Universitat Politecnica de Catalunya

Master of Science in Computational Mechanics

Finite Elements in Fluids

Final Assignment #10
Hybridizable Discontinuous Galerkin

Author:
Carlos Eduardo Ribeiro Santa Cruz Mendoza

June 5, 2019

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

1 Problem formulation

Considering a domain defined by $\Omega = [0,1]^2$ such that $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ with $\Gamma_D \cap \Gamma_N = \emptyset$, $\Gamma_D \cap \Gamma_R = \emptyset$ and $\Gamma_N \cap \Gamma_R = \emptyset$. Specifically:

$$\begin{aligned}\Gamma_N &:= \{(x,y) \in \mathbb{R}^2 \ : \ x = 0\} \\ \Gamma_R &:= \{(x,y) \in \mathbb{R}^2 \ : \ x = 1\} \\ \Gamma_D &:= \partial\Omega \backslash (\Gamma_N \cup \Gamma_R)\end{aligned}$$

which is subject to the following partial differential equation:

$$\begin{cases} -\boldsymbol{\nabla} \cdot (\kappa \boldsymbol{\nabla} u) = s & \text{in } \Omega, \\ u = u_D & \text{on } \Gamma_D \\ \boldsymbol{n} \cdot (\kappa \boldsymbol{\nabla} u) = t & \text{on } \Gamma_N \\ \boldsymbol{n} \cdot (\kappa \boldsymbol{\nabla} u) + \gamma u = g & \text{on } \Gamma_R \end{cases} \tag{1.1}$$

A new problem can be formulated considering a divided domain with n_{el} elements. The boundary conditions and the governing equation remain, but a "link" between elements is introduced (note the jump operator $[\![\cdot]\!]$) as it can be seen on the strong form at Equation 1.2.

$$\begin{cases} -\boldsymbol{\nabla} \cdot (\kappa \boldsymbol{\nabla} u) = s & \text{in } \Omega_i, \text{ for } i = 1, ..., n_{el}, \\ u = u_D & \text{on } \Gamma_D \\ \boldsymbol{n} \cdot (\kappa \boldsymbol{\nabla} u) = t & \text{on } \Gamma_N \\ \boldsymbol{n} \cdot (\kappa \boldsymbol{\nabla} u) + \gamma u = g & \text{on } \Gamma_R \\ [\![u \boldsymbol{n}]\!] = \mathbf{0} & \text{on } \Gamma \\ [\![\boldsymbol{n} \cdot \boldsymbol{\nabla} u]\!] = 0 & \text{on } \Gamma \end{cases} \tag{1.2}$$

This "broken" domain can be treated as local and global problems. The local problem, shown in Equation 1.3, treats the governing equations and the Dirichlet boundary condition.

$$\begin{cases} \boldsymbol{\nabla} \cdot \mathbf{q}_i = s & \text{in } \Omega_i, \\ \mathbf{q}_i + \kappa \boldsymbol{\nabla} u_i = \mathbf{0} & \text{in } \Omega_i, \\ u_i = u_D & \text{on } \partial\Omega_i \cap \Gamma_D, \\ u_i = \hat{u} & \text{on } \partial\Omega_i \backslash \Gamma_D, \end{cases} \tag{1.3}$$

The global problem, on the other hand, is solved on the element edges using a new variable $\hat{u}$, imposing as well the Neumann and Robin boundary conditions at Γ_N and Γ_R. The global problem is stated in Equation 1.4.

$$\begin{cases} [\![\boldsymbol{n}\cdot\boldsymbol{q}]\!] = 0 & \text{on } \Gamma \\ \boldsymbol{n}\cdot\boldsymbol{q} = -t & \text{on } \Gamma_N \\ \boldsymbol{n}\cdot\boldsymbol{q} = \gamma\hat{u} - g & \text{on } \Gamma_R \end{cases} \tag{1.4}$$

For the local problem, the weak formulation defines the following expressions in which the unknowns are $(\mathbf{q}_i, u_i) \in \mathcal{W}(\Omega_i) \times \mathcal{V}(\Omega_i)$.

$$\begin{aligned} -(\boldsymbol{\nabla} v, \boldsymbol{q}_i)_{\Omega_i} + \langle v, \boldsymbol{n}_i \cdot \hat{\mathbf{q}}_i \rangle_{\partial\Omega_i} &= (v, s)_{\Omega_i} \\ -(\boldsymbol{w}, \boldsymbol{q}_i)_{\Omega_i} + \kappa(\boldsymbol{\nabla}\cdot\boldsymbol{w}, u_i)_{\Omega_i} &= \kappa\langle \boldsymbol{n}_i\cdot\boldsymbol{w}, u_D\rangle_{\partial\Omega_i\cap\Gamma_D} + \kappa\langle \boldsymbol{n}_i\cdot\boldsymbol{w}, \hat{u}\rangle_{\partial\Omega_i\setminus\Gamma_D} \end{aligned} \tag{1.5}$$

As for the global problem, the unknown is the introduced variable $\hat{u}$ such that:

$$\sum_{n=1}^{n_{el}} \langle \mu, \boldsymbol{n}_i\cdot\hat{\mathbf{q}}_i\rangle_{\partial\Omega_i\setminus\partial\Omega} + \sum_{n=1}^{n_{el}} \langle \mu, \boldsymbol{n}_i\cdot\hat{\mathbf{q}}_i + t\rangle_{\partial\Omega_i\cap\Gamma_N} + \sum_{n=1}^{n_{el}} \langle \mu, \boldsymbol{n}_i\cdot\hat{\mathbf{q}}_i + g - \gamma\hat{u}\rangle_{\partial\Omega_i\cap\Gamma_R} = 0 \tag{1.6}$$

or, by substituting the definition of the fluxes with stabilization, the following global formulation is obtained:

$$\begin{aligned} \sum_{n=1}^{n_{el}} \Big\{ \langle \mu, \boldsymbol{n}_i\cdot\mathbf{q}_i\rangle_{\partial\Omega_i\setminus\Gamma_D} + \langle \mu, \tau_i u_i\rangle_{\partial\Omega_i\setminus\Gamma_D} - \langle \mu, \tau_i\hat{u}\rangle_{\partial\Omega_i\setminus\Gamma_D} - \langle \mu, \gamma\hat{u}\rangle_{\partial\Omega_i\cap\Gamma_R} \Big\} \\ = -\sum_{n=1}^{n_{el}} \Big\{ \langle \mu, g\rangle_{\partial\Omega_i\cap\Gamma_R} + \langle \mu, t\rangle_{\partial\Omega_i\cap\Gamma_N} \Big\} \end{aligned} \tag{1.7}$$

We can introduce, then, the interpolation element-by-element of the variables:

$$\boldsymbol{q} \approx \boldsymbol{q}^h = \sum_{n=1}^{n_{el}} N_j \boldsymbol{q}_j \tag{1.8}$$

$$u \approx u^h = \sum_{n=1}^{n_{el}} N_j u_j \tag{1.9}$$

$$\hat{u} \approx \hat{u}^h = \sum_{n=1}^{n_{el}} \hat{N}_j \hat{u}_j \tag{1.10}$$

Finally, using this interpolation and matrix notation, the local system of equations can be defined:

$$\begin{bmatrix} \mathbf{A_{uu}} & \mathbf{A_{uq}} \\ \kappa\mathbf{A_{uq}}^T & \mathbf{A_{qq}} \end{bmatrix}_i \begin{bmatrix} \mathbf{u_i} \\ \mathbf{q_i} \end{bmatrix} = \begin{bmatrix} \mathbf{f_u} \\ \kappa\mathbf{f_q} \end{bmatrix}_i + \begin{bmatrix} \mathbf{A_{u\hat{u}}} \\ \kappa\mathbf{A_{q\hat{u}}} \end{bmatrix}_i \hat{\mathbf{u}}_i \tag{1.11}$$

To define the global system, we must define the matrices related to the robin boundary condition:

$$\mathbf{A}^R_{\hat{\mathbf{u}}\hat{\mathbf{u}}} = -\sum_{\partial\Omega_i\cap\Gamma_R} \gamma \sum_{g=1}^{n^f_{ip}} \hat{\mathbf{N}}_{\mathbf{n}}(\xi^{\boldsymbol{f}}_{\boldsymbol{g}})\hat{\mathbf{N}}^T(\xi^{\boldsymbol{f}}_{\boldsymbol{g}})w^f_g$$

$$\mathbf{f}^R_{\hat{\mathbf{u}}} = -\sum_{\partial\Omega_i\cap\Gamma_R} \sum_{g=1}^{n^f_{ip}} \mathbf{N}(\xi^{\boldsymbol{f}}_{\boldsymbol{g}})g(\mathbf{x}(\xi^{\boldsymbol{f}}_{\boldsymbol{g}}))w^f_g$$

Then, the global system yields:

$$\sum_{n=1}^{n_{el}} \left\{ \begin{bmatrix} \mathbf{A}^T_{\mathbf{u}\hat{\mathbf{u}}} & \mathbf{A}^T_{\mathbf{q}\hat{\mathbf{u}}} \end{bmatrix}_i \begin{bmatrix} \mathbf{u_i} \\ \mathbf{q_i} \end{bmatrix} + [\mathbf{A}_{\hat{\mathbf{u}}\hat{\mathbf{u}}}]_i \, \hat{\mathbf{u}}_i + \left[\mathbf{A}^R_{\hat{\mathbf{u}}\hat{\mathbf{u}}}\right]_i \hat{\mathbf{u}}_i \right\} = \sum_{n=1}^{n_{el}} \left\{ [\mathbf{f}_{\hat{\mathbf{u}}}]_i + \left[\mathbf{f}^R_{\hat{\mathbf{u}}}\right]_i \right\} \quad (1.12)$$

Once the local solution is found element-by-element by Equation 1.11, the global system can be solved simply by

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

where

$$\hat{\mathbf{K}} = \mathop{\mathbf{A}}_{i=1}^{n_{el}} \begin{bmatrix} \mathbf{A}^T_{\mathbf{u}\hat{\mathbf{u}}} & \mathbf{A}^T_{\mathbf{q}\hat{\mathbf{u}}} \end{bmatrix}_i \begin{bmatrix} \mathbf{A}_{\mathbf{uu}} & \mathbf{A}_{\mathbf{uq}} \\ \kappa\mathbf{A}^T_{\mathbf{uq}} & \mathbf{A}_{\mathbf{qq}} \end{bmatrix}^{-1}_i \begin{bmatrix} \mathbf{A}_{\mathbf{u}\hat{\mathbf{u}}} \\ \kappa\mathbf{A}_{\mathbf{q}\hat{\mathbf{u}}} \end{bmatrix}_i + [\mathbf{A}_{\hat{\mathbf{u}}\hat{\mathbf{u}}}]_i + \left[\mathbf{A}^R_{\hat{\mathbf{u}}\hat{\mathbf{u}}}\right]_i$$

and

$$\hat{\mathbf{f}} = \mathop{\mathbf{A}}_{i=1}^{n_{el}} [\mathbf{f}_{\hat{\mathbf{u}}}]_i + \left[\mathbf{f}^R_{\hat{\mathbf{u}}}\right]_i - \begin{bmatrix} \mathbf{A}^T_{\mathbf{u}\hat{\mathbf{u}}} & \mathbf{A}^T_{\mathbf{q}\hat{\mathbf{u}}} \end{bmatrix}_i \begin{bmatrix} \mathbf{A}_{\mathbf{uu}} & \mathbf{A}_{\mathbf{uq}} \\ \kappa\mathbf{A}^T_{\mathbf{uq}} & \mathbf{A}_{\mathbf{qq}} \end{bmatrix}^{-1}_i \begin{bmatrix} \mathbf{f_u} \\ \kappa\mathbf{f_q} \end{bmatrix}_i$$

2 Implementation

The analytical expression (using $a = 6$, $b = 0.6$, $\kappa = 0.75$ and $\gamma = 0.7$) given on Equation 2.1 is used to find the boundary conditions.

$$u(x, y) = \sin(a\pi x + \kappa)\exp(-\gamma(x + by)^2) \tag{2.1}$$

yielding

$$u_D = \begin{cases} \sin(a\pi x + \kappa)\exp(-\gamma(x)^2) & \text{for } y = 0 \\ \sin(a\pi x + \kappa)\exp(-\gamma(x + b)^2) & \text{for } y = 1 \end{cases} \tag{2.2}$$

for the Dirichlet boundary condition,

$$t = 0,43\,\mathrm{e}^{-\frac{63\,y^2}{250}}\,y - 3,3\,\pi\,\mathrm{e}^{-\frac{63\,y^2}{250}} \tag{2.3}$$

for the Neumann boundary condition and

$$g = \kappa\,\pi\,\cos\left(\kappa + \pi\,a\right)\,\mathrm{e}^{-\gamma\,(b\,y+1)^2}\,a - \gamma(\kappa(2by + 2) + 1)\sin\left(\kappa + \pi\,a\right)\,\mathrm{e}^{-\gamma\,(b\,y+1)^2} \tag{2.4}$$

for the robin boundary condition.

The analytical solution provides, thus, the distribution found on Figure 2.1.

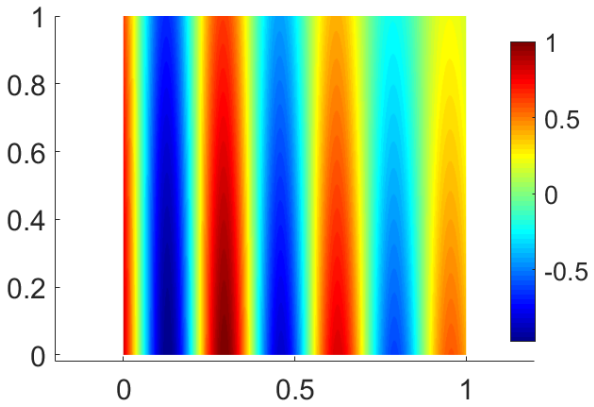
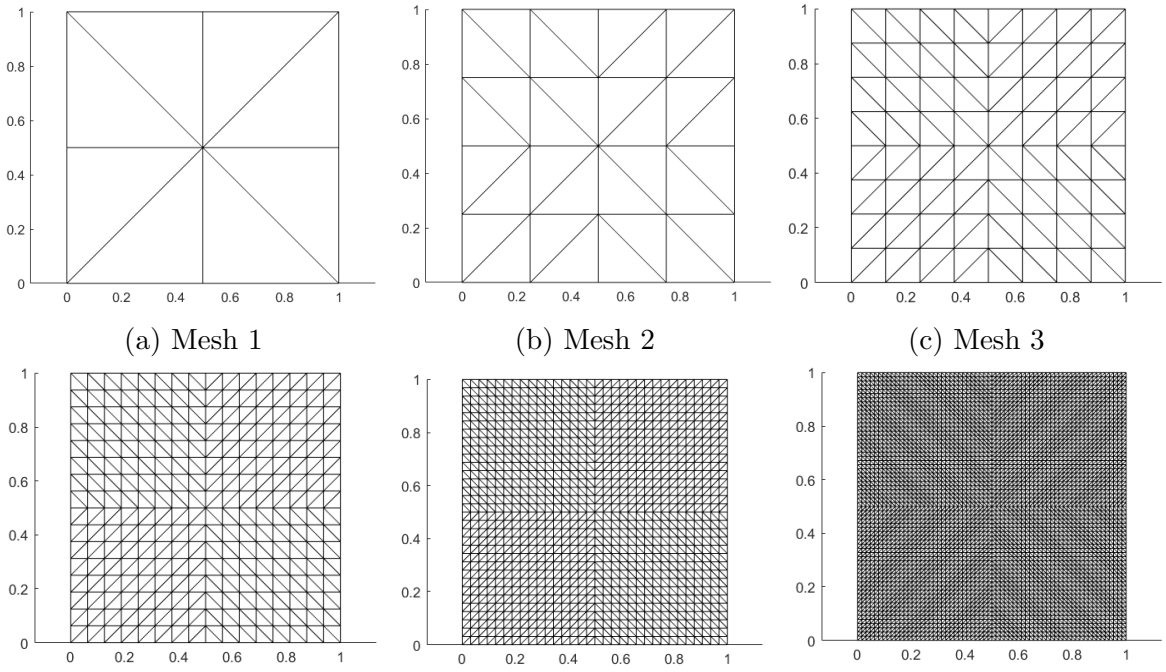


Figure 2.1: Analytical solution

Once the element faces contained in boundaries are identified, the corresponding boundary conditions are imposed (Dirichlet for $y = 0$ and $y = 1$, Neumann for $x = 0$ and Robin for $x = 1$). A routine was implemented to identify the boundaries and ultimately allow the computation of Equations 2.3 and 2.4.

The provided code also needed to be modified to include the Robin and Neumann degrees of freedom. Finally, the system of equations in matrix notation could, then, be solved as before.

The provided meshes are presented on Figure 2.2, ranging from 8 to 8192 elements. Also available are different polynomial degrees of approximation. Starting in linear until fourth order.



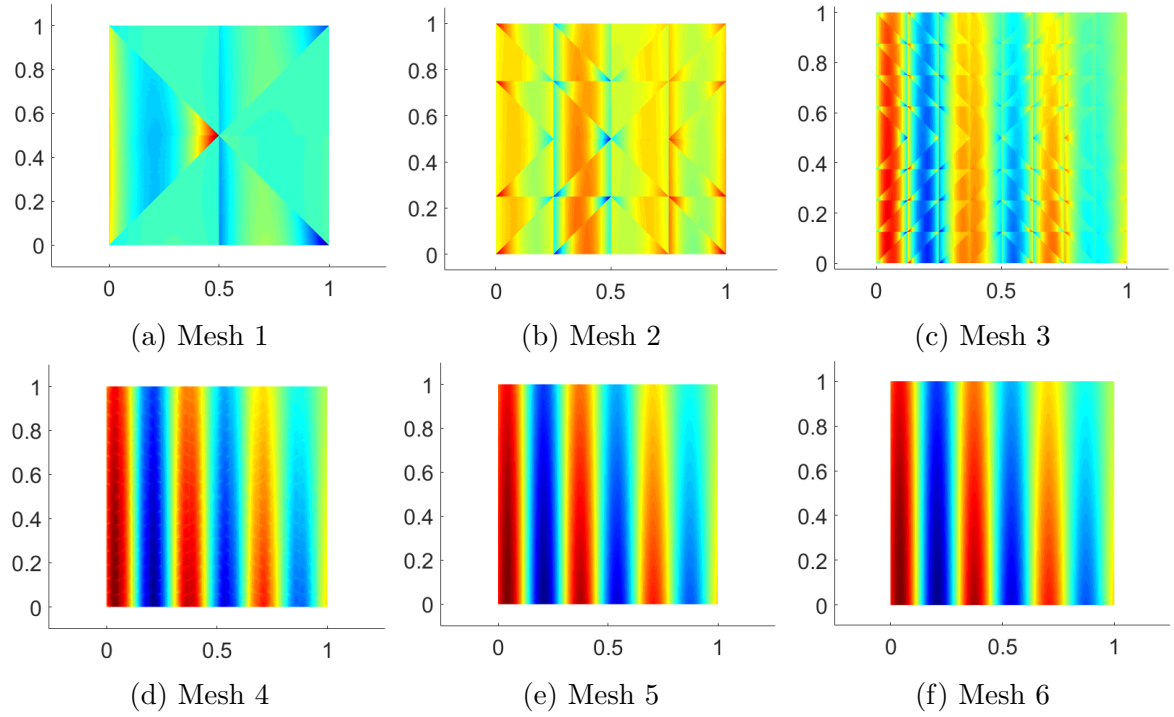
(a) Mesh 1 (b) Mesh 2 (c) Mesh 3

(d) Mesh 4 (e) Mesh 5 (f) Mesh 6

Figure 2.2: Available grids

3 Results

The results obtained using all grids and quadratic approximation are presented on Figure 3.1. As it can be observed, the simulation reached good agreement with the analytical solution (2.1) from the fourth mesh on. It's also evident the effect of the "jumps", namely the discontinuities of the solution. Specially on the three coarser meshes, the discontinuities of the solution between elements are clear, delimiting the elements edges.

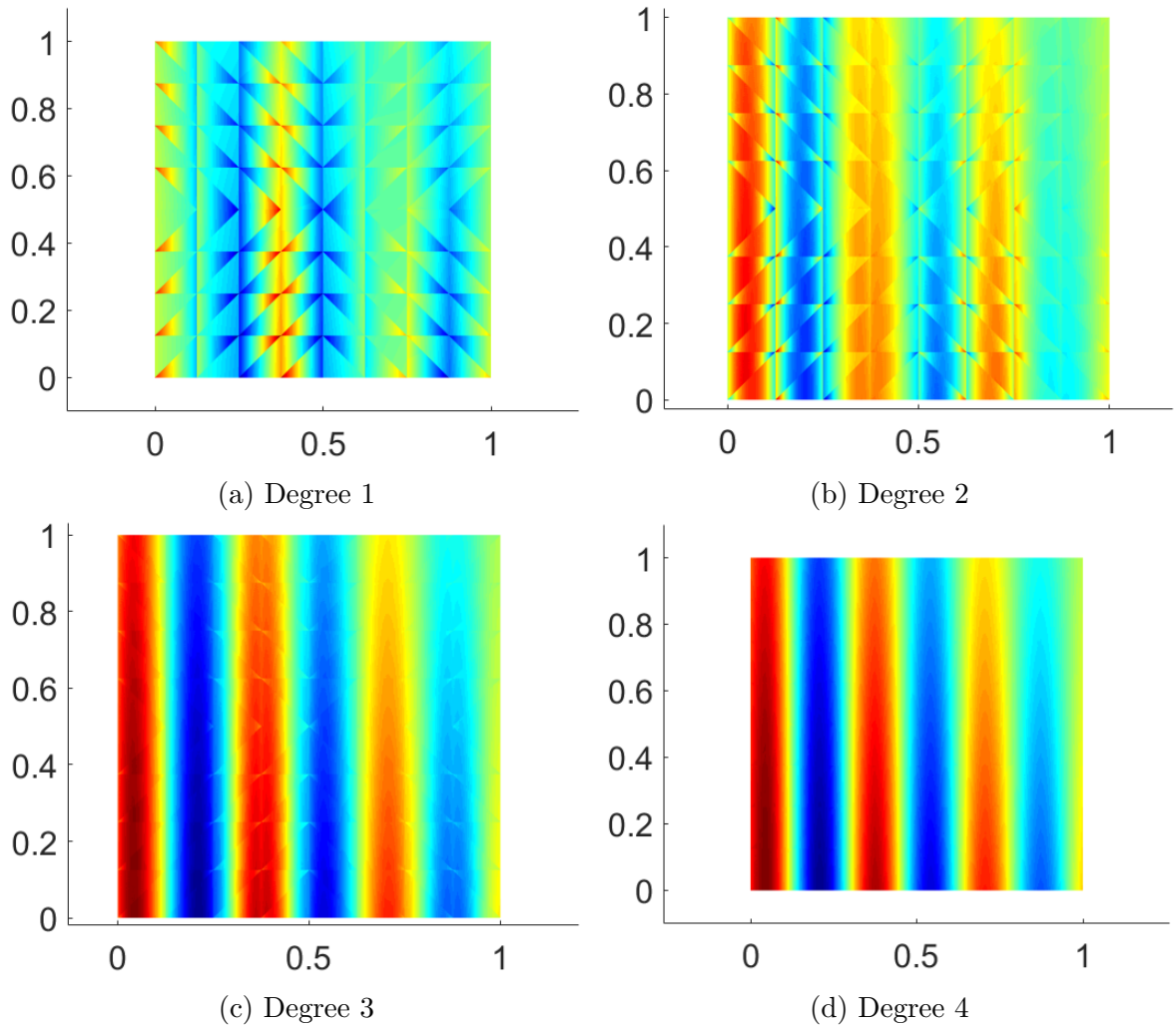


(a) Mesh 1 (b) Mesh 2 (c) Mesh 3

(d) Mesh 4 (e) Mesh 5 (f) Mesh 6

Figure 3.1: Obtained results using quadratic approximation with different meshes

Regarding the effect of the approximation's degree, Figure 3.2 compares the solutions obtained with Mesh no. 3 and degrees from 1 to 4. Again, the lowest the degrees portrait the discontinuities, whilst the higher degrees are capable of providing quite continuous solutions with this relatively coarse mesh.



(a) Degree 1

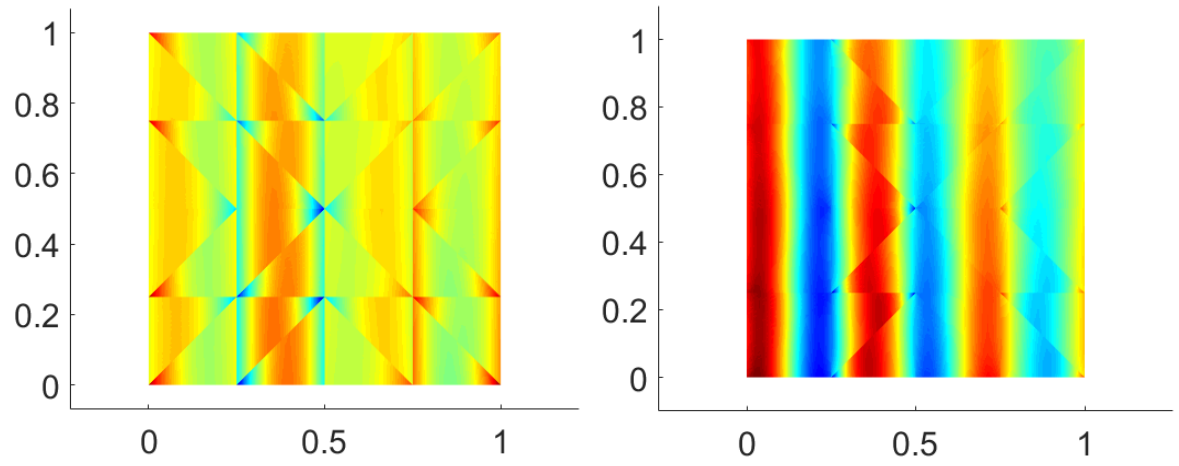
(b) Degree 2

(c) Degree 3

(d) Degree 4

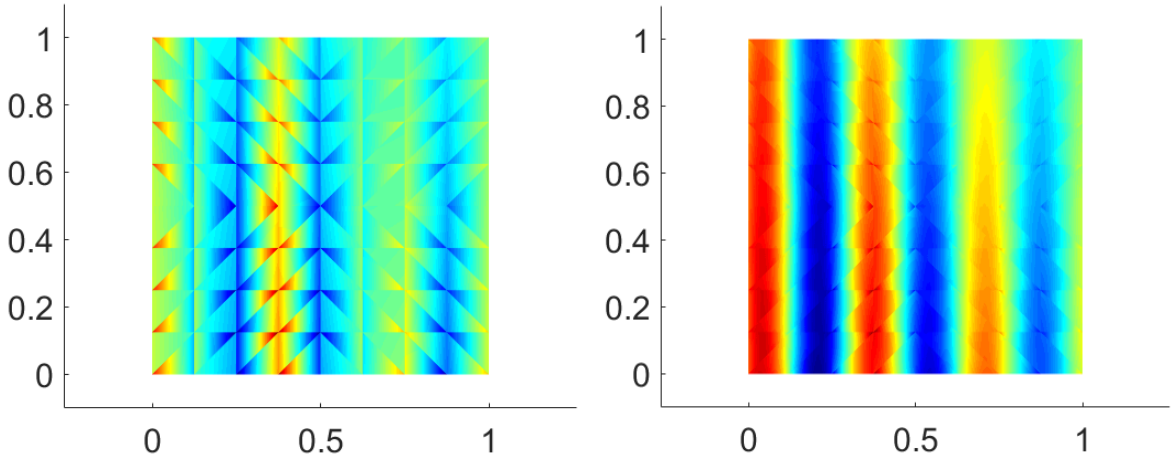
Figure 3.2: Obtained results using mesh no. 3 with different polynomial degrees of approximation

The post-processing aims on allowing the usage of coarser mesh and lower degree of approximation (lowering the computational cost) by re-computing the variables inside the elements by using the obtained fluxes on the edges. As it can be seen on Figures 3.3 and 3.4, this method is quite satisfactory, allowing even a very coarse mesh (such as the Mesh no. 2) with a low order of approximation to provide a result in good agreement to the exact solution. A previously discontinuous result and yields after post-processing a similar response to the analytical one.



(a) Before post-processing (b) After post-processing

Figure 3.3: Effect of post-processing on the solution obtained with the mesh 2 and quadratic approximation



(a) Before post-processing (b) After post-processing

Figure 3.4: Effect of post-processing on the solution obtained with the mesh 3 and linear approximation

Ultimately, a the error for each tested case against the analytical solution is plotted, yielding the convergence regarding mesh for each polynomial degree k considering both post-processed (u) and not post-processed (u^*)

Worth noticing is the effect of the approximation degree. It alters significantly the slope of the convergence curve, diminishing two orders of magnitude of the error for each added degree of approximation (considering the mesh no. 6). To obtain the same level of error, thus, a much coarser mesh can be used. For example, mesh no. 4 with degree

2 of approximation could replace the mesh no. 6 with one degree of approximation to yield the same error level.

The convergence plot also emphasizes the effect of the post-processing, one of the benefits of the HDG method. The difference of the error between post-processes and not post-processed reaches almost three orders of magnitude for the finest meshes. In practice, post-processing the results is even better than increasing the degree of the polynomial, as the curves also show.

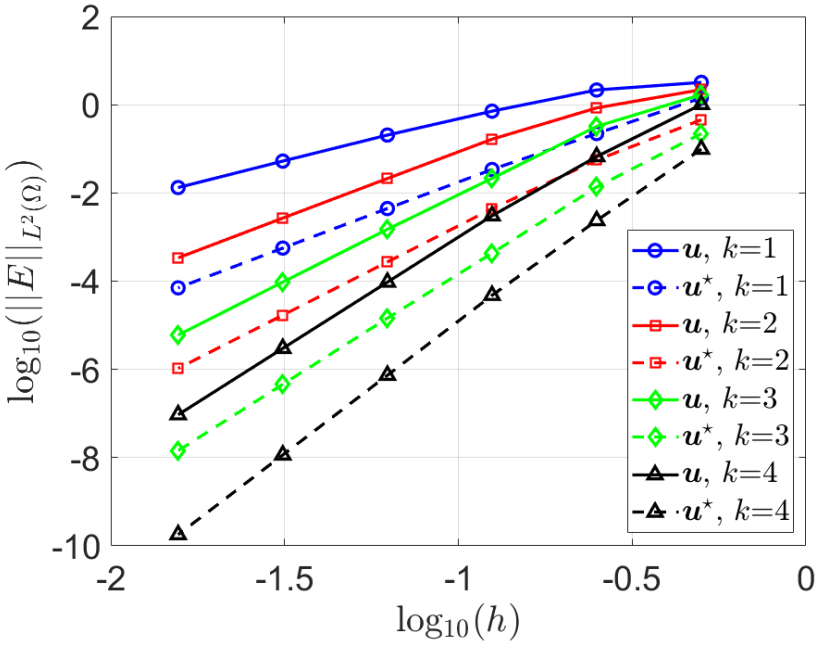


Figure 3.5: Convergence analysis