Comparison of Continuous and Discontinuous Galerkin methods in incompressible fluid flow problems

Mahendra Paipuri[1-2], Sonia Fernández-Méndez[2] and Carlos Tiago[1]

[1]Instituto Superior Técnico (IST)

[2]LaCàN, Universitat Politècnica de Catalunya (UPC-BarcelonaTech)

Overview of CG and DG

Continuous Galerkin (CG)

- ✓ Well established
- ✓ Computationally efficient
- ✓ Static condensation

- × Unstable for high convective flows
- × Cumbersome p-adaptivity

DG (DG, IPM, CDG, LDG..)

- × duplication of nodes
- × no static condensation

- ✓ Stable for convective flows
- ✓ Suitability for adaptivity, parallel computing,…

Hybridizable Discontinous Galerkin (HDG)

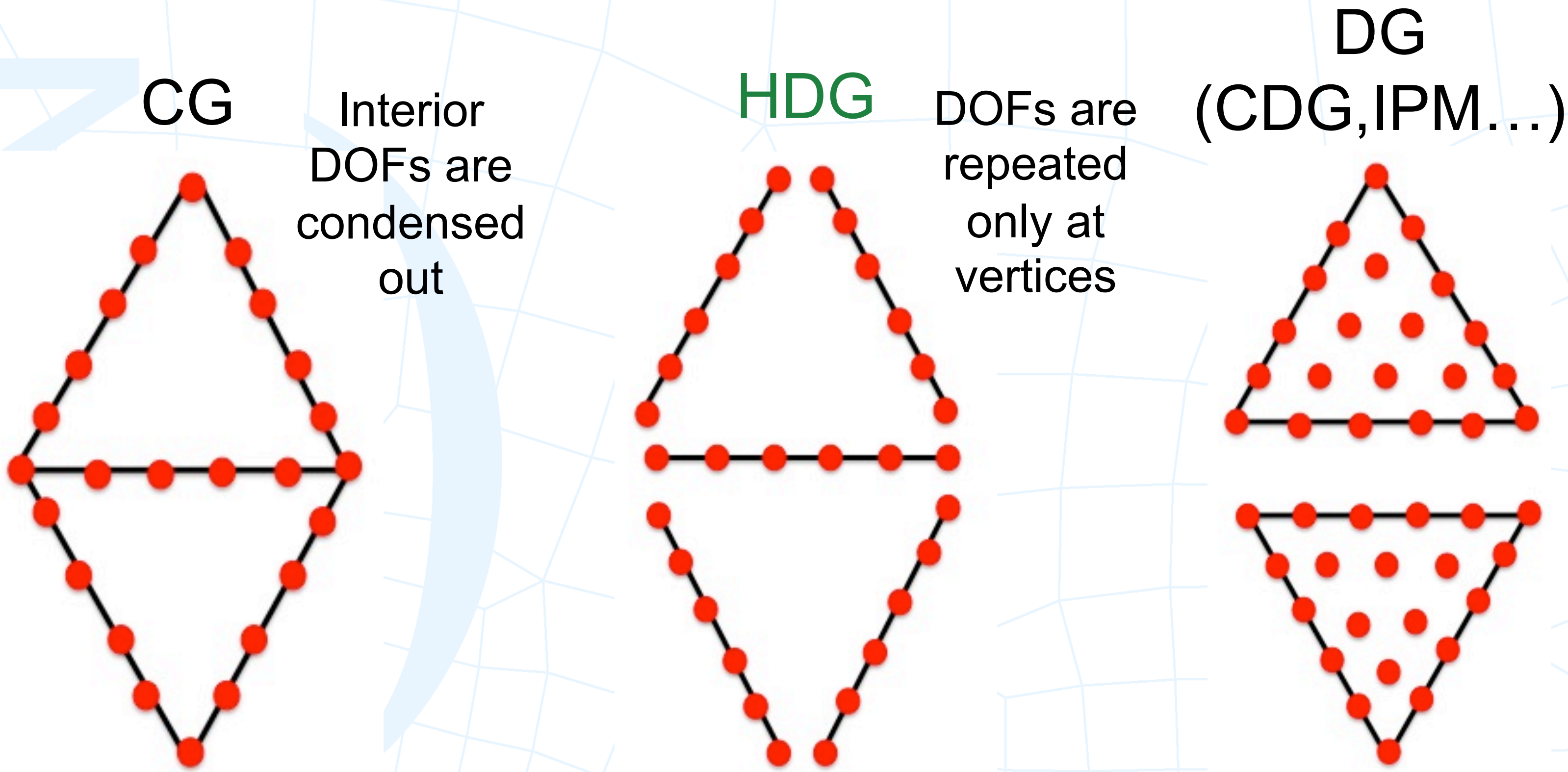
Computationally efficient?

UNIVERSITAT POLITÈCNICA DE CATALUNYA

LABORATORI DE CÀLCUL NUMÈRIC

Sailent features of HDG

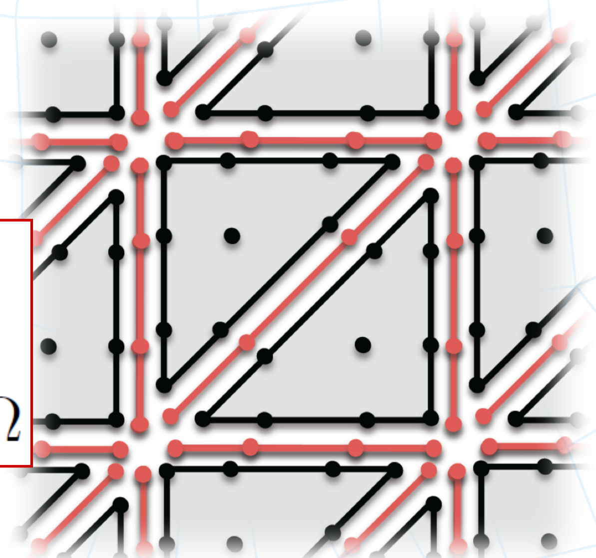
- Less DOFs compared to other DG methods



- Convergence of order k+1 for all variables: velocity, 1st derivatives, and pressure
- Superconvergence (order k+2 for velocity) with local postprocessing

1. Introduction to HDG

Incompressible Navier-Stokes equations



$$\begin{aligned} \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot (-p\mathbf{I} + \nu \nabla \boldsymbol{u}) &= \boldsymbol{f} \quad \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega \end{aligned}$$

Consider an element K_i in FE mesh

Equivalent problem inside each element

Local Problem

$$\begin{aligned} \mathbf{L} - \nabla \boldsymbol{u} &= \mathbf{0} \quad \text{in } K_i \\ \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \nabla \cdot (-p\mathbf{I} + \nu \mathbf{L}) &= \boldsymbol{f} \quad \text{in } K_i \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } K_i \\ \boldsymbol{u} &= \hat{\boldsymbol{u}} \quad \text{on } \partial K_i \\ \frac{1}{|K_i|} \int_{K_i} p \, dV &= \rho^{K_i} \quad \text{in } K_i \end{aligned}$$

$$\Gamma := \bigcup_{i=1}^{\mathtt{n_{el}}} \partial K_i$$

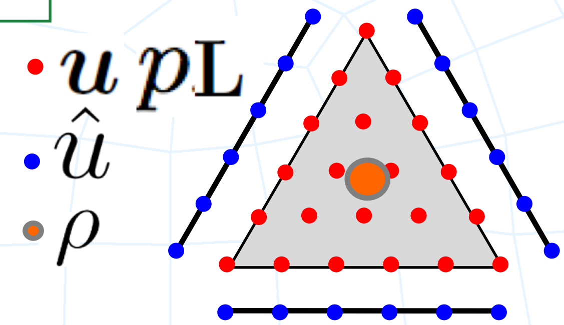
Global Problem

$$\begin{aligned} [\![(-p\mathbf{I} + \nu \mathbf{L}) \cdot \boldsymbol{n}]\!] &= \mathbf{0} \quad \text{on } \Gamma \backslash \partial\Omega \\ \int_{\partial K_i} \hat{\boldsymbol{u}} \cdot \boldsymbol{n} \, dS &= 0 \quad \text{on } \Gamma \backslash \partial\Omega \\ \boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega \end{aligned}$$

HDG Local Problem

$$
\begin{aligned}
\mathbf{L} - \boldsymbol{\nabla} \boldsymbol{u} &= \mathbf{0} && \text{in } K_i \\
\boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot (-p\mathbf{I} + \nu \mathbf{L}) &= \boldsymbol{f} && \text{in } K_i \\
\boldsymbol{\nabla} \cdot \boldsymbol{u} &= 0 && \text{in } K_i \\
\boldsymbol{u} &= \hat{\boldsymbol{u}} && \text{on } \partial K_i \\
\frac{1}{|K_i|} \int_{K_i} p \, dV &= \rho^{K_i} && \text{in } K_i
\end{aligned}
$$

HDG discretization leads to a non-linear system **in each element**. The Newton-Raphson linealization is: given **δû**, δρ find δ**u**, δ**L** and δ**p** s.t.

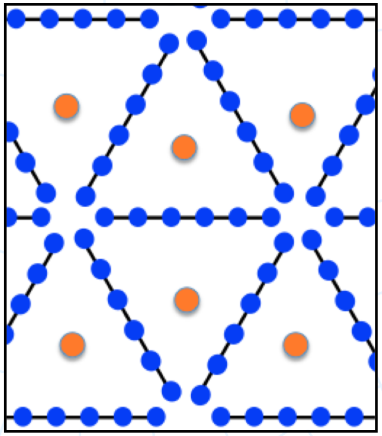


$$
\begin{bmatrix} \mathbf{A}_{uu}(\boldsymbol{u}) & \mathbf{A}_{uL} & \mathbf{A}_{up} \\ \mathbf{A}_{Lu} & \mathbf{A}_{LL} & \\ \mathbf{A}_{pu} & & \end{bmatrix}
\begin{bmatrix} \delta \mathbf{u} \\ \delta \mathbf{L} \\ \delta \mathbf{p} \end{bmatrix}
= -
\begin{bmatrix} \mathbf{r}_u \\ \mathbf{r}_L \\ \mathbf{r}_p \end{bmatrix}
-
\begin{bmatrix} \mathbf{A}_{u\hat{u}}(\hat{\boldsymbol{u}}) \\ \mathbf{A}_{L\hat{u}} \\ \mathbf{A}_{p\hat{u}} \end{bmatrix}
\begin{bmatrix} \delta \hat{\mathbf{u}}_{F1} \\ \delta \hat{\mathbf{u}}_{F2} \\ \delta \hat{\mathbf{u}}_{F3} \end{bmatrix}
$$

with the constraint $\mathbf{A}_{\rho p} \delta \mathbf{p} = \rho^k$ **Local Solver**

HDG Global problem

$$
\begin{aligned}
[\![(-p\mathbf{I} + \nu\mathbf{L}) \cdot \boldsymbol{n}]\!] &= \mathbf{0} \quad \text{on } \Gamma \backslash \partial\Omega \\
\int_{\partial K_i} \hat{\boldsymbol{u}} \cdot \boldsymbol{n} \, dS &= 0 \quad \text{on } \Gamma \backslash \partial\Omega \\
\boldsymbol{u} &= \boldsymbol{u}_D \quad \text{on } \partial\Omega
\end{aligned}
$$



- $\hat{u}$: velocity trace
- ρ : mean pressure

HDG discretization
NR linealization
replace local problem

$$
\begin{bmatrix} \mathbf{A}_{\hat{u}\hat{u}}(\hat{\mathbf{u}}) & \mathbf{A}_{\rho\hat{u}}^T \\ \mathbf{A}_{\rho\hat{u}} & \end{bmatrix}
\begin{bmatrix} \delta\hat{\mathbf{u}} \\ \delta\rho \end{bmatrix}
=
\begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix}
$$

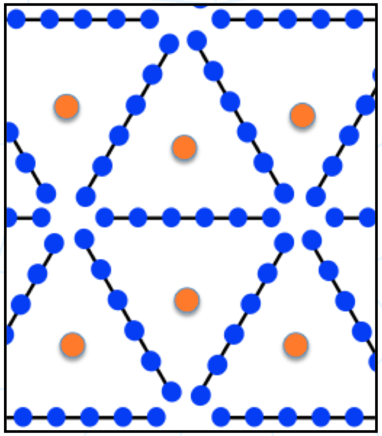
Global Solver

UNIVERSITAT POLITÈCNICA DE CATALUNYA
LABORATORI DE CÀLCUL NUMÈRIC

HDG computations

1. <u>Global problem</u>

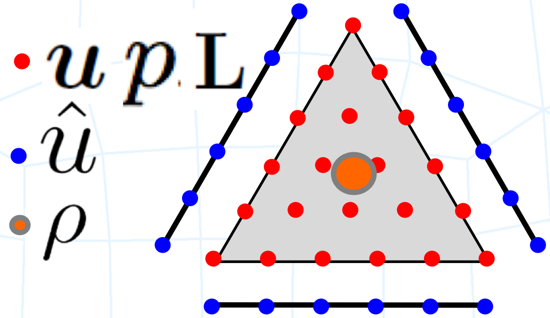
$$\begin{bmatrix} \mathbf{A}_{\hat{u}\hat{u}}(\hat{\boldsymbol{u}}) & \mathbf{A}_{\rho\hat{u}}^T \\ \mathbf{A}_{\rho\hat{u}} & \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{u}} \\ \delta\rho \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix}$$



***matrix** with nice block structure*

2. Cheap <u>element-by-element</u> postprocess (local solver)

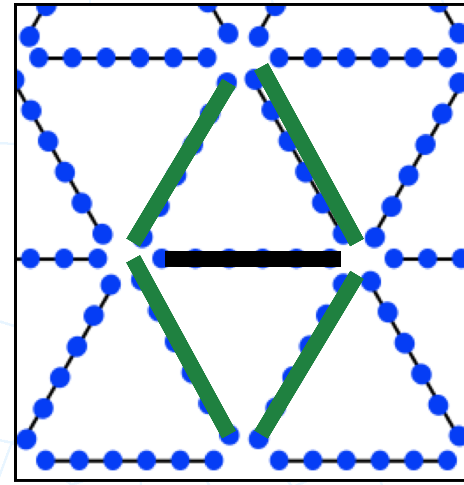
$$\delta\hat{\boldsymbol{u}}\ \delta\rho \rightarrow \delta\boldsymbol{u}^K \delta\boldsymbol{L}^K \delta\boldsymbol{p}^K$$



3. Cheap element-by-element 2nd postprocess
→ superconvergent solution (order k+2 for u)

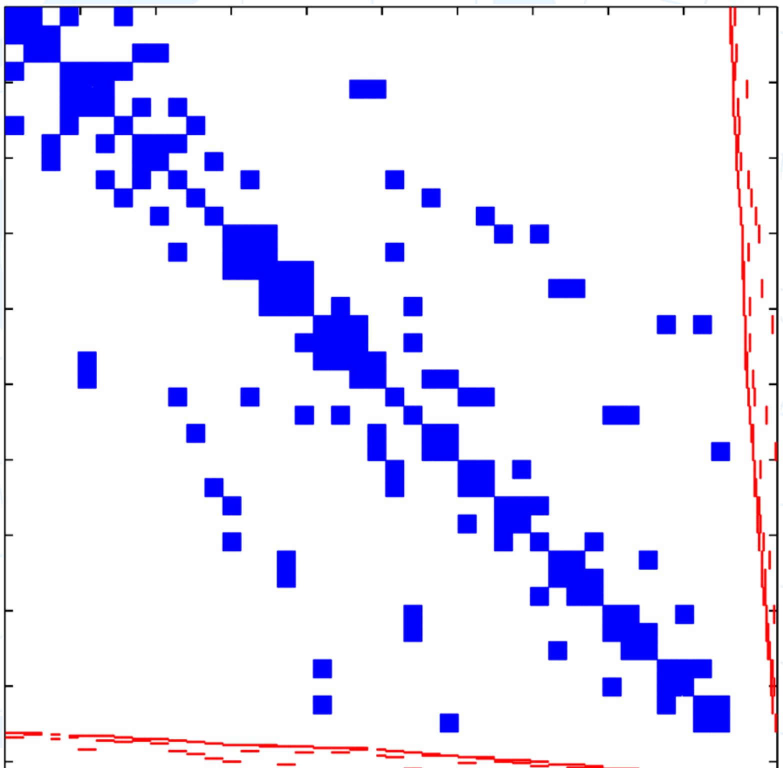
HDG block structure

- Each face has contributions from 4 faces of neighbouring elements.
- Each row has 5 blocks of same size

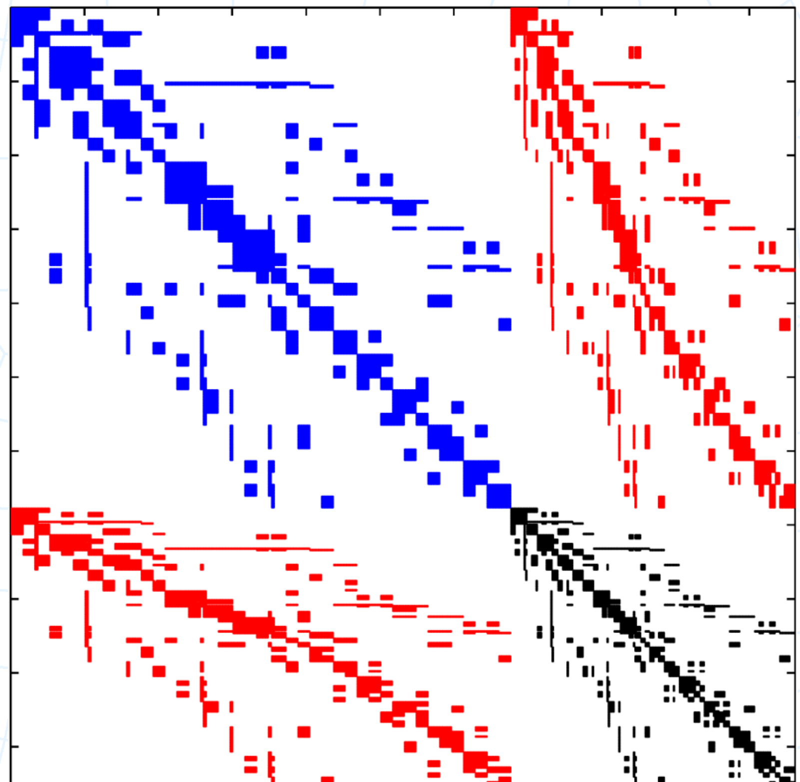


HDG is more efficient for linear solvers

HDG #ndofs = 511



CG #ndofs = 530



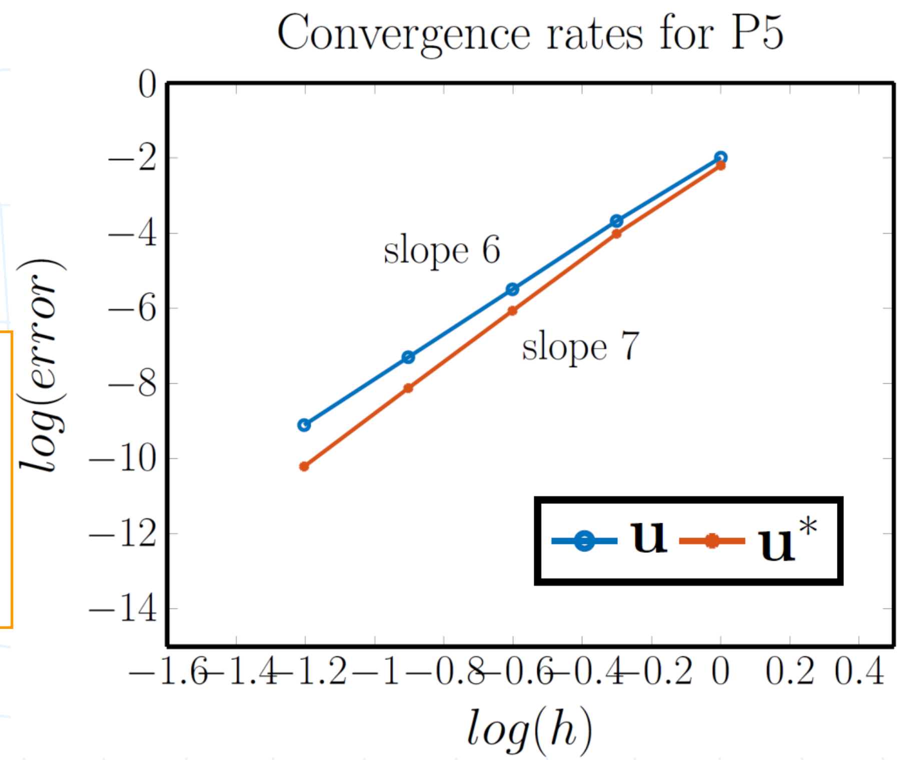
Degree k= 5

HDG super convergence

- Optimal convergence: order k+1 for *u*, its gradient, and p
- Element-by-element postprocess in $P^{k+1}(K)$ solving the following equations

$$\boxed{\begin{aligned} \nabla\cdot\nabla\boldsymbol{u}^* &= \nabla\cdot\boldsymbol{L} \quad \text{in } K_i \\ \boldsymbol{n}\cdot\nabla\boldsymbol{u}^* &= \boldsymbol{n}\cdot\boldsymbol{L} \quad \text{on } \partial K_i \\ \int_{K_i}\boldsymbol{u}^* dV &= \int_{K_i}\boldsymbol{u}\, dV \end{aligned}}$$

leads to a superconvergent solution *u**, with convergence of order k+2

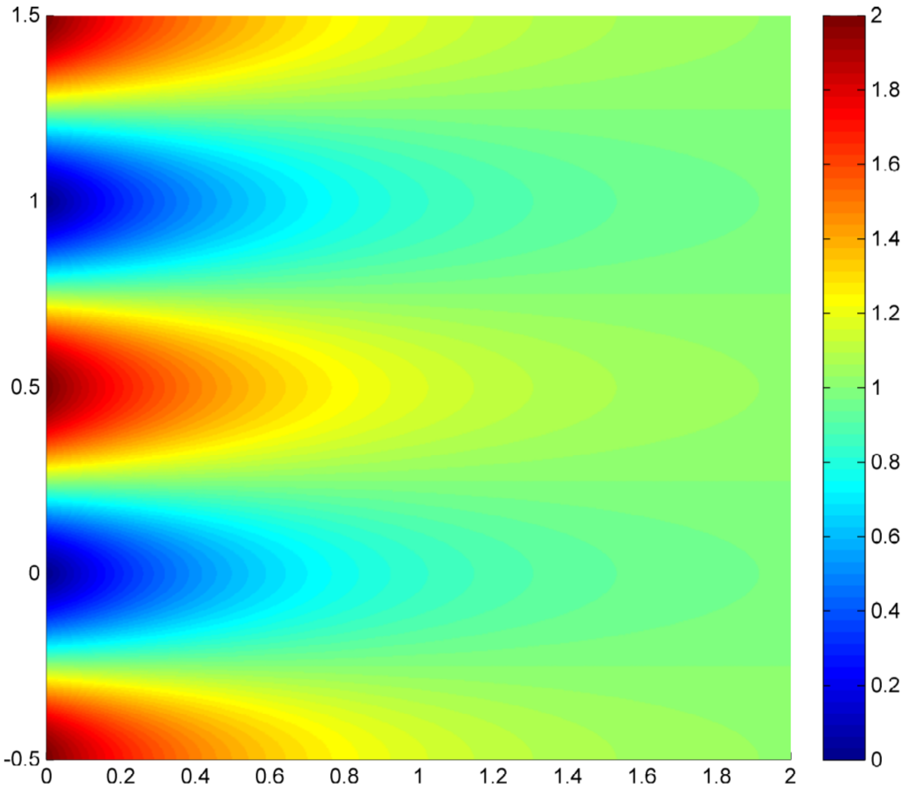


Inexpensive and reliable
Error Estimator

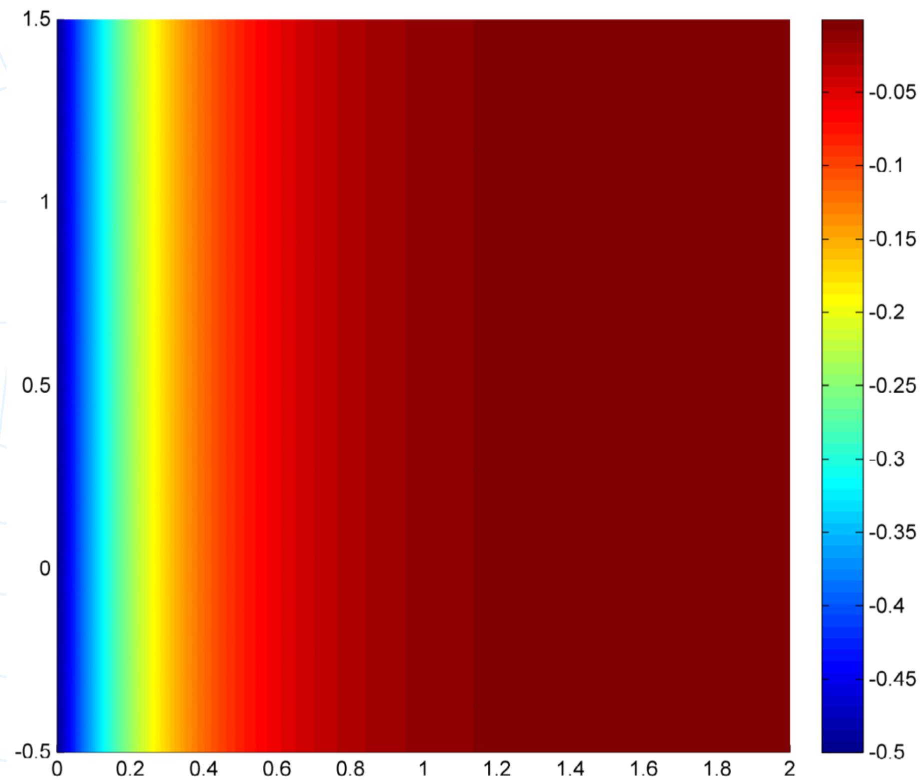
$$\boxed{u - u^*}$$

EFFICIENCY COMPARISON CG vs HDG
Numerical examples

Analytical solution: Kovasznay flow at Re = 20 in [0 2] x [-0,5 1,5]



Velocity

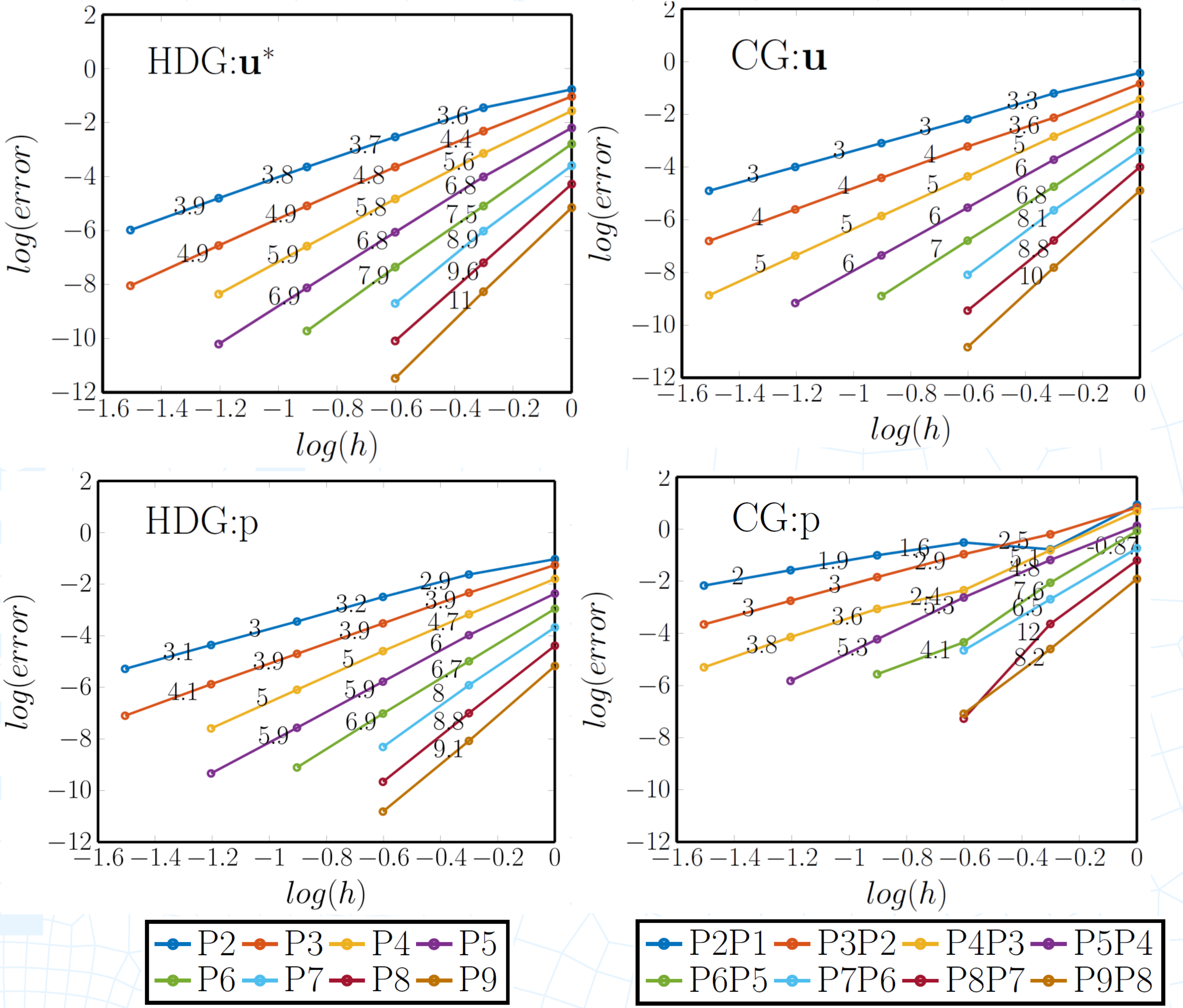


Pressure

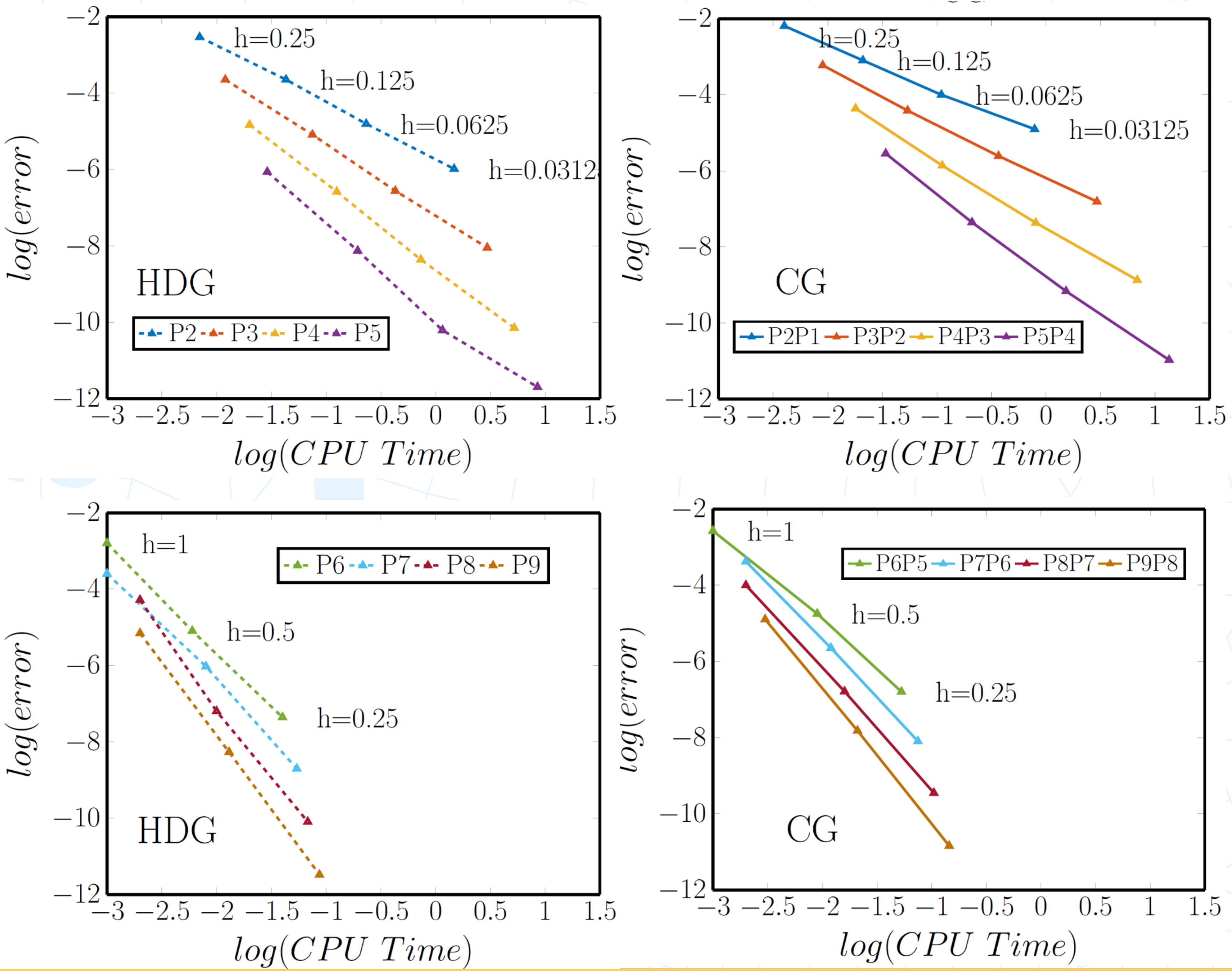
UNIVERSITAT POLITÈCNICA DE CATALUNYA

LABORATORI DE CÀLCUL NUMÈRIC

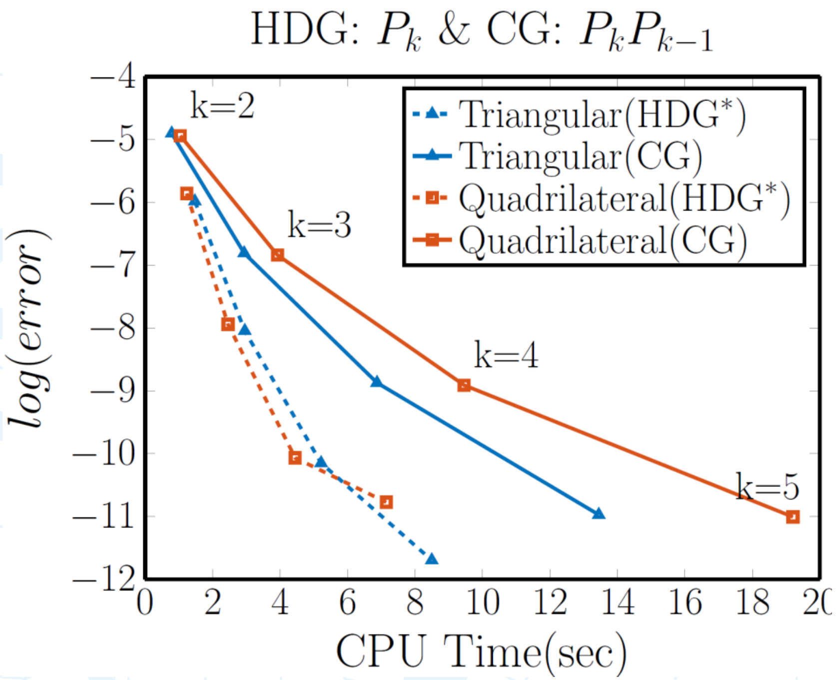
Stokes: Convergence Plots



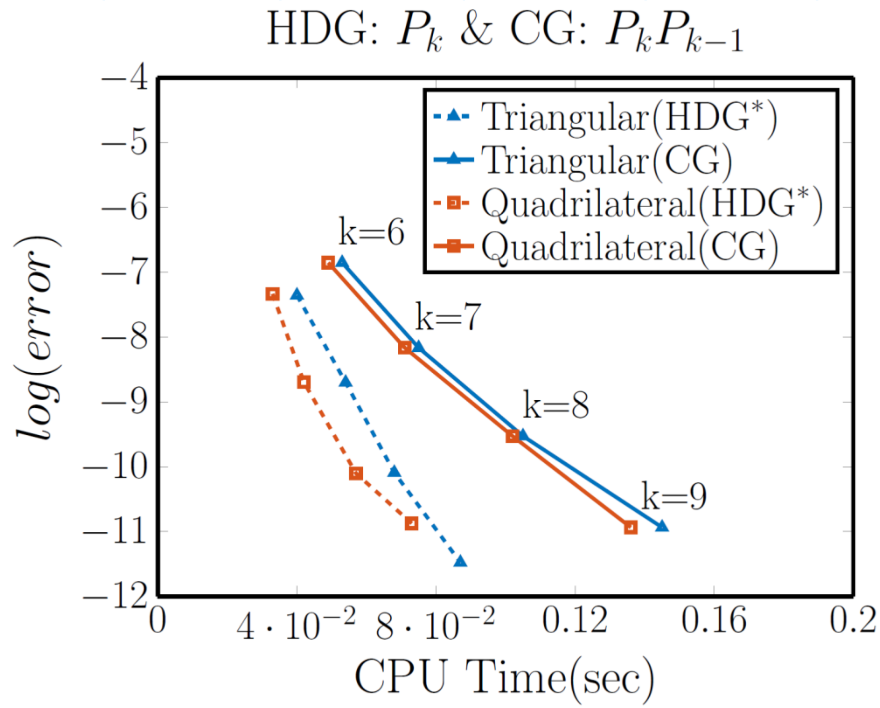
Stokes: CPU Time for linear solver



Stokes: CPU Time for linear solver



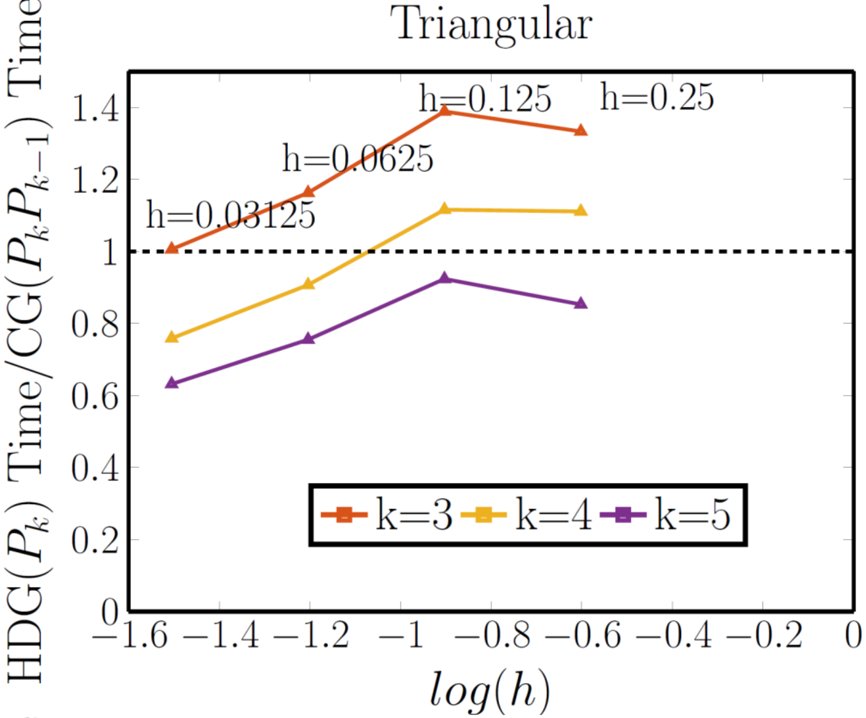
h = 0,03125

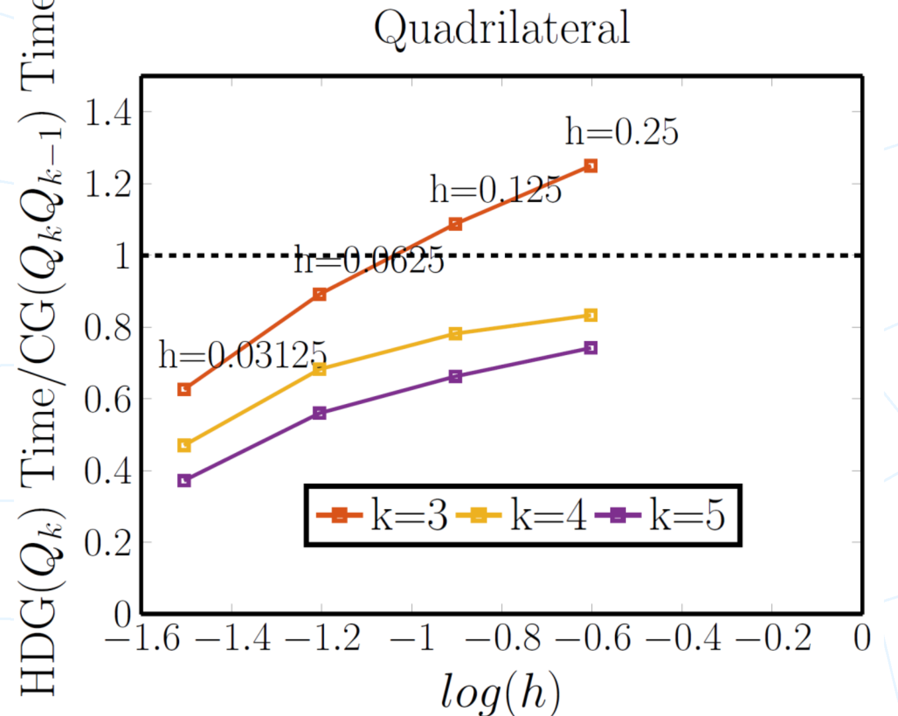


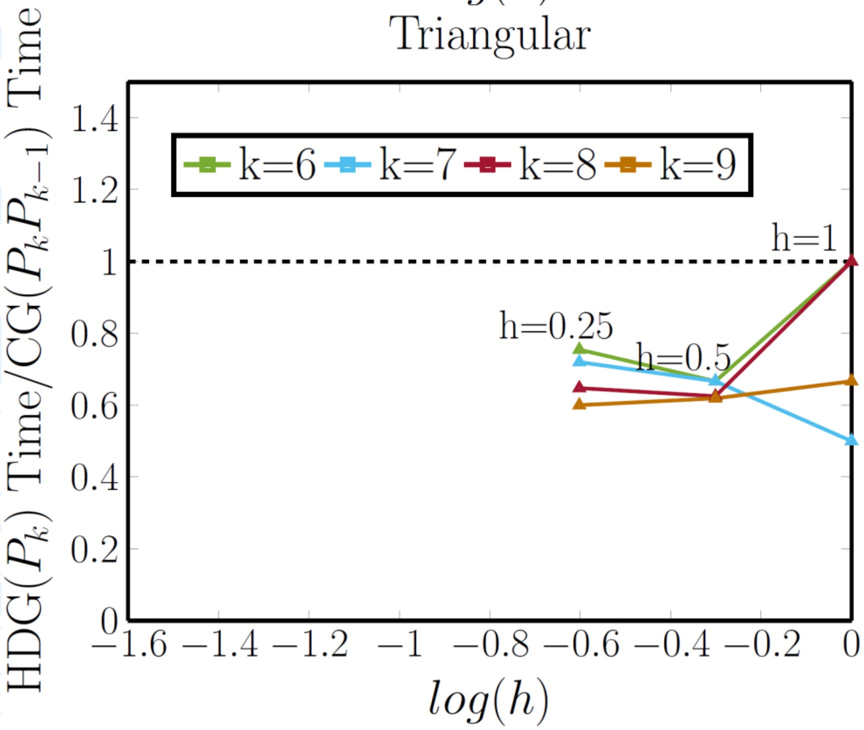
h = 0,25

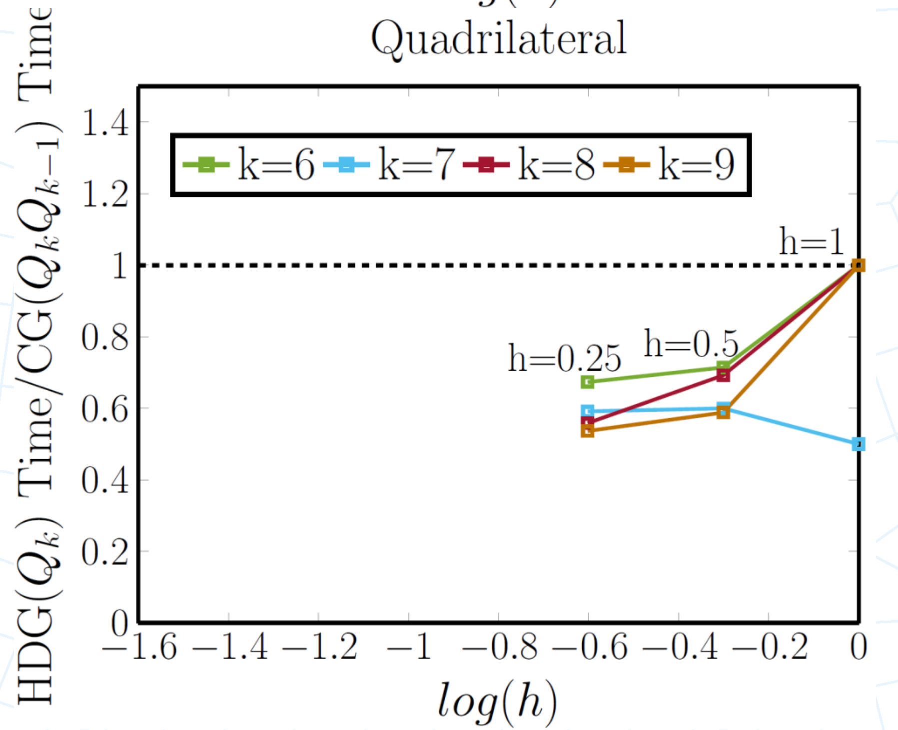
Similar trend is observed in the case of Navier-Stokes.

Stokes: CPU Time for linear solver

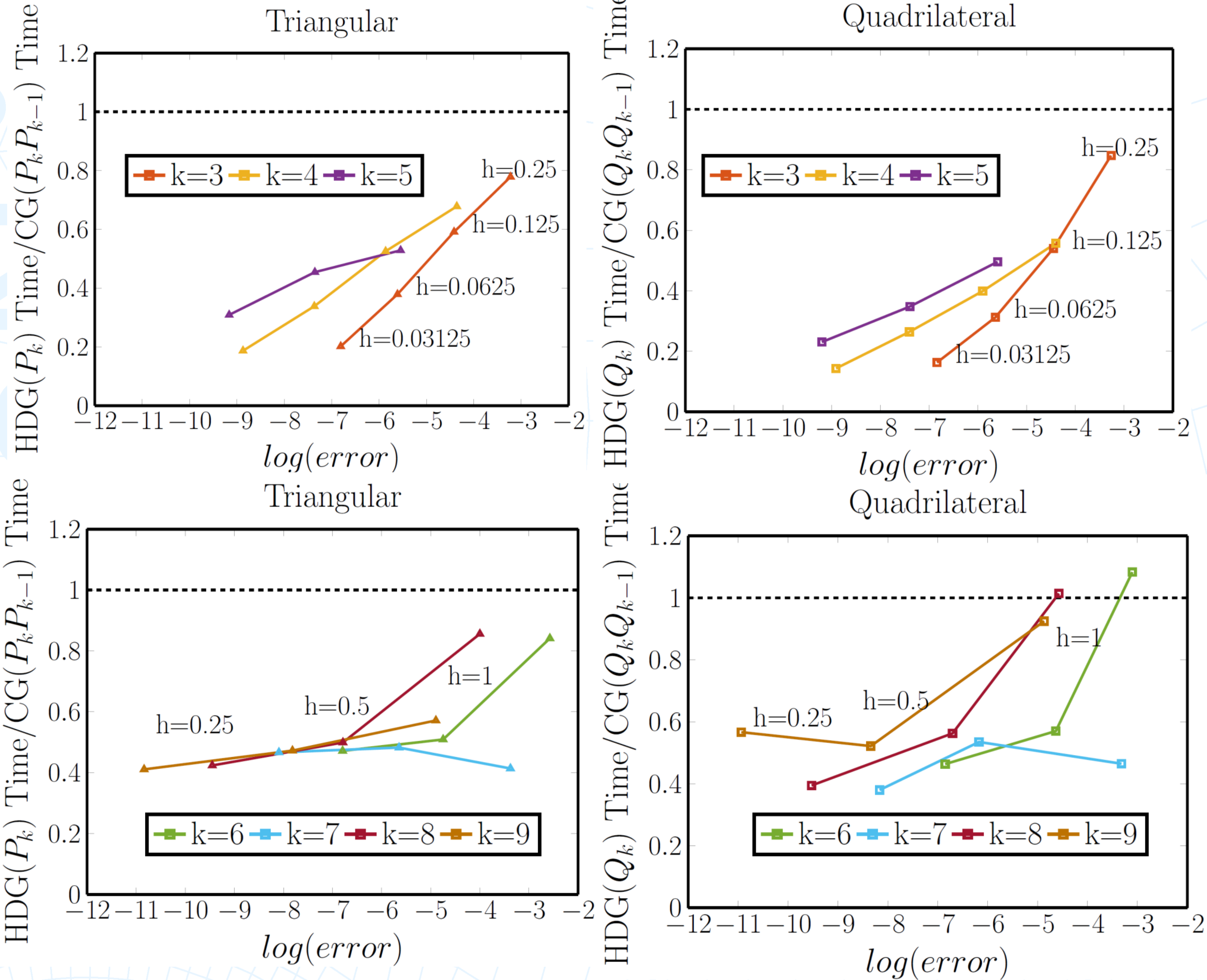






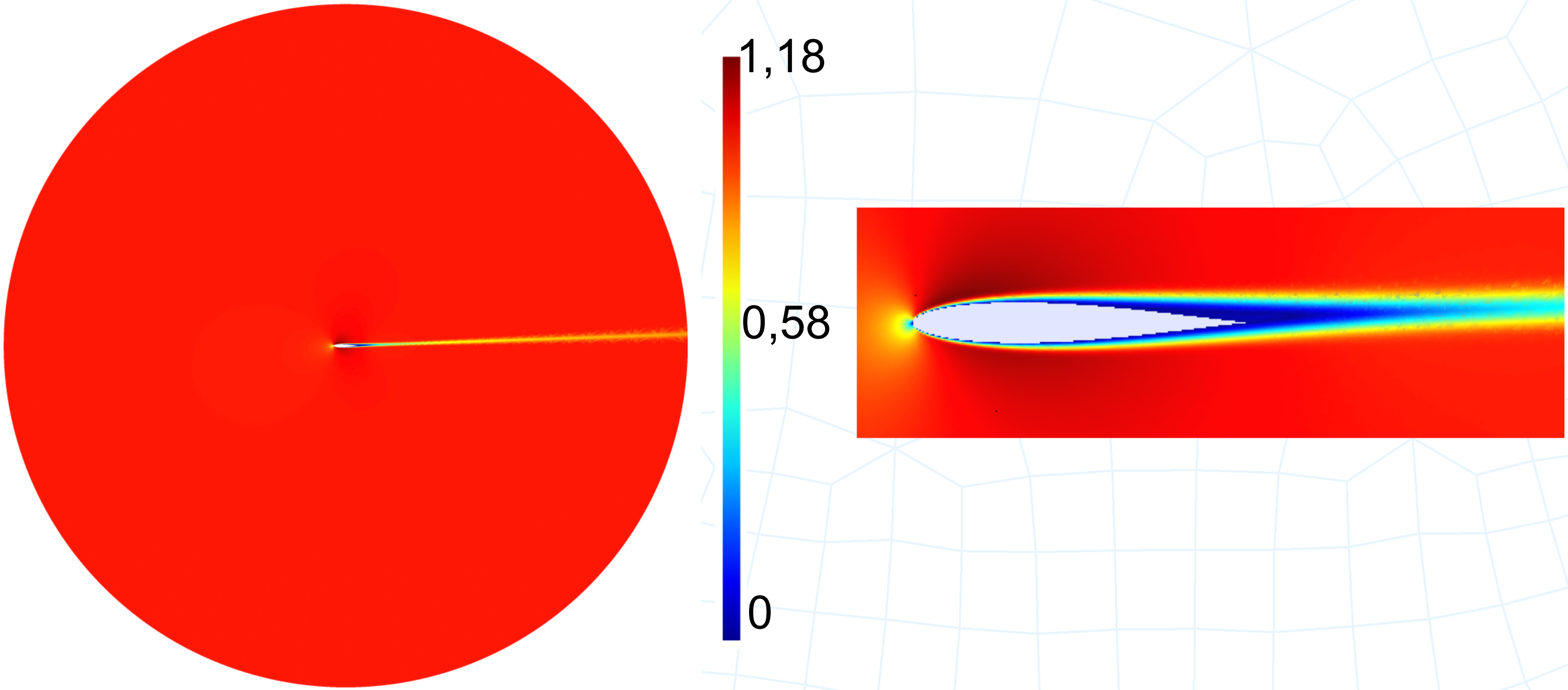


Stokes: CPU Time for linear solver



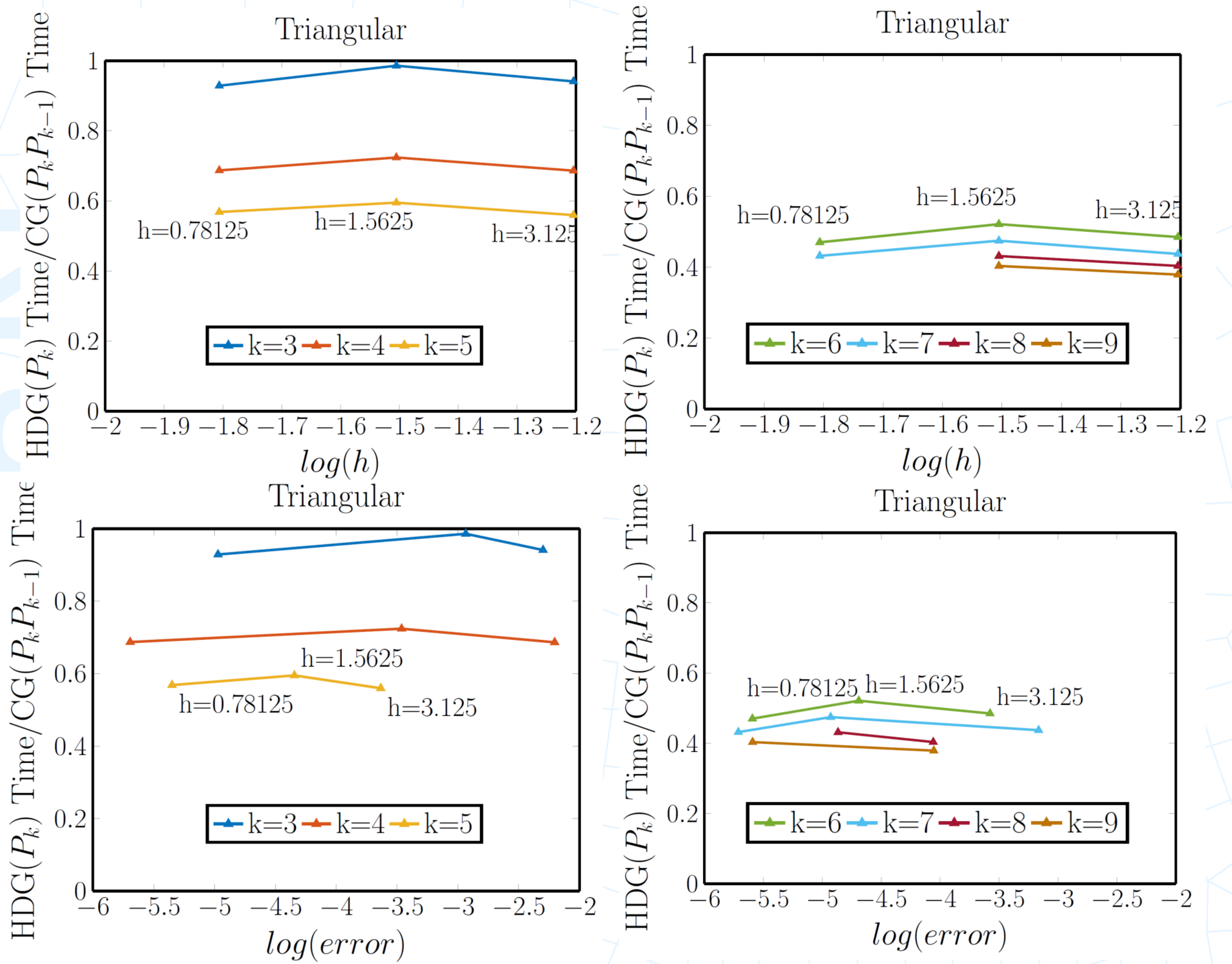
NACA0012 Airfoil

Incompressible Navier-Stokes is solved at Re = 5000 and Angle of attack (AoA) = 2º ; Error in lift coefficient is compared



Velocity

NACA: CPU Time for linear solver



Concluding remarks

- Computational efficiency between HDG and CG for Stokes problem
  - High order-coarser mesh is more efficient than low order-finer mesh in both HDG and CG.
  - HDG is more efficient at order $k \geq 3$ for same level of accuracy, in terms of CPU time of linear solver
- Similar computational efficiency is observed in Navier-Stokes example (NACA).