_\infty = 4$

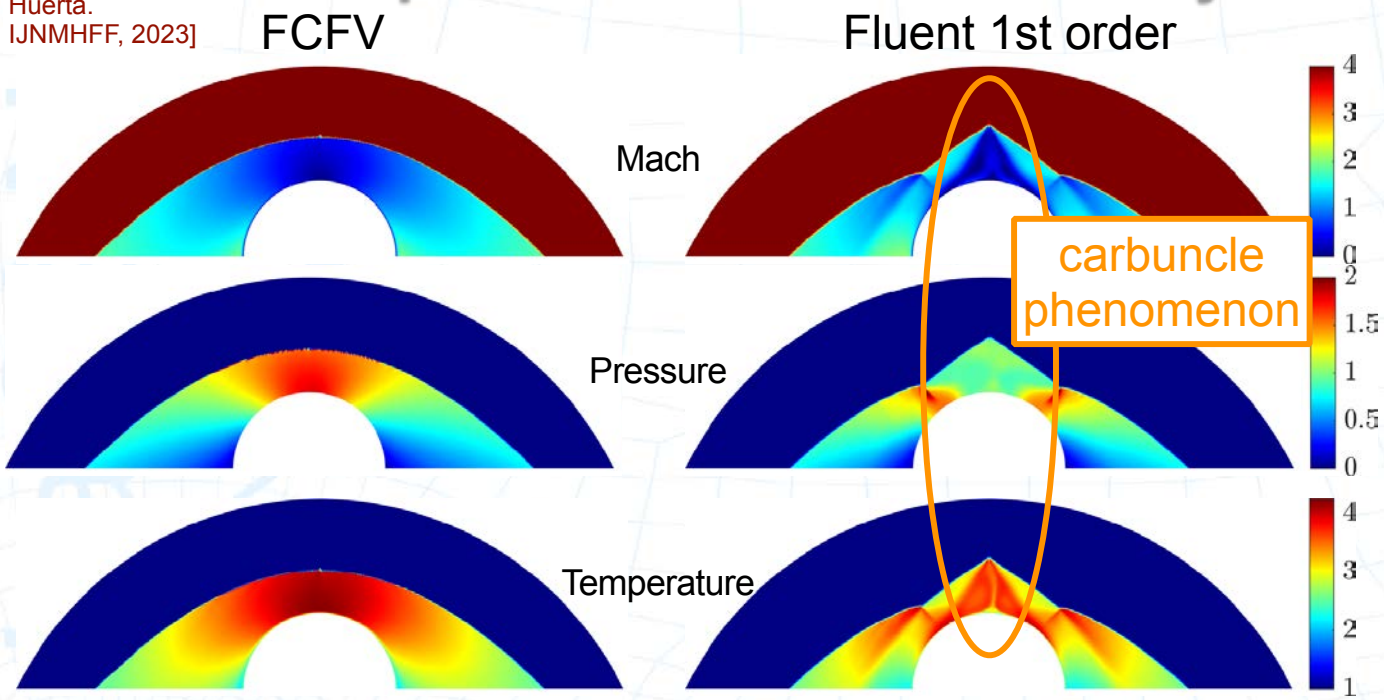
$Re = 10^4$





[Vila-Pérez, Giacomini, Huerta. IJNMHFF, 2023]

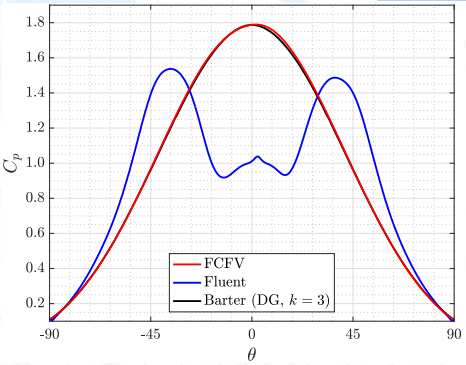
Supersonic laminar flow over a cylinder



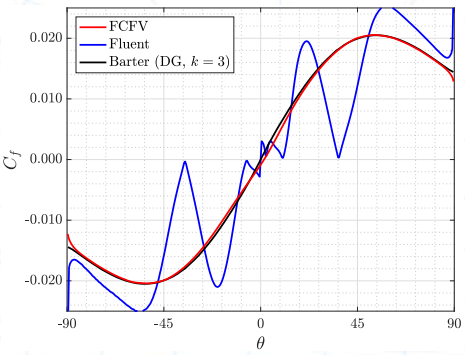
[Vila-Pérez, Giacomini, Huerta. IJNMHFF, 2023]

Supersonic laminar flow over a cylinder

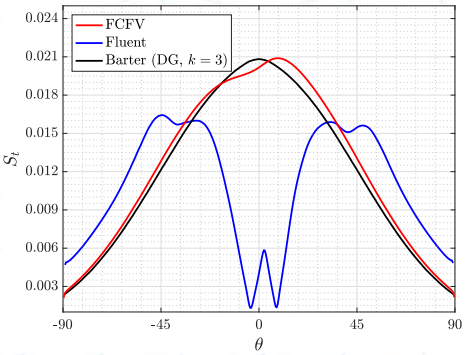
Wall Quantities of Interest



Pressure coefficient



Skin friction coefficient



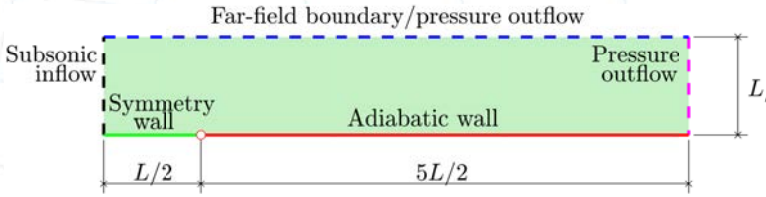
Stanton number

Laminar flow over a flat plate

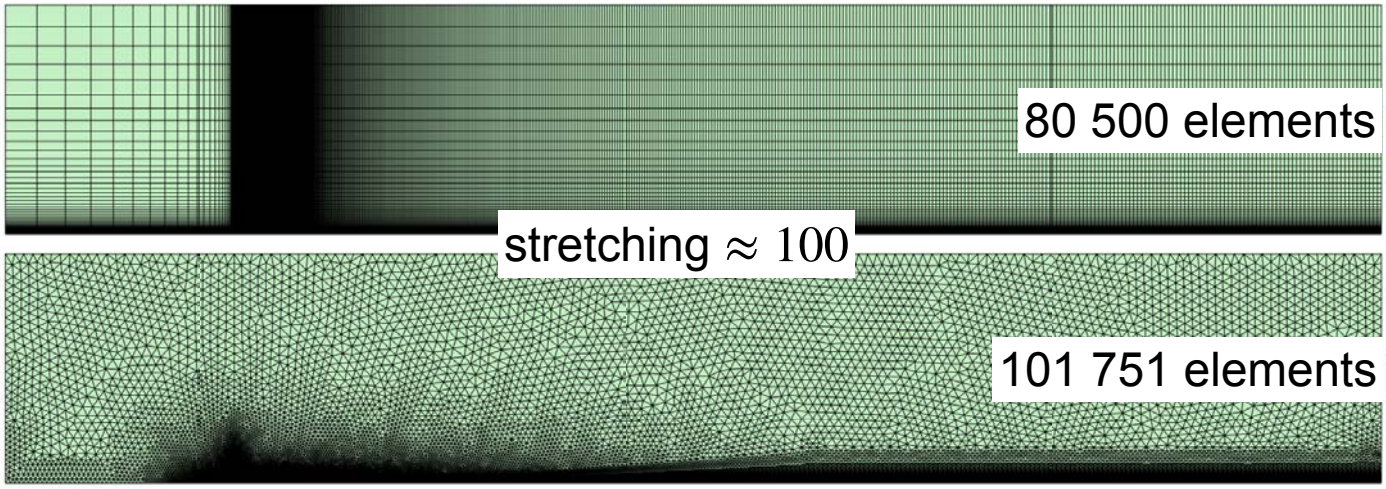
Flow conditions

$M_\infty = 0.1$

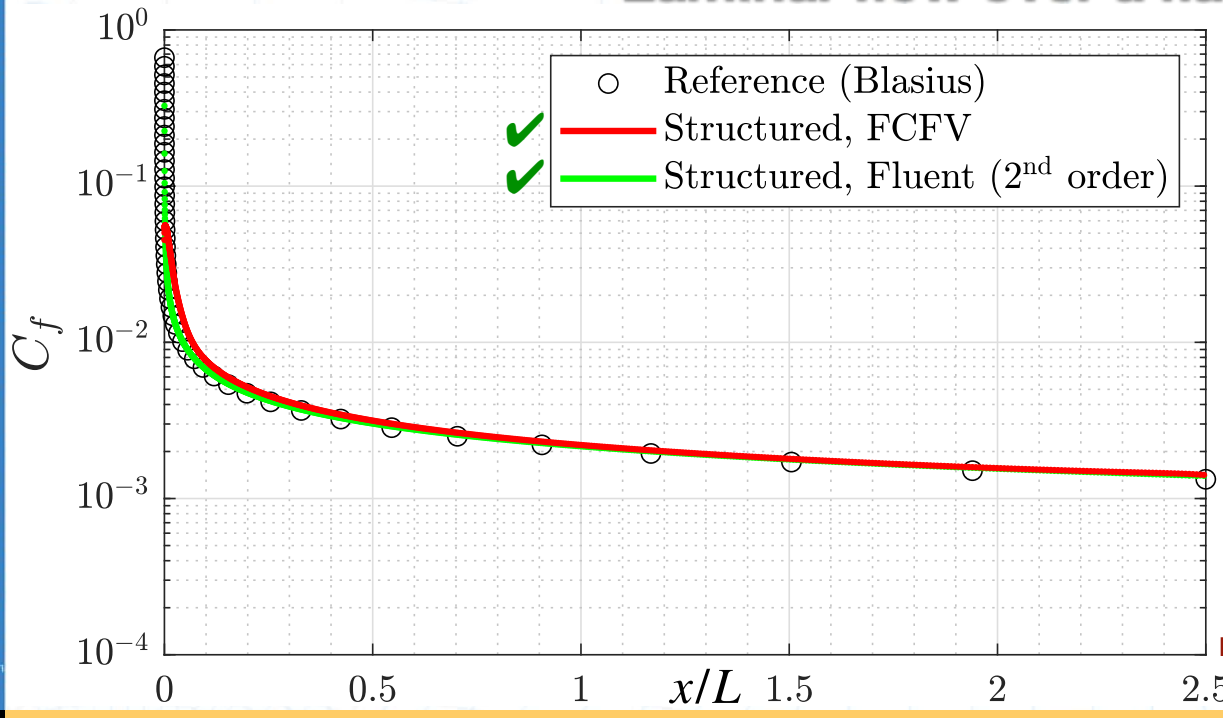
$Re = 10^5$



[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023]

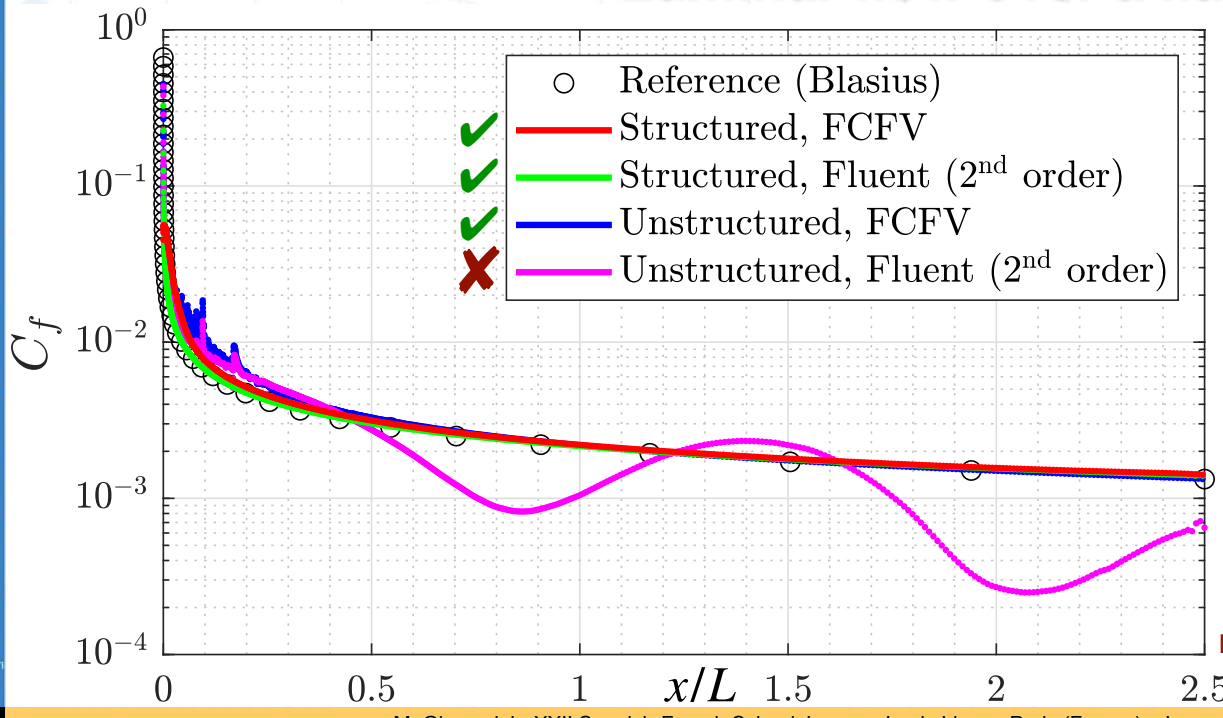


Laminar flow over a flat plate



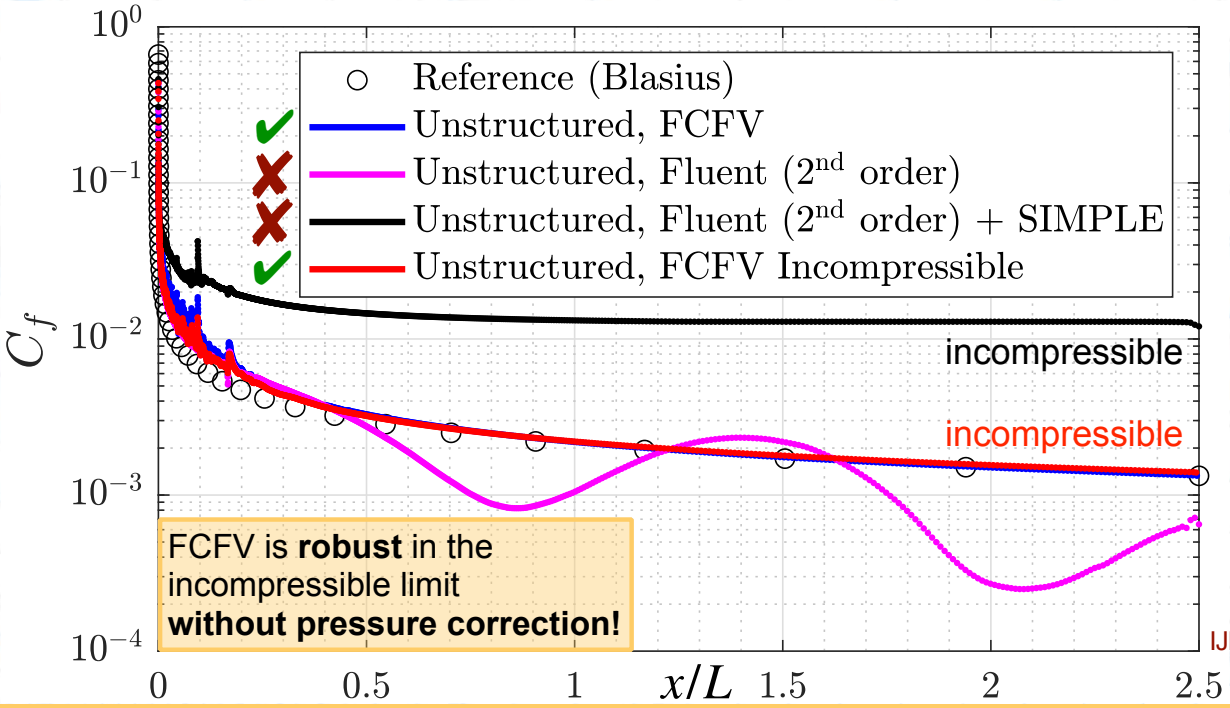
[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023]

Laminar flow over a flat plate



[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023]

Laminar flow over a flat plate



FCFV is **robust** in the incompressible limit **without pressure correction!**

[Vila-Pérez, Giacomini, Huerta, IJNMHFF 2023]

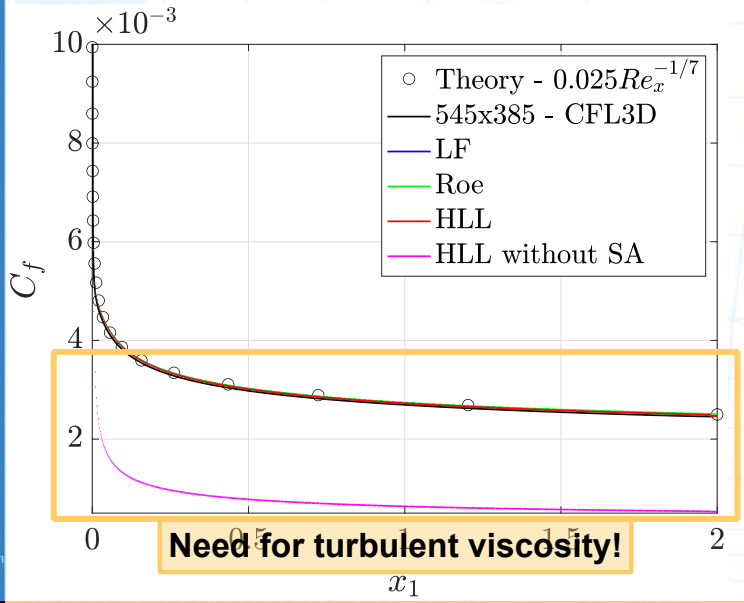
Turbulent flow over a flat plate

- RANS with **Spalart-Allmaras** model

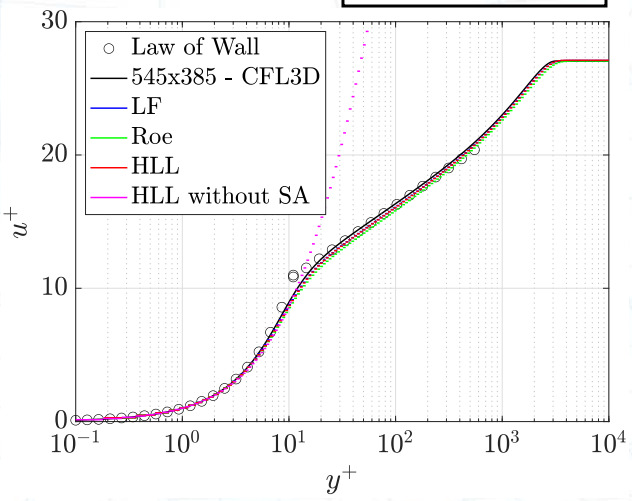
[Vieira, Giacomini, Sevilla, Huerta, Comput & Fluids 2024]

Flow conditions

$Re = 5 \times 10^6$



Need for turbulent viscosity!



Dimensionless velocity profile

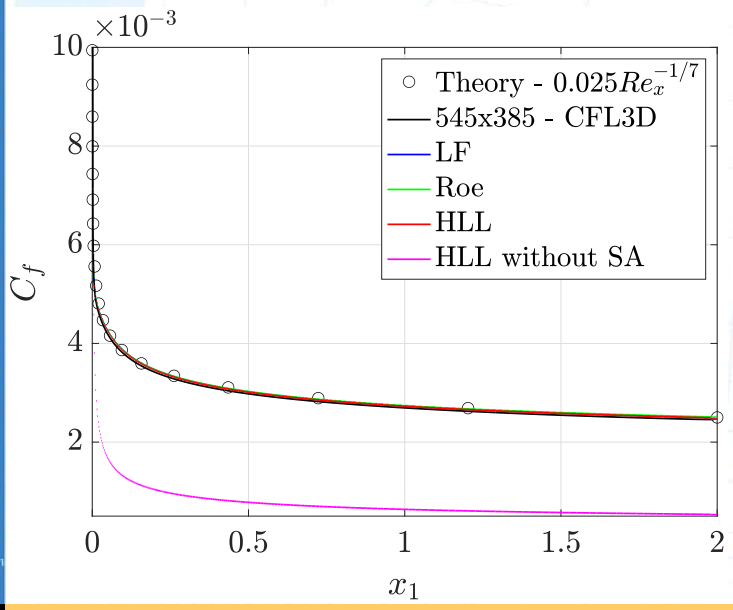
Turbulent flow over a flat plate

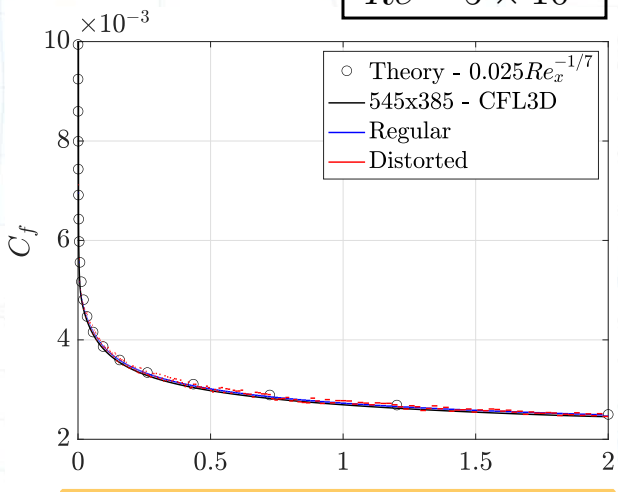
- RANS with **Spalart-Allmaras** model

[Vieira, Giacomini, Sevilla, Huerta, Comput & Fluids 2024]

Flow conditions

$Re = 5 \times 10^6$

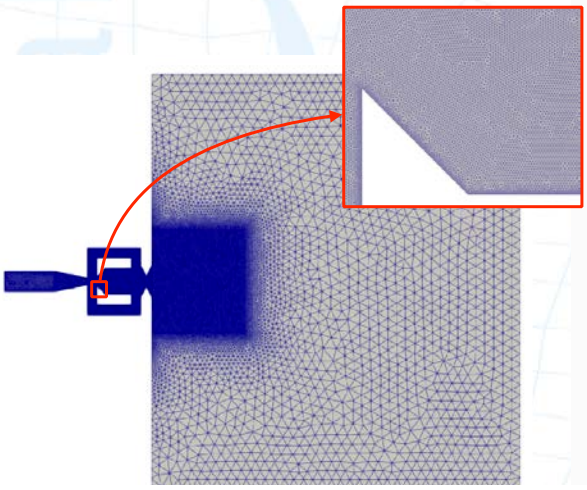


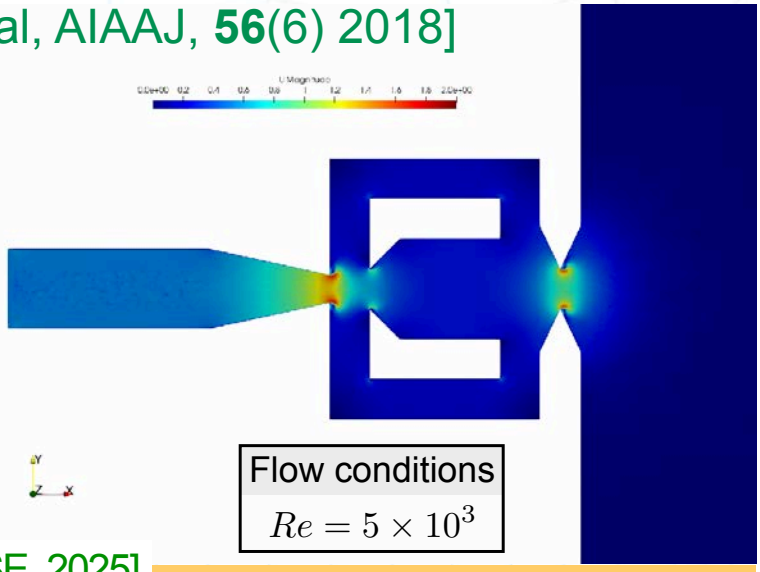


FCFV is **robust to grid distortion** also in the turbulent case

Transient incompressible flows in a fluidic oscillator

- Triangular mesh automatically generated (GiD & Delaunay)
- Low non-orthogonality, high face skewness
- Inspired by [Seo, Zhu & Mittal, AIAAJ, **56**(6) 2018]





Flow conditions

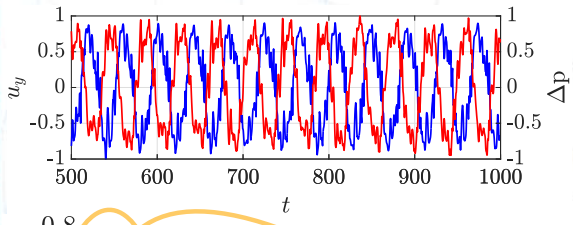
$Re = 5 \times 10^3$

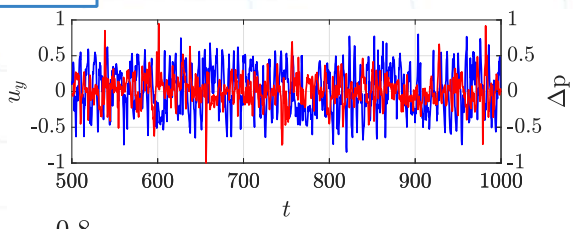
[Cortellessa, Giacomini, Huerta, LNCSE, 2025]

Transient incompressible flows in a fluidic oscillator

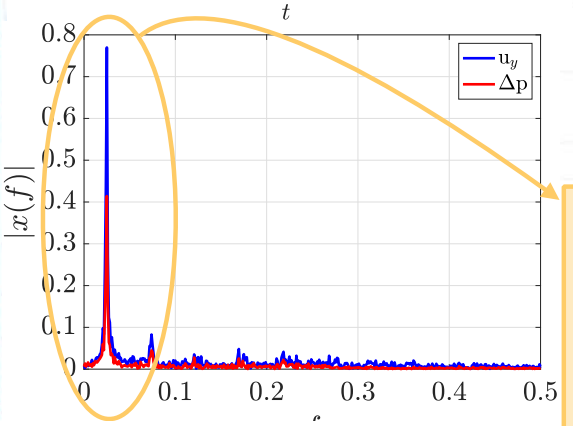
FCFV | OpenFOAM

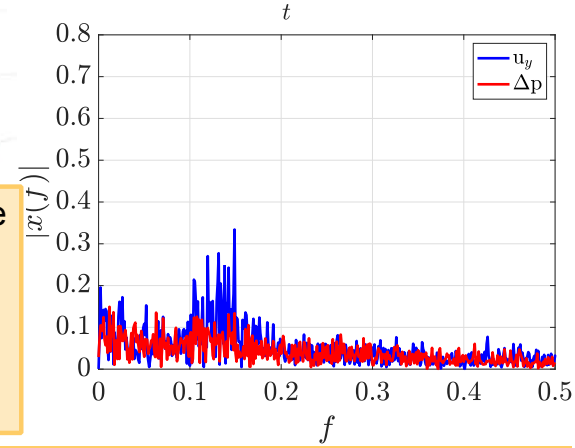
Temporal evolution of velocity and pressure





Frequencies spectra

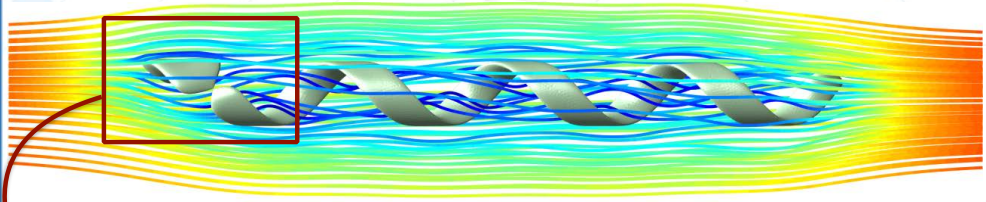




FCFV is capable of capturing dominant frequency even on **highly skewed mesh**

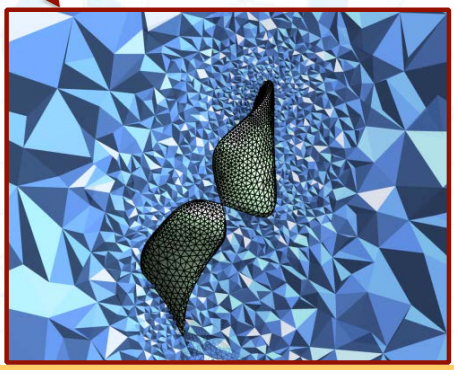
[Cortellessa, Giacomini, Huerta, LNCSE, 2025]

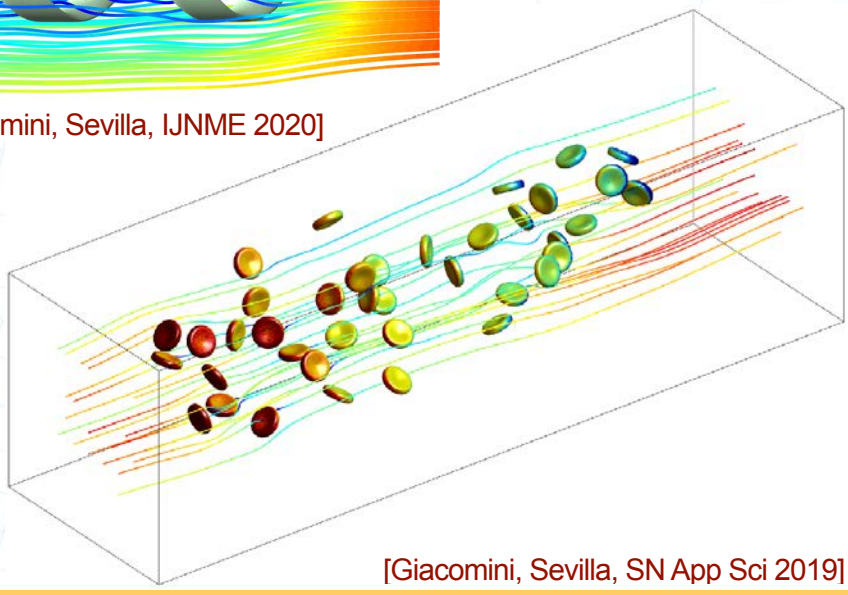
Complex geometries with localised features



Need for **automatically adapted** meshes!

[Giacomini, Sevilla, IJNME 2020]





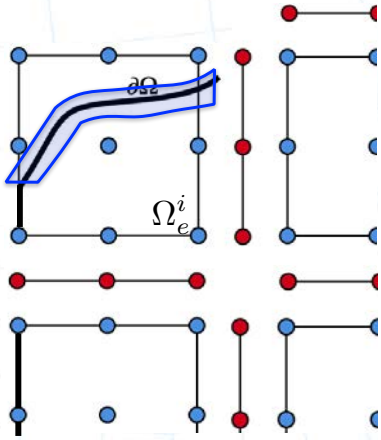
[Giacomini, Sevilla, SN App Sci 2019]

Ongoing research

Unfitted HDG with exact NURBS boundaries

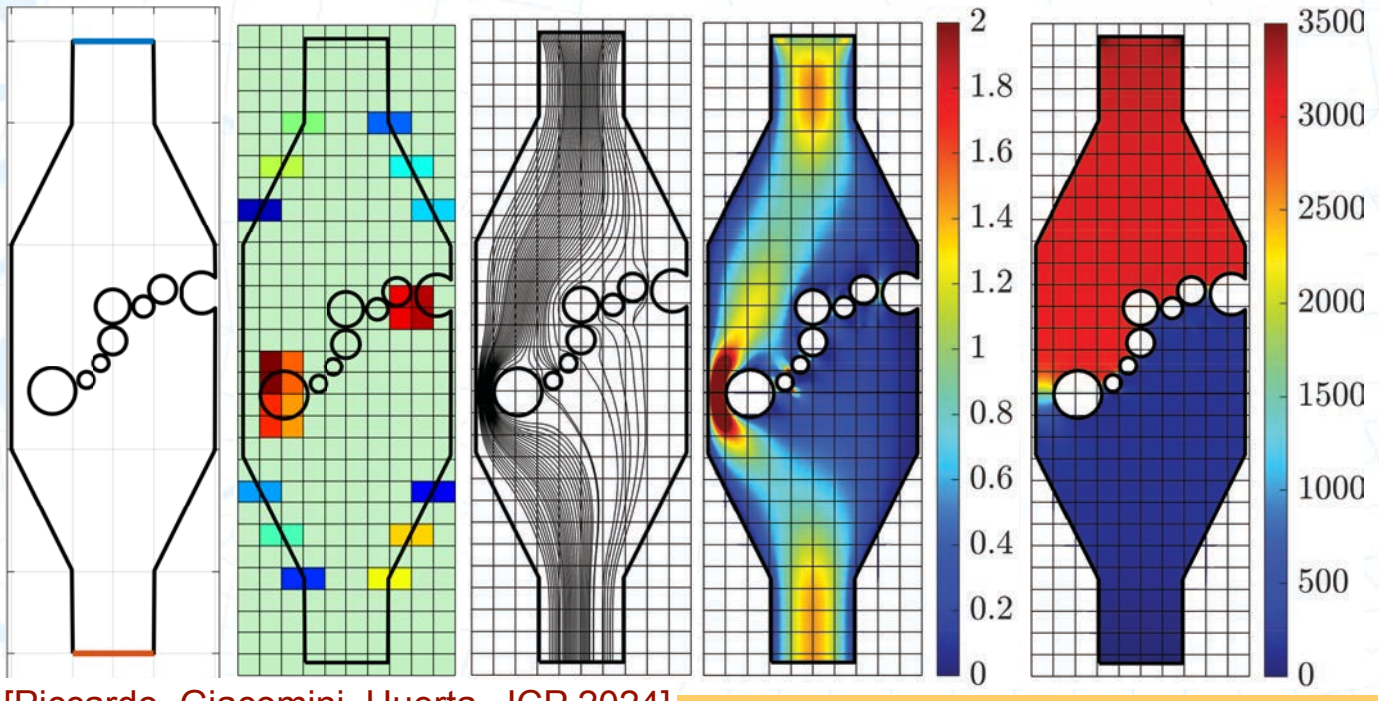
Basic ideas:

- **Embed NURBS boundaries** in a non-matching Cartesian grid. (no need for high-order curved mesh generation)
- Construct a high-order HDG approximation **without adding unknowns along the contour** (same unknowns as standard HDG)
- **Interpolate** and **integrate** along NURBS (exact geometry with NURBS-Enhanced FEM)
- Develop an automatic **degree-adaptive procedure** (using the HDG superconvergence).



[Piccardo, Giacomini, Huerta, JCP 2024]

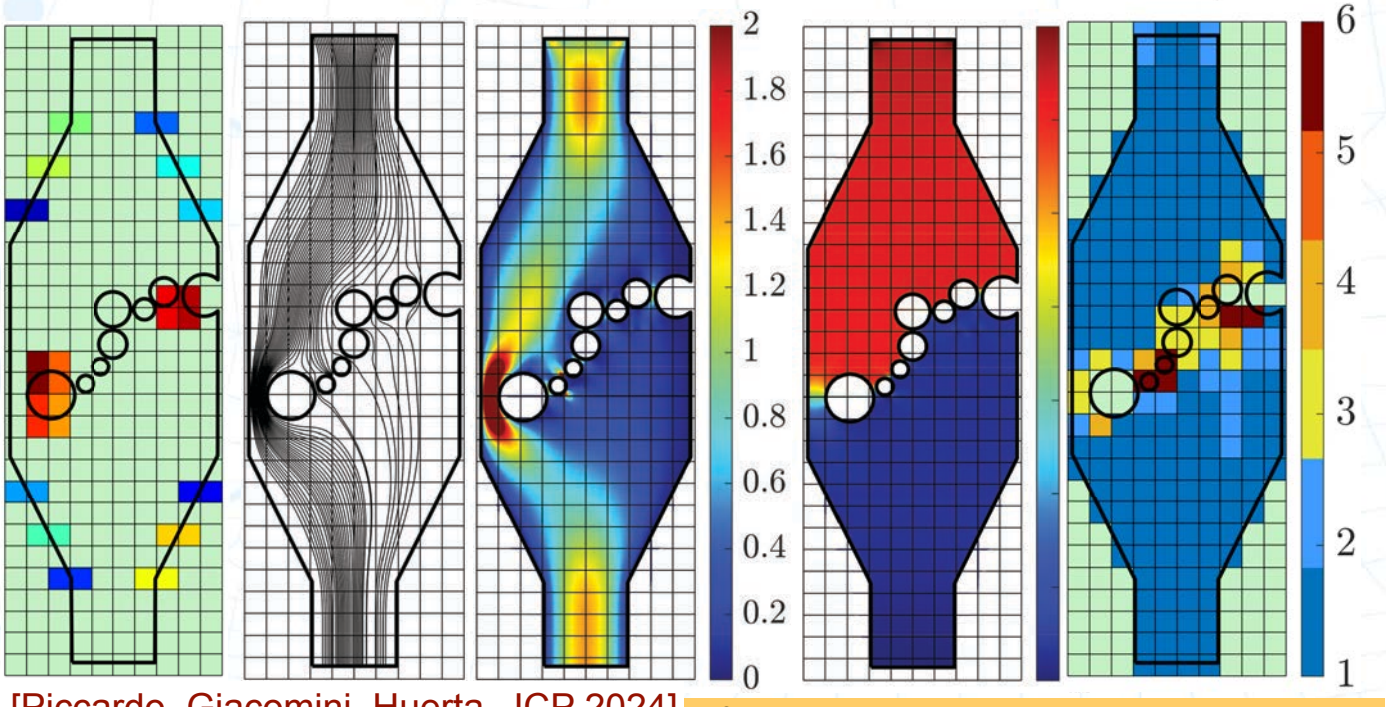
Microchannel with obstacles



[Piccardo, Giacomini, Huerta, JCP 2024]

[Piccardo, Giacomini, Huerta, JCP 2024]

Microchannel with obstacles



[Piccardo, Giacomini, Huerta, JCP 2024]

Characteristic boundary conditions for HDG

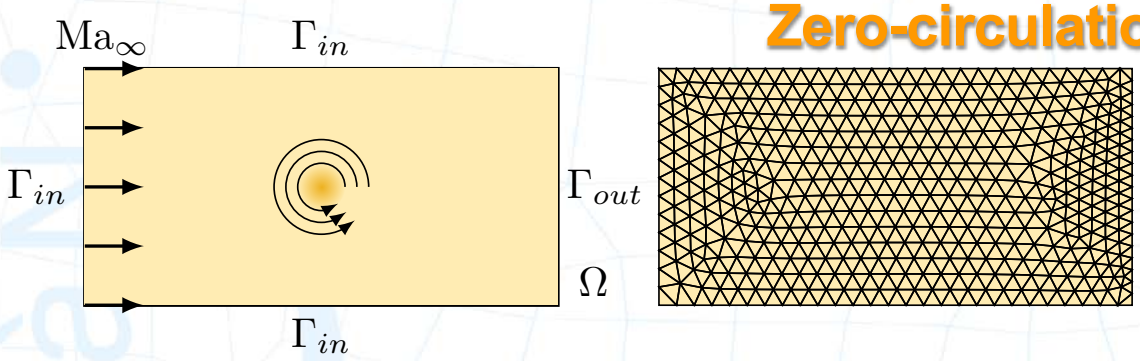
[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026]

Standard **boundary condition** operators

	Boundary type	Boundary condition operator
Γ_∞	Far-field, subsonic inflow, supersonic inflow/outflow	$\widehat{\boldsymbol{\Gamma}}_h = \boldsymbol{A}_n^+(\widehat{\boldsymbol{U}})(\boldsymbol{U}_e - \widehat{\boldsymbol{U}}) + \boldsymbol{A}_n^-(\widehat{\boldsymbol{U}})(\boldsymbol{U}_\infty - \widehat{\boldsymbol{U}})$, **may present spurious wave reflections**
Γ_{out}	Subsonic outflow (pressure outflow)	$\widehat{\boldsymbol{\Gamma}}_h = \left\{\rho_e - \widehat{\rho},\ [\rho \boldsymbol{v}_e - \widehat{\rho \boldsymbol{v}}]^T,\ p_{\text{out}}/(\gamma-1) + \rho_e \|\boldsymbol{v}_e\|^2/2 - \widehat{\rho E}\right\}^T$,
Γ_{ad}	Adiabatic wall	$\widehat{\boldsymbol{\Gamma}}_h = \left\{\rho_e - \widehat{\rho},\ \widehat{\rho \boldsymbol{v}}^T,\ (\mu/\text{RePr})\boldsymbol{\phi}_e \boldsymbol{n} - \tau_{\rho E}^d \left(\rho E_e - \widehat{\rho E}\right)\right\}^T$,
Γ_{iso}	Isothermal wall	$\widehat{\boldsymbol{\Gamma}}_h = \left\{\rho_e - \widehat{\rho},\ \widehat{\rho \boldsymbol{v}}^T,\ \rho_e T_w/\gamma - \widehat{\rho E}\right\}^T$,
Γ_{inv}	Inviscid wall or symmetry surface	$\widehat{\boldsymbol{\Gamma}}_h = \left\{\rho_e - \widehat{\rho},\ [(\mathbf{I}_{n_{sd}} - \boldsymbol{n} \otimes \boldsymbol{n})\rho \boldsymbol{v}_e - \widehat{\rho \boldsymbol{v}}]^T,\ \rho E_e - \widehat{\rho E}\right\}^T$.
$\widehat{\Gamma}_C^{\mathcal{T}}$	CBC	$\widehat{\boldsymbol{A}}^+(\widehat{\boldsymbol{U}}_h - \boldsymbol{U}_h) - \widehat{\boldsymbol{A}}^- \left(\widehat{\boldsymbol{U}}_h - \widehat{\boldsymbol{U}}_h^n + \Delta t \left[\widehat{\boldsymbol{P}}\widehat{\boldsymbol{D}}\widehat{\boldsymbol{P}}^{-1}\left(\widehat{\boldsymbol{U}}_h - \boldsymbol{U}^-\right) + \beta \widehat{\boldsymbol{B}} \frac{\partial \widehat{\boldsymbol{U}}}{\partial n}\right]\right)$

outgoing — incoming

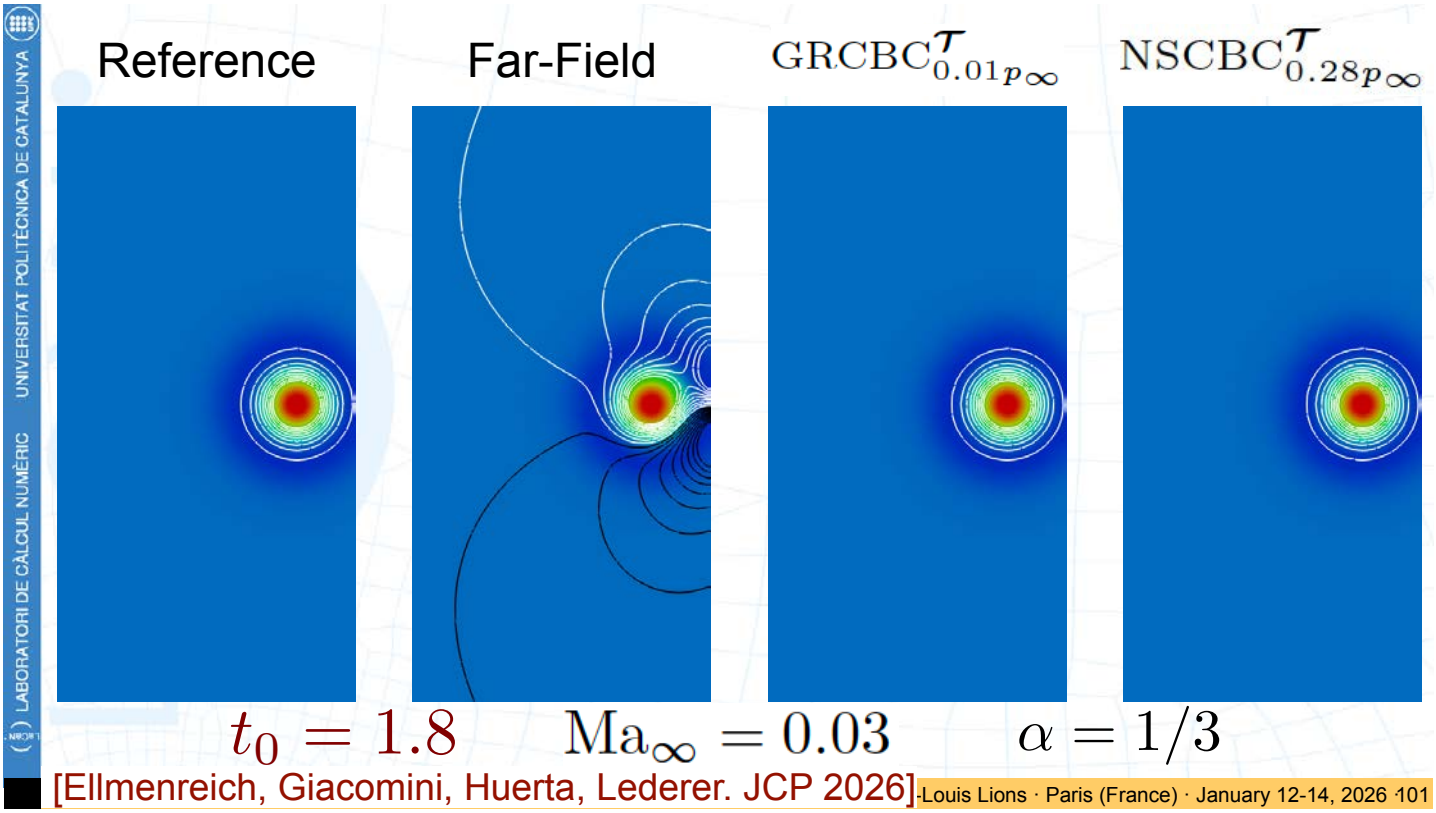
Zero-circulation vortex



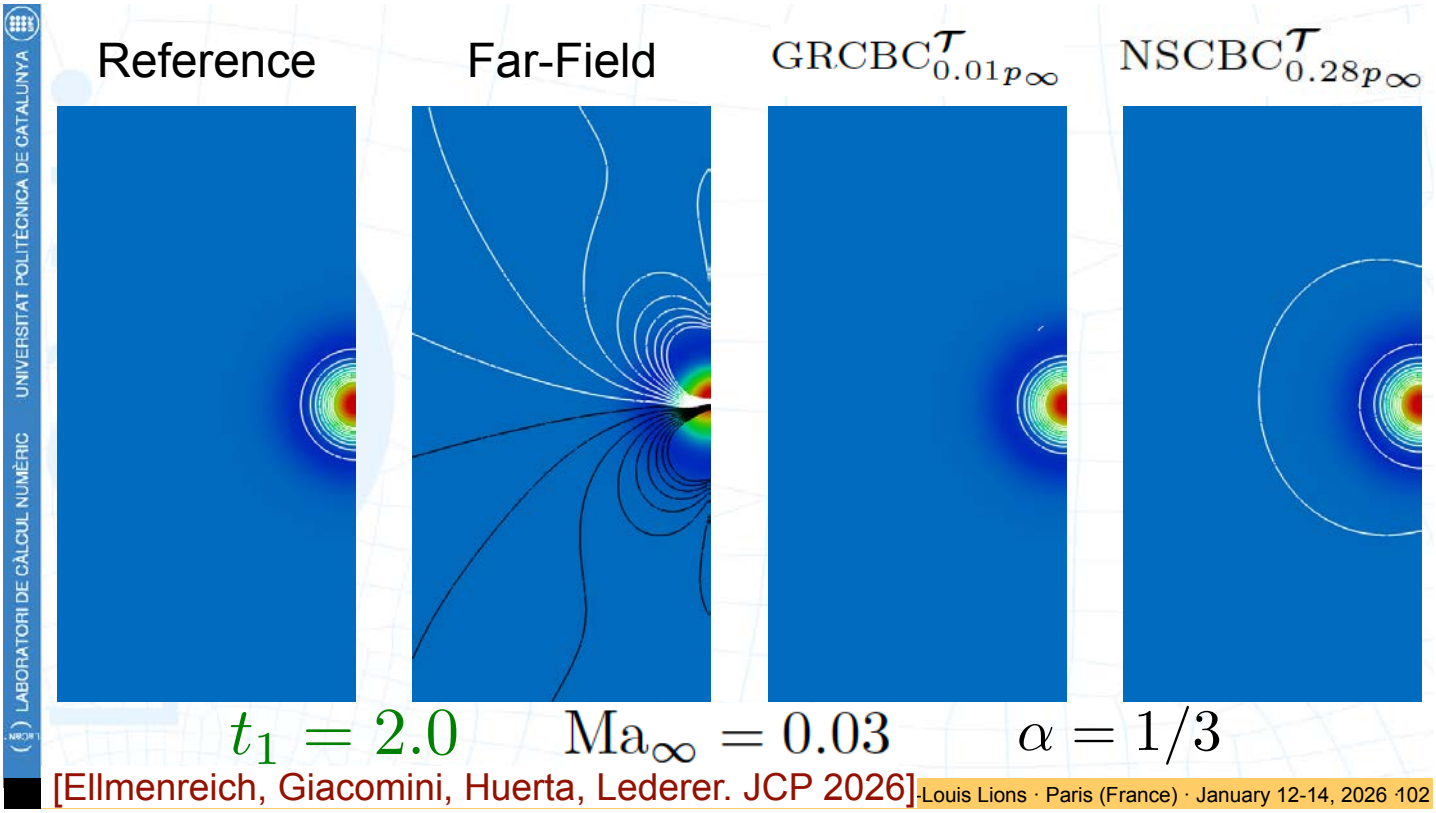
$$\begin{cases} \rho_0 = \rho_\infty \left(1 - \dfrac{\gamma-1}{2}\text{Ma}_\infty^2 \alpha^2 \exp\left(1 - r^2/R^2\right)\right)^{1/(\gamma-1)} \\ \boldsymbol{u}_0 = |\boldsymbol{u}_\infty| \left[\begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}-y\\x\end{pmatrix} \dfrac{\alpha}{R} \exp\left(\dfrac{1}{2}\left(1 - r^2/R^2\right)\right)\right], \\ p_0 = p_\infty \left(1 - \dfrac{\gamma-1}{2}\text{Ma}_\infty^2 \alpha^2 \exp\left(1 - r^2/R^2\right)\right)^{\gamma/(\gamma-1)} \end{cases} \qquad R = 0.1$$

α is the vortex strength ratio of the vortex to the base flow

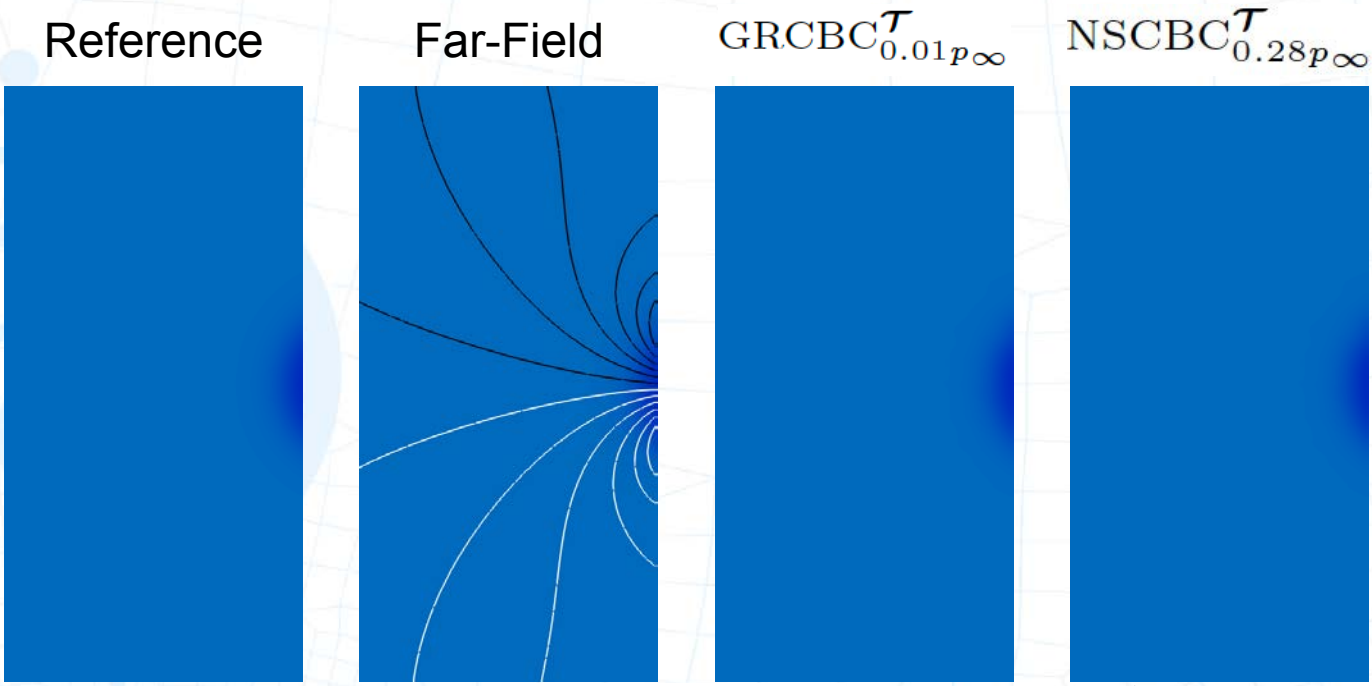
[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026]



[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026]



[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026]



[Ellmenreich, Giacomini, Huerta, Lederer. JCP 2026]

Summary II

- HDG outperforms DG, greatly reducing the number of DOFs.
- HDG has a slightly larger number of DOFs than CG.
- HDG provides stable approximations using the same order of polynomial discretisations for all the variables (easier mesh generation).
- HDG features a sparsity pattern particularly advantageous for the development of efficient linear solvers.
- HDG has specific advantages in the context of mechanical and flow problems (mesh robustness, locking-free, …).
- HDG is suitable for high and low order and degree adaptivity.
- FCFV can be interpreted as the lowest-order version of HDG.